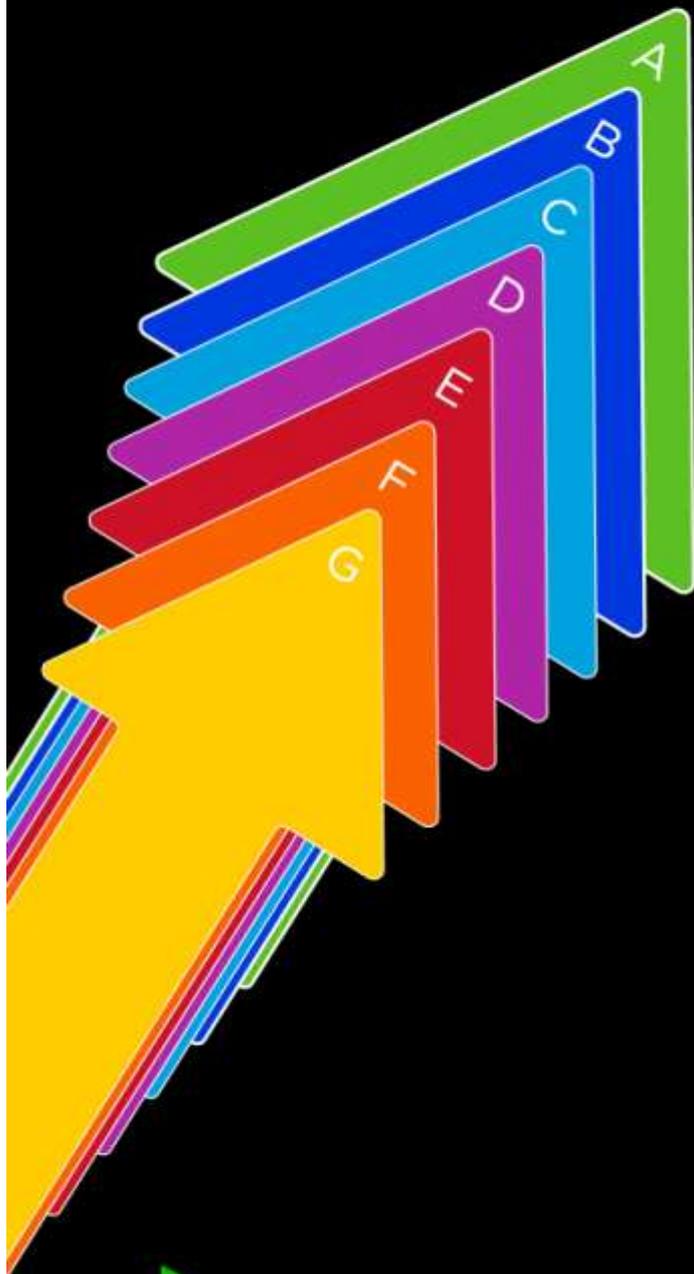


# A HISTÓRIA DA TAHINI-TAHINI

MELHORIA DE PROCESSOS DE SOFTWARE  
COM MÉTODOS ÁGEIS E MODELO MPS



## Autores

Jorge Luis Boria  
Viviana L. Rubinstein  
Andrés Rubinstein

## Desenho

Angela Martín Martín



Ministério da  
**Ciência, Tecnologia  
e Inovação**

GOVERNO FEDERAL  
**BRASIL**  
PAÍS RICO É PAÍS SEM POBREZA

**Presidente da República**

Dilma Vana Rousseff

**Ministro da Ciência, Tecnologia e Inovação**

Marco Antonio Raupp

**Secretário de Política de Informática**

Virgílio Augusto Fernandes Almeida

**Coordenador Geral de Serviços e Programas de Computador**

Rafael Henrique Rodrigues Moreira

**Comitê Editorial**

Ana Liddy Cenni de Castro Magalhães

Ana Regina Cavalcanti da Rocha

Christiane Gresse von Wangenheim

Diva da Silva Marinho

Gleison dos Santos Souza

Kival Chaves Weber

Marcos Vinícius Amorim Ferreira Guimarães

Rodrigo Quites Reis

Sandra Camargo Pinto Ferraz Fabbri

**Jorge Luis Boria**

**Viviana Leonor Rubinstein**

**Andrés Rubinstein**

**A História da Tahini-Tahini:  
Melhoria de Processos de *Software*  
com Métodos Ágeis e Modelo MPS**

---



Julho 2013

A História da Tahini-Tahini:  
Melhoria de Processos de *Software* com Métodos Ágeis e Modelo MPS  
N.9 (2013) – Brasília  
Ministério da Ciência, Tecnologia e Inovação  
Secretaria de Política de Informática, 2013  
201 p.

ISSN 1679-1878

A História da Tahini-Tahini:  
Melhoria de Processos de *Software* com Métodos Ágeis e Modelo MPS  
I. Ministério da Ciência, Tecnologia e Inovação  
Secretaria de Política de Informática  
Jorge Luis Boria  
Viviana Leonor Rubinstein  
Andrés Rubinstein

## PREFÁCIO

A discussão sobre a possibilidade ou não da utilização de métodos ágeis em conjunto com modelos de maturidades em processos de *software* é frequente e atual.

Alguns consideram que as exigências dos modelos de maturidade não podem ser implementadas em organizações com desenvolvimento ágil. Outros consideram que a implementação destes modelos irá ferir a agilidade do desenvolvimento. Esta incompatibilidade é discutida, portanto, por defensores do uso de métodos ágeis e por defensores dos modelos de maturidade.

Neste contexto se situa o livro “A História da Tahini-Tahini: Melhoria de Processos de *Software* com Métodos Ágeis e Modelo MPS” de Jorge Boria, Viviana Rubinstein e Andrés Rubinstein.

O livro teve origem em uma chamada realizada em dezembro de 2011 pela Secretaria de Política de Informática – SEPIN, do Ministério de Ciência, Tecnologia e Inovação – MCTI, responsável pela condução do Programa Brasileiro de Qualidade e Produtividade em *Software* – PBQP *Software*, para o Ciclo 2012 do Programa “Série de Livros do PBQP *Software*”. Entre vários concorrentes foi o texto selecionado para publicação.

E foi, sem dúvida, uma excelente escolha. Nele, os autores, a partir de sua riquíssima experiência como consultores, avaliadores MPS e *lead appraiser* CMMI, nos mostram que não existem contradições entre modelos de maturidade, melhoria de processos e métodos ágeis. Existe, pelo contrário, um excelente caminho que pode ser trilhado com sucesso pelas organizações até serem atingidos níveis muito altos de qualidade e maturidade nos processos de *software*.

De acordo com os autores, o livro tem como objetivo mostrar como se inter-relacionam as técnicas de consultoria, com os métodos ágeis para atender aos resultados esperados do MR-MPS-SW. Para isto utilizam quatro métodos ágeis, Kanban, Scrum, XP e FDD (*Feature Driven Development*), e a história de Tahini-Tahini, uma empresa fictícia que certamente todos gostaríamos que existisse.

É um livro técnico, mas fascinante e de leitura muito agradável. Por vezes nos leva a rir, tamanho o bom humor dos autores ao tratar o tema. Certamente será um caso de sucesso, nesta excelente iniciativa do MCTI.

Agradeço aos autores o convite para prefaciar um livro tão interessante e com contribuições tão importantes para a qualidade e a Engenharia de *Software*.

Abril, 2013

Ana Regina Cavalcanti da Rocha  
Universidade Federal do Rio de Janeiro  
COPPE/UFRJ

## PRÓLOGO - CONSULTORIA EM MELHORIA DE PROCESSOS

### A Origem do Livro

Este livro nasceu de um chamado realizado em dezembro de 2011 pela Secretaria de Política de Informática – SEPIN, do Ministério de Ciência, Tecnologia e Inovação – MCTI, responsável pela condução do Programa Brasileiro de Qualidade e Produtividade em *Software* – PBQP *Software*, para seu Ciclo 2012 do Programa “Série de Livros de PBQP *Software*”. Entre os temas nos quais era possível apresentar propostas, um nos pareceu bastante útil para os profissionais de engenharia de *software*: a Melhoria de Processos de *Software* com Métodos Ágeis e o Modelo MPS. Sobre os outros temas, de igual importância, há muita literatura básica e avançada. Fomos atraídos pelo desafio de conciliar três vertentes, tal a complexidade do tema. Estamos agradecidos a todos que intervieram no processo que levou à seleção, edição e publicação deste livro.

### O Objetivo do Livro

Este livro se coloca como livro de ficção para profissionais. A empresa que é usada como caso de sucesso não existe, nem existiu, esperamos que exista algum dia. Os personagens surgiram de amigos, conhecidos e situações que alguma vez tenhamos vivenciado, como funcionários, consultores ou donos de empresas de *software*. Seu objetivo é ilustrar como se inter-relacionam as técnicas de consultoria, capazes de facilitar o caminho quando é bem feita, às vezes atuando como ensino-aprendizagem, nunca como uma ditadura; com os métodos ágeis, que são muito mais que *Scrum*; para atender aos resultados esperados do MR-MPS-SW.

Este livro não é um receituário, não há nele nenhum algoritmo, nem sequer uma heurística que permita a outros percorrerem o mesmo caminho que o dos protagonistas de nossa história. No entanto, a proposta de utilização que fazemos é que todos possam aprender como identificar problemas e visualizar soluções. Esperamos que os leitores apreciem a história, porque ela está aí para ajudá-los a pensar nas situações que ocorrem diariamente na indústria de *software*.

Por último, este livro não é auto-suficiente. Ele exige que o leitor utilize as referências bibliográficas que apresentamos, seis páginas no total, de materiais superlativos. Se algo destacamos deste livro é que a bibliografia vale a pena por si só.

### As Vertentes do Livro

O título deste livro mistura três ideias poderosas. Fala de melhoria de processos com métodos ágeis e do modelo de referência para melhoria de processos de *software* MPS. Qualquer uma das três ideias merece um livro independente, de maneira que escrever só um, e no curto prazo com que contaram os autores, implica uma escolha de conteúdos. Este é um livro sobre as atividades que são realizadas em consultorias de melhoria de processos, embora a personagem principal que permite criar um fio condutor na história seja uma mulher, Marcela, que trabalha como funcionária da empresa, suas atividades são as de uma consultora de primeiro nível.

Marcela encarna a heroína do romance inglês do século XVIII que é inteligente, sabe o que quer e sabe como conseguir. É o exemplo de liderança deste livro, mesmo com os demais sócios e companheiros de estrada sendo bons gerentes e excelentes profissionais, cada um com sua veia técnica, é Marcela que guia com o exemplo, que questiona o *status quo*, quem, em definitivo, leva a empresa Tahini-Tahini ao cume da qualidade. Quando escrevíamos o livro era com Marcela que preferíamos nos identificar, pois afinal era a personagem mais bem-sucedida. As lições que devem ser aproveitadas deste livro se devem à Marcela.

Não é um livro de consultoria, estes existem e são muito bons, escritos por consultores melhores que nós. No entanto, há muitos conselhos sobre como realizar as coisas importantes, as que levam a mudanças sérias, que estão contidas nas ações dos personagens. Também há muitíssimas técnicas que costumamos introduzir, de um modo ou de outro, em nossas consultorias. O Capítulo 2 inicia o caminho mostrando variantes de métodos de melhoria contínua, culminando com o método *Lean*. Recomendamos leituras adicionais para poder entendê-lo e aproveitá-lo melhor.

Não é um livro sobre o MPS, preferimos que o leitor aprenda sobre este robusto modelo nos seus próprios guias e nos cursos autorizados que são oferecidos. No entanto, não há nada no

livro que não tenha sido escrito com o MPS em mente. Toda a história da Tahini-Tahini, os vaivens com as técnicas, frequentemente apresentadas para sua discussão antes que possam ser aproveitadas, ilustram nosso ponto de vista sobre os modelos de maturidade, centrado no MPS: é preciso ter a visão global do destino para que o caminho possa ser percorrido com comodidade.

Tampouco é um curso de métodos ágeis. O leitor deve entender que, para introduzir um método ágil em uma organização, é preciso um consultor ou um mentor que guie a implantação dia a dia. Produzir uma organização ágil, a partir de uma que não é, exige experiência e adaptação às necessidades da organização.

E embora não seja um livro sobre mudança organizacional, foi desta disciplina que tomamos emprestados muitos conceitos. De qualquer modo, a literatura de mudança organizacional é muito vasta e muito rica, e lhe fariamos pouca justiça se tentássemos resumir-la nestas poucas páginas.

O livro procura descrever as atividades de consultoria que acontecem em muitas organizações. Os autores escolheram um mecanismo de apresentação do material que está no meio do caminho do livro técnico e da narrativa de uma história, o que permite que o leitor se divirta com os personagens. Esperamos que todos se entretendam com a leitura.

### **Nota de Cautela**

Nenhum livro de qualidade pode deixar de citar Deming. Este super-homem da qualidade nos deixou dezenas de pensamentos e ideias para andar com maior comodidade seguindo seus passos. Neste Prólogo queremos lhe render uma pequena homenagem ao mesmo tempo em que usamos suas palavras para advertir o leitor do erro que muitas vezes é cometido com o objetivo de conter gastos: “O Obstáculo de Deming, A esperança da sobremesa instantânea – a suposição de que uma ou duas consultas com um estatístico competente colocarão a empresa no caminho da qualidade e da produtividade – sobremesa instantânea. Não é tão simples, sempre será necessário estudar e trabalhar.”

Não somos estatísticos competentes, mas já vimos o mesmo sintoma em muitas empresas sendo traduzido em um convite para almoçar em troca de um conselho gratuito que depois tenta-se colocar em prática sem os conhecimentos necessários. Os consultores não são insubstituíveis, mas o conhecimento que trazem, este é.

### **Nota Sobre os Autores**

Um livro sempre é uma criação coletiva. Tolkien falava do “húmus” que o autor junta para plantar suas ideias, húmus que é fruto de suas leituras e experiências. As musas inspiradoras só trabalham em mentes abertas que passaram pelas experiências que enriquecem a vida, muitas vezes dolorosas. Mas, além da herança, os autores sempre sentem obrigações com muitas pessoas que possibilitaram o impossível.

*Queremos agradecer especialmente a Ana Regina Cavalcanti da Rocha por sua confiança e amizade, a Kival Chaves Weber, Nelson Henrique Franco de Oliveira e José Antonio Antonioni pelo apoio e as oportunidades brindadas, e a Richard Denney pelo material emprestado.*

**Autores**▪ **Jorge Luis Boria**

*Master of Engineering in Computer Science* pela Cornell University, Estados Unidos. *Visiting Scientist* do *Software Engineering Institute* da *Carnegie-Mellon University*. *Senior Advisor* do Modelo MPS. Avaliador Líder Experiente MPS. SCAMPI *Lead Appraiser* para alta maturidade. Instrutor certificado dos cursos do modelo CMMI. Foi Professor Titular das Universidades UBA, UNICEN, UNSL, USaI, UB e outras na Argentina.

*Jorge quer agradecer especialmente a Joyce Statz pela formação que recebeu trabalhando para ela na TeraQuest Metrics. Joyce foi ao mesmo tempo amiga, formadora, orientadora e treinadora.*

▪ **Viviana Leonor Rubinstein**

Licenciada em Computação Científica pela UBA. *Certified Project Manager*, UT-SQI. Avaliador Líder Experiente MPS. SCAMPI *Lead Appraiser* DEV e SVC para alta maturidade. Instrutora certificada dos cursos do modelo CMMI. Foi Professora Titular nas Universidades UNS em Ushuaia, UBA, UNICEN e outras na Argentina.

*Viviana quer agradecer a Regina, sua mãe, com quem compartilhou a escrita de seu primeiro livro.*

▪ **Andrés Oscar Rubinstein**

Programador e Analista de Sistemas da Pontifícia Universidade Católica do Rio de Janeiro. Avaliador Líder Intermediário MPS. SCAMPI *Lead Appraiser* DEV e SVC. Sócio da TecnoVoz S.A. Argentina. Foi professor e auxiliar docente na PUC-Rio e na Universidad de Belgrano e Colegio Nacional de Buenos Aires na Argentina.

*Andrés quer agradecer a Viviana e a Jorge pela confiança, a Adri e aos amigos de sempre pelo apoio, e a Male e a Nico por existirem.*

*Finalmente, Jorge e Viviana querem agradecer a Beto e Franca por darem um novo sentido a suas vidas. E Viviana e Andrés agradecem a Jorge por sua liderança na confecção deste livro, sem a qual teria sido impossível sua realização.*

---

---

**Sumário**

Prefácio .....	i
Prólogo - Consultoria em Melhoria de Processos .....	ii
A Origem do Livro.....	ii
O Objetivo do Livro .....	ii
As Vertentes do Livro .....	ii
Nota de Cautela.....	iii
Nota Sobre os Autores .....	iii
Autores.....	iv
PARTE I – Introdução	
Capítulo 1 - Introdução .....	1
1.1 O propósito do livro .....	1
1.2 Definição de método ágil para este livro .....	1
1.3 Se a melhoria de processos de <i>software</i> é a resposta, qual é a pergunta? .....	1
1.4 Caso de estudo: A empresa Tahini-Tahini.....	2
Capítulo 2 - O Método da Melhoria Contínua.....	6
2.1 Melhoria de processos .....	6
2.2 Plan-Do-Check-Act.....	8
2.3 O Método IDEAL.....	9
2.4 Focus-Improve-Sustain-Honor .....	11
2.5 <i>Lean Simplified</i> .....	13
Capítulo 3 - Os Métodos Ágeis: <i>Kanban</i> , <i>Scrum</i> , XP e FDD .....	16
3.1 O que são os métodos ágeis?.....	16
3.2 <i>Kanban</i> : medindo o fluxo .....	17
3.3 <i>Scrum</i> : Organizando o sistema para um estado de equilíbrio orgânico .....	21
Momentos da Verdade em um <i>Scrum</i> .....	22
3.4 XP: Inspeções, <i>Design</i> , Cooperação e Muitas Provas.....	23
O Jogo do Planejamento.....	24
Pequenas Entregas .....	24
Metáfora .....	24
<i>Design</i> Simples .....	24
Desenvolvimento Dirigido por Testes .....	24
Refatoração .....	25
Programação em Duplas (ou em Pares).....	25
Propriedade Coletiva (dos produtos pela equipe).....	26

Integração Contínua .....	26
Semana de 40 Horas (hoje chamada Passo Sustentável) .....	26
Cliente Presente .....	26
Padrões do Código.....	26
Escalonamento.....	27
3.5 Feature Driven Development.....	27
3.6 Resumo.....	31
Capítulo 4 - O Modelo de Melhoria de Processos de <i>Software</i> MPS-SW .....	32
4.1 Competir com a excelência .....	32
4.2 Um caminho para a excelência organizacional .....	33
4.3 Arquitetura do MPS.....	35
4.4 Os Níveis de Maturidade do MPS.....	36
4.5 Para que a Mudança Aconteça .....	38
4.6 Quando a mudança é cultural .....	42
4.7 Avaliação do Estado de Maturidade .....	44
PARTE II – Primeiros Passos	
Capítulo 5 - Uma Organização com Problemas (Nível G do MPS-SW) .....	45
5.1 A Pequena História de Tahini-Tahini .....	45
5.2 Quem assume a responsabilidade? .....	46
5.3 Mostrando a Carga de Trabalho e o Estado das Tarefas.....	49
5.4 Planejamento, Monitoramento e Controle do Projeto em Doses Homeopáticas.....	52
5.5 O Cliente quer Mudanças.....	53
5.6 Avanços na implementação do MPS-SW .....	56
5.7 Preparando a Avaliação.....	59
Capítulo 6 - Uma Organização em Crescimento .....	63
6.1 Nível F do MPS-SW .....	63
6.2 Crescem os pedidos.....	63
6.3 Procurando Ajuda Fora da Tahini-Tahini.....	63
6.4 O que deveríamos fabricar? .....	67
6.5 Medindo resultados .....	70
6.6 Protegendo os Ativos .....	76
6.7 Garantindo a Qualidade dos Processos e Produtos .....	78
6.8 A pequena fábrica de <i>software</i> com <i>Scrum</i> .....	80

## Parte III – Evolução

Capítulo 7 - Organizando a Organização .....	85
7.1 Nível E do MR-MPS-SW .....	85
7.2 Uma Empresa em Franco Crescimento .....	85
7.3 A Necessidade de Ativos de Processo .....	93
7.4 Retrospectivas e processos .....	96
7.5 Diminuição de custos por reuso de componentes.....	98
7.6 Utilizando o conhecimento histórico nos projetos .....	99
Capítulo 8 - Eliminando os Defeitos .....	102
8.1 Nível D do MPS-SW .....	102
8.2 Determinando o Problema.....	102
8.3 Busca da Solução .....	104
8.4 Corrigindo os Processos de Requisitos.....	105
8.5 Perfil Operacional.....	112
8.6 Detalhando Um Caso.....	114
8.7 Detectando Defeitos nos Produtos .....	117
8.8 Procedimentos de Verificação .....	119
8.9 Revisões.....	121
8.10 Atividades do Processo de Inspeção .....	121
Reunião de Instrução .....	122
Inspeção Individual do Produto .....	122
Reunião de Unificação .....	122
Decisão sobre o Produto.....	123
Retrabalho e Conclusão.....	123
Informe Final.....	123
8.11 Fatores Críticos de Sucesso .....	124
8.12 Fatores de Fracasso.....	124
8.13 Diferenças Entre Inspeções, <i>Walkthrough</i> e Revisões Estruturadas .....	124
8.14 Usos Ágeis .....	126
8.15 Testes de Produto .....	126
8.16 Critérios Relacionados com Testes.....	127
8.17 Cobertura .....	128
8.18 Projeto e Construção.....	134
8.19 Integração .....	137
Capítulo 9 - Ampliando a Capacidade de Decisão (Nível C do MPS-sw) .....	140

---

9.1 Uma Gestão Ambiciosa .....	140
9.2 Líder em Ação.....	140
9.3 Visão Estratégica dos Riscos.....	141
9.4 Definição de um Risco .....	143
9.5 Captura dos Riscos .....	144
9.6 Estratégias de Controle de Riscos .....	145
9.7 Estratégia Conjunta .....	146
9.8 Nota de Cautela.....	147
9.9 Decisões Formais.....	148
9.10 A Fábrica de Componentes .....	153
9.11 <i>Service Oriented Architecture</i> (SOA) para Principiantes .....	154
9.12 Montando a Fábrica .....	156
9.13 O nível C do MPS .....	157
Parte IV – Apogeu	
Capítulo 10 - Estabilizar para a Melhoria Contínua.....	159
10.1 Níveis B e A do MPS.....	159
10.2 Estabilidade .....	159
10.3 Eliminando Aleatoriedade.....	161
10.4 O Céu Desaba .....	161
10.5 Contenção .....	164
10.6 Causas-Raiz.....	164
10.7 A Previsão Como Ferramenta .....	166
10.8 Previsão e Análise.....	166
10.9 Correlação e Regressão .....	168
10.10 Identificação de processos críticos .....	169
10.11 Processos Capazes.....	172
10.12 <i>Baselines</i> .....	173
10.13 Indicadores Líderes .....	173
10.14 Composição do Processo Definido do Projeto .....	174
10.15 Gestão Quantitativa .....	174
10.16 A Melhoria Contínua Como Estratégia de Negócio.....	175
10.17 Custo de Competir .....	175
10.18 Inovação tecnológica.....	175

Capítulo 11 - Conclusões .....	179
Referências Bibliográficas .....	180

---

---

## Lista de Figuras

Figura 1.1: Relação entre ferramentas e competência das pessoas.....	2
Figura 2.1: O Método IDEAL (adaptado de [McFEELEY, 1996]).....	9
Figura 2.2: Visão de Melhoria de Processos do IDEAL (adaptado de [McFEELEY, 1996]) .....	11
Figura 3.1: Painel <i>kanban</i> .....	18
Figura 3.2: Nota do Painel <i>kanban</i> .....	19
Figura 3.3: Processo <i>Scrum</i> .....	22
Figura 3.4: Ciclo de Desenvolvimento de XP.....	25
Figura 3.5: Ciclo de Desenvolvimento de FDD.....	28
Figura 3.6: Árvore de Funções derivada da Arquitetura .....	30
Figura 4.1: Relação da Refatoração vs. Nível do CMM [DIAZ e KING, 2002] .....	32
Figura 4.2: Organização do MPS.BR [SOFTEX, 2011] .....	34
Figura 4.3: Componentes do Modelo MPS [SOFTEX, 2012a] .....	34
Figura 4.4: Níveis de maturidade do MR-MPS-SW [SOFTEX, 2012a].....	35
Figura 4.5: Estrutura do MPS [SOFTEX, 2011] .....	36
Figura 4.6: Sequencia da Resistência à Mudança (adaptado de [KUBLER-ROSS, 1997]) .....	39
Figura 4.7: Modelo de Transição Ilusória (1) .....	40
Figura 4.8: Modelo de Transição Ilusória (2) .....	40
Figura 4.9: Modelo de Transição Administrada .....	40
Figura 4.10: Modelo de Transição Sem Administrar.....	41
Figura 4.11: Passos do Comprometimento (adaptado de [CONNER 1982]) .....	41
Figura 4.12: Valores em Competência (adaptado de [CAMERON e QUINN, 2011]).....	43
Figura 5.1: Causas da Falta de Conhecimento do Estado do Projeto.....	47
Figura 5.2: Painel <i>kanban</i> elementar .....	50
Figura 5.3: Painel <i>kanban</i> com ciclo de vida das histórias -1-.....	50
Figura 5.4: História no Painel <i>kanban</i> .....	51
Figura 5.5: Painel <i>kanban</i> com ciclo de vida das histórias -2-.....	53
Figura 5.6: Planilha para Definição de Histórias .....	54
Figura 5.7: Planilha para Definição e Análise de Riscos .....	55
Figura 5.8: Planilha de Proposta de Projeto .....	55
Figura 5.9: Diagrama de Raias .....	58
Figura 5.10: Planilha de Detalhamento de um Procedimento .....	61
Figura 5.11: Evidências para GPR no Nível G.....	62
Figura 6.1: Diagrama <i>Ishikawa</i> sobre Crescimento 1 .....	64
Figura 6.2: Diagrama <i>Ishikawa</i> sobre Crescimento 2 .....	64
Figura 6.3: Organograma da Tahini-Tahini .....	70
Figura 6.4: Dados vs. Informação .....	71
Figura 6.5: Gráfico sobre Preços Futuros do Petróleo .....	72
Figura 6.6: Processo de Auditoria da Qualidade .....	80
Figura 6.7: A Morte do <i>Scrum</i> .....	81
Figura 6.8: Cobertura dos Resultados Esperados com <i>Scrum</i> e <i>Kanban</i> .....	82
Figura 7.1: Formulário Completo Missão e Função para Engenheiro de Teste .....	87
Figura 7.2: Análise de Opções sobre Reuso .....	99
Figura 8.1: Estrutura Típica de uma Folha de Resultados Equilibrados .....	103
Figura 8.2: Diagrama de Contexto de um Sistema .....	106
Figura 8.3: Diagrama de Classe de Acordo .....	107
Figura 8.4: Diagrama de Classes de Acordo .....	108
Figura 8.5: Processo de Captura de Requisitos .....	110
Figura 8.6: Resultado dos Passos 1 e 2 .....	110
Figura 8.7: Diagrama de Arquitetura .....	111
Figura 8.8: Modelo V de Desenvolvimento de <i>Software</i> .....	120
Figura 8.9: Zona de Caos por Postergação de Atividades de Remoção.....	120
Figura 8.10: Modelo em V com Revisões entre Atividades de Tradução .....	121
Figura 8.11: Diagrama de Fluxo do Caso de Uso da Tabela 8.14.....	129
Figura 8.12: Diagrama de Fluxo de Controle com Funcionalidade Abstraída .....	129
Figura 8.13 Caminho Feliz .....	130
Figura 8.14 Primeira Área Fechada .....	130
Figura 8.15 Segunda Área Fechada .....	131
Figura 8.16 Terceira Área Fechada .....	131
Figura 8.17 Quarta Área Fechada.....	131

---

Figura 8.18 Quinta Área Fechada .....	131
Figura 8.19 Sexta Área Fechada .....	132
Figura 8.20 Todas as Áreas Fechadas .....	132
Figura 8.21: Probabilidade de Cada Transição do Grafo .....	133
Figura 9.1: Árvore de Decisão.....	141
Figura 9.2: Planilha de Definição, Monitoração e Controle de Riscos.....	144
Figura 9.3: Matriz de Riscos.....	146
Figura 9.4: Árvore de Objetivos.....	150
Figura 9.5: Árvore de Decisão Refinada .....	151
Figura 9.6: Tabela de Pagamentos Correspondente à Árvore de Decisão Refinada.....	151
Figura 9.7: Diagrama de Influências com Inclusão de Outros Investimentos.....	152
Figura 9.8: Diagrama de Tornado .....	153
Figura 9.9: Ilustração da Arquitetura SOA .....	156
Figura 10.1: Distribuições de Esforço de Tarefas Semelhantes em Duas Populações .....	160
Figura 10.2: Distribuição do Erro em Dois Relógios .....	161
Figura 10.3: O Método das Oito Disciplinas.....	163
Figura 10.4: Curvas da Família Weibull .....	167
Figura 10.5: Zonas de SPC Sob a Distribuição Normal.....	167
Figura 10.6: Diagrama do Modelo de Kano .....	169
Figura 10.7: Barreiras à Qualidade .....	170
Figura 10.8: Análise de Fatores CTQ .....	171
Figura 10.9: BSC Aplicado a Identificar Processos Críticos .....	171
Figura 10.10: Fatores Na Escolha de Processos Críticos .....	172
Figura 10.11: Processos Capazes e Processos Estáveis .....	173
Figura 10.12: Indicadores Líderes .....	173
Figura 10.13: Diferentes Possibilidades de Composição do Processo .....	174
Figura 10.14: Definição de Adjacências e Espaço em Branco Segundo Johnson.....	176
Figura 10.15: Construir-Medir-Aprender .....	177
Figura 10.16: Diagrama de Pareto de Defeitos de Código .....	177

---

---

## Lista de Tabelas

Tabela 2.1: Seleção do Método de Melhoria de Processos .....	15
Tabela 5.1: Tamanhos de Tarefas .....	51
Tabela 5.2 Processo GERÊNCIA DE PROJETOS no Nível G [SOFTEX, 2012a] .....	57
Tabela 5.3 Processo GERÊNCIA DE REQUISITOS [SOFTEX, 2012a] .....	59
Tabela 6.1: Pensamentos Negativos sobre a Mudança .....	64
Tabela 6.2: Preparando-nos para Crescer .....	65
Tabela 6.3: Processo AQUISIÇÃO [SOFTEX, 2012a] .....	66
Tabela 6.4: Matriz de Pugh sobre Propostas .....	68
Tabela 6.5: Riscos do Crescimento .....	69
Tabela 6.6: Processo GERÊNCIA DE PORTFÓLIO DE PROJETOS [SOFTEX, 2012a] .....	70
Tabela 6.7: Missão e Funções .....	71
Tabela 6.8: Riscos Associados .....	73
Tabela 6.9: Processo MEDIÇÃO [SOFTEX, 2012a] .....	74
Tabela 6.10: Definição de Medidas .....	74
Tabela 6.11: Definição de Indicadores .....	76
Tabela 6.12: Riscos Derivados da Falta de Controle de Ativos .....	76
Tabela 6.13: Propriedades de Cada Tipo de Item de Configuração .....	77
Tabela 6.14: Processo GERÊNCIA DE CONFIGURAÇÃO [SOFTEX, 2012a] .....	78
Tabela 6.15: Riscos de não Auditar .....	79
Tabela 6.16: Severidade de Não conformidades em Auditorias .....	79
Tabela 6.17: Processo GARANTIA DA QUALIDADE [SOFTEX, 2012a] .....	80
Tabela 7.1: Atividades de Recrutamento e Incorporação de Pessoal .....	89
Tabela 7.2: Porcentagem de Êxito em Atividades Seleccionadas por Tipo de Cargo .....	90
Tabela 7.3: Riscos da T <sup>2</sup> Derivados de Políticas Pobres em Recursos Humanos .....	90
Tabela 7.4: Primeira Parte de um Modelo de Competências .....	92
Tabela 7.5: Lista de Avaliação Correspondente à Primeira Tarefa .....	92
Tabela 7.6: Processo GERÊNCIA DE RECURSOS HUMANOS [SOFTEX, 2012a] .....	93
Tabela 7.7: Orientação Sugerida por Perfil de <i>Sprint</i> .....	96
Tabela 7.8: Processo DEFINIÇÃO DO PROCESSO ORGANIZACIONAL [SOFTEX, 2012a] .....	96
Tabela 7.9: Processo AVALIAÇÃO E MELHORIA DO PROCESSO ORGANIZACIONAL [SOFTEX, 2012a] .....	97
Tabela 7.10: Processo GERÊNCIA DE REUTILIZAÇÃO [SOFTEX, 2012a] .....	99
Tabela 7.11: Processo GERÊNCIA DE PROJETOS (a partir do nível E) [SOFTEX, 2012a] .....	101
Tabela 8.1: Problemas de Requisitos .....	105
Tabela 8.2: Comparação entre Métodos de Desenvolvimento pelo Benefício .....	109
Tabela 8.3: Comparação entre Métodos de Desenvolvimento pelo Risco .....	109
Tabela 8.4: Matriz CRUD sem Completar .....	111
Tabela 8.5: Matriz CRUD já Completada .....	112
Tabela 8.6: Estimativas de Atividade .....	113
Tabela 8.7: Perfil Operacional dos Casos de Uso .....	113
Tabela 8.8: Componentes Sugeridos dos Casos de Uso .....	115
Tabela 8.9: Lista de Controle de Mitigação de Riscos em Requisitos .....	116
Tabela 8.10: Implementação de DESENVOLVIMENTO DE REQUISITOS na T <sup>2</sup> .....	117
Tabela 8.11: Problemas de Validação .....	118
Tabela 8.12: Problemas de Verificação .....	119
Tabela 8.13: Comparação entre Revisões .....	125
Tabela 8.14: Planilha de Registro de Questões .....	126
Tabela 8.15: Exemplo de Sequência Principal e Alternativa de um CDU .....	128
Tabela 8.16: Resultados Esperados de VERIFICAÇÃO e Atividades Internas em T <sup>2</sup> .....	134
Tabela 8.17: Resultados Esperados de VALIDAÇÃO e Atividades Internas em T <sup>2</sup> .....	134
Tabela 8.18 Processo PROJETO E CONSTRUÇÃO DO PRODUTO [SOFTEX, 2012a] .....	135
Tabela 8.19: Problemas de <i>Design</i> .....	135
Tabela 8.20: Análise de Opções sobre <i>Design</i> .....	136
Tabela 8.21: Cobertura de Resultados Esperados de PCP .....	137
Tabela 8.22: Ações Relacionadas com Integração, Derivadas de Retrospectivas .....	138
Tabela 9.1: Tabela de Pagamentos da Árvore de Decisão .....	141
Tabela 9.2: Estratégia de Riscos de Alto Nível, Fontes e Categoria .....	143
Tabela 9.3: Exemplo de Tabela (Parcial) de Riscos .....	147
Tabela 9.4: Definição dos Passos do PAAeTdD .....	149

Tabela 10.1: Os Problemas do Projeto Fora de Controle ..... 165  
Tabela 10.2: A Última Filha da Análise depois de Completa ..... 168

---

---

# PARTE I – Introdução

---

## CAPÍTULO 1 - INTRODUÇÃO

### 1.1 O propósito do livro

Este livro é dirigido a (em ordem crescente de interesse):

- implementadores de melhoria de processos de *software* que queiram conhecer melhor os métodos ágeis para implantá-los em organizações da área;
- gerentes de projeto interessados em conhecer melhor os métodos ágeis para desenvolvimento de *software*, seja para adotá-los ou para avaliar a sua adoção;
- engenheiros de *software* que tentam trabalhar em um projeto ágil;
- professores de graduação em Computação;
- estudantes de graduação em Computação;
- professores de pós-graduação em Engenharia de *Software*;
- estudantes de pós-graduação em Engenharia de *Software*.

Na medida que os métodos ágeis e os modelos de maturidade têm sido considerados termos opostos nas disciplinas de desenvolvimento e manutenção de *software*, é difícil conceber um texto que tente fazer a ponte entre os dois mundos. Já foi feito, porém, com grande sucesso, no moderno clássico [BOEHM e TURNER, 2003]. O modelo de referência para eles é o CMMI, sendo fácil imaginar a passagem para o MR-MPS-SW, dada a intencional compatibilidade entre os dois modelos.

### 1.2 Definição de método ágil para este livro

Este livro está focado, principalmente, em quatro métodos ágeis<sup>1</sup>: *Kanban*, *Scrum*, XP e FDD (*Feature Driven Development*). A escolha não é por acaso. Esses quatro métodos cobrem a maioria das implementações ágeis realizadas no mundo. Além disso, eles cobrem quase todas, se não todas, as necessidades de uso de métodos ágeis.

Cada um destes métodos será devidamente explicado no Capítulo 3. Obviamente, isto ocorrerá em ordem crescente de complexidade: primeiro o mais simples e o que tem o maior retorno em comparação ao investimento, o *Kanban*. Ele possui grande retorno porque, ao organizar as tarefas e detectar rapidamente os problemas, permite à equipe que o utiliza aumentar o tempo disponível para aprimorar seus processos e tentar novas melhorias. *Scrum*, que frequentemente é o método eleito desde o começo, aqui só é visto quando a empresa conseguiu se estabilizar o suficiente para ter tempo de conseguir que as atividades do *Scrum* possam ser seguidas com a disciplina necessária. Os outros dois métodos se somam aos anteriores à medida em que a empresa ganhe em definição de seus processos e no número de seus colaboradores.

### 1.3 Se a melhoria de processos de *software* é a resposta, qual é a pergunta?

O principal inimigo de uma empresa desenvolvedora de *software* é a baixa qualidade. Até hoje, ninguém desenvolveu uma proposta válida para melhorar a qualidade que não fosse relacionada à melhoria de processos, que passa a ser então a questão principal. É possível argumentar que as pessoas e as ferramentas (de *software*, como CASE) são importantes em seu impulso para a melhoria da produtividade. Isto é verdade, mas só quando os processos estão no lugar onde se consiga o aproveitamento das condições dos indivíduos e das

---

<sup>1</sup> As principais referências de cada um dos métodos estão indicadas quando se descreve cada um nos capítulos seguintes.

ferramentas de *software*. É comum o caso de empresas e organizações<sup>2</sup> que não aproveitam seus recursos humanos e têm licenças que não são usadas. Por isso deduzimos que, por mais importante que sejam as ferramentas e as competências das pessoas, são os processos que, realmente, tornam possível o aumento de produtividade.

Na Figura 1.1 simbolizamos isto com ícones. Na primeira “equação”, as pessoas capacitadas, somadas às ferramentas de *software* e a processos bem definidos culminam (depois do sinal de igual) em sucesso e felicidade. Na segunda equação, a falta de processos bem definidos aumenta os riscos e produz frequentes problemas nos produtos resultantes.

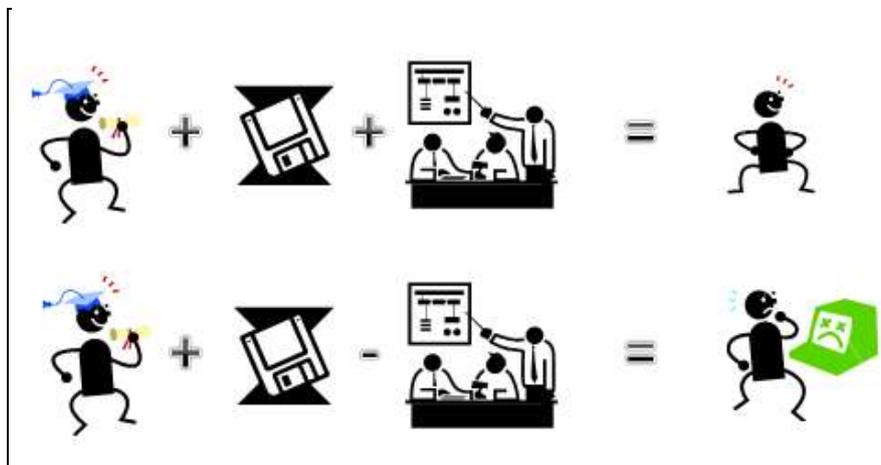


Figura 1.1: Relação entre ferramentas e competência das pessoas

A disciplina induzida pelos processos é a que permite aproveitar os conhecimentos e as ferramentas. Sem esta disciplina não é possível reproduzir os sucessos que podem ser alcançados pelos projetos, porque a memória organizacional é perdida.

#### 1.4 Caso de estudo: A empresa Tahini-Tahini

Neste livro, acompanhamos o desenvolvimento de uma organização que nasce de uma ideia de estudantes universitários. A empresa criada é pequena e, como uma piada entre eles, recebe o nome de Tahíni-Tahíni. O nome surgiu quando uma das fundadoras chegou à reunião de criação um pouco atrasada e ao olhar o número de pessoas ao redor da mesa disse: “Somos uma empresa *tiny-tiny*”. Seus futuros sócios acharam isso engraçado e colocaram o nome de Tahíni-Tahíni, fazendo um duplo jogo de palavras. No livro, geralmente vamos nos referir a ela pelas siglas que seus sócios usam para chamá-la: T<sup>2</sup>, TT ou o Duplo T.

Como toda empresa recém-nascida, criada por jovens empreendedores, não foi seguido um plano de crescimento ideal; na verdade ela cresceu aos trancos e os mares agitados que tiveram de contornar foi o que a fortaleceu. Os problemas da empresa não são incomuns, são os mais frequentes para uma organização desse tamanho e com essa história, em cada etapa de seu crescimento. A cada passo, os sócios tiveram de tomar decisões que afetaram os resultados e, a cada oportunidade, alteraram os processos que guiavam o desenvolvimento do produto. Em cada caso quiseram aprimorar a qualidade e o controle dos processos para melhorar a qualidade e o controle sobre o produto.

Durante o desenvolvimento do caso de estudo utilizaremos a descrição de *Kanban* para o início de um projeto de melhoria de processos; *Scrum* em relação às atividades de gerência de projetos mais abrangentes; *XP (Extreme Programming)* para o que constitui a categoria das áreas de engenharia tratadas no nível D (Largamente Definido) do MR-MPS-SW. Quando uma empresa cresce, os métodos anteriores começam a apresentar limitações. A coordenação de muitos grupos em *Scrum* de *Scrums* apresenta maiores limitações que os métodos tradicionais. *XP* é difícil de controlar. Acompanhando o crescimento da T<sup>2</sup>, a proposta é um método baseado no *Feature Driven Development (FDD)*. Em torno da questão da mudança organizacional, seguiremos o caminho do método *Lean*, que se concentra principalmente na resolução de problemas por meio da melhoria de processos.

<sup>2</sup> Neste livro utilizaremos o termo “organização” para nos referir a todo tipo de grupo humano organizado para a realização de tarefas com um propósito comum, seja com ou sem fins lucrativos.

Neste capítulo também descrevemos os capítulos restantes. Cada capítulo que faz referência a um nível do MR-MPS-SW explica os resultados requeridos dos processos seguindo as exigências do modelo. O livro está dividido em quatro partes, cada uma atendendo a uma necessidade diferente. Na primeira parte (Capítulos 1 a 4) estão colocadas as bases para que o leitor possa compreender os métodos e a filosofia de trabalho que os autores propõem. A segunda parte (Capítulos 5 e 6) trata dos temas de baixa maturidade (Níveis G e F do MPS) e introduz em detalhes os primeiros passos da T<sup>2</sup> e a solução encontrada pelos sócios com base no uso de *Kanban* e *Scrum*. A terceira parte (Capítulos 7 a 9) desenvolve a temática da Maturidade Média, os níveis E, D, e C do Modelo MPS-SW, onde aparecem XP e mais adiante FDD. Finalmente, a quarta parte (Capítulos 10 e 11) expõe como a maturidade definida da T<sup>2</sup> permite alcançar um conhecimento profundo do funcionamento de seus processos, caracterizando-os quantitativamente. O livro termina com um resumo da viagem da T<sup>2</sup> desde sua criação até sua venda bilionária.

No Capítulo 2 explicamos nossa filosofia de melhoria de processos. Para isso, utilizamos o método *Lean*, palavra inglesa que significa enxuto, porque é o que melhor se adapta às nossas ideias. Adotando mensagens de diversas fontes, explicamos como é melhor fazer o mínimo possível para resolver um problema do que fazer grandes mudanças sem efeito aparente. Também aproveitamos o Capítulo 2 para falar de uma visão sistêmica das organizações e da relação multicausal dos eventos. Deste modo, preparamos o leitor para entender porque uma só ação pode resolver muitos problemas e como às vezes é necessário aplicar múltiplas ações (e esperar) para obter resultados. O livro *Leading Lean Software Development* de [POPPENDIECK *et al.*, 2010], é nosso guia por este território. Também, onde for útil, citaremos material de *Thinking in Systems, A Primer* do clássico de [MEADOWS, 2008]. Os dois livros revolucionam o pensamento clássico, linear, dos gerentes tradicionais, e abrem a porta a uma gerência mais ágil, mais integrada e com maiores possibilidades de sucesso.

No Capítulo 3 apresentamos os métodos ágeis propriamente ditos. Apesar de *Lean* ser um método ágil segundo seus criadores, e é reconhecido como tal pela comunidade ágil<sup>3</sup>, seu uso exige muito conhecimento e está fora do escopo deste livro. Por outro lado, as técnicas propostas por *Kanban*, *Scrum*, XP (*Extreme Programming*) e *Feature Driven Development* (FDD) são igualmente bem-sucedidas e muito mais fáceis de adotar, sobretudo se forem feitas na ordem proposta neste livro. Esta é uma introdução aos métodos e, em nenhum momento, deve substituir a leitura dos textos clássicos sobre o tema, os quais estão listados na bibliografia e são recomendados ao leitor.

O Capítulo 4 está dedicado ao eixo central deste livro, o modelo de Melhoria de Processos de *Software* (MR-MPS-SW). Outra vez, o livro é incapaz de conter todo o conhecimento necessário para fazer um bom uso do modelo, de modo que indicamos ao leitor os guias publicados pela SOFTEX e que se encontram acessíveis no *website* desta organização<sup>4</sup>. Os guias são material indispensável para o leitor que quer seguir nossas sugestões e implementar métodos ágeis com o MPS. O que exploraremos profundamente são as grandes pinceladas que precisam ser entendidas para que o modelo faça sentido e ninguém se perca nos detalhes que devem ser aplicados. Vamos tratar da evolução da cultura existente na empresa, induzida pelo amadurecimento dos processos por meio da manifestação de mudança seguindo os níveis de maturidade do modelo, conceito em que nos deteremos neste capítulo, assim como na arquitetura do modelo que permite encontrar nele as ferramentas para sua implementação. Para concluir o capítulo, explicaremos o conceito de avaliação e como uma organização pode aplicá-lo a si mesma, para entender onde se encontra e que caminho lhe falta percorrer.

O Capítulo 5 apresenta o detalhe dos problemas iniciais da T<sup>2</sup>. Utilizando exemplos que foram encontrados em sua atuação como consultores e avaliadores de processos, os autores apresentam ao leitor os problemas típicos de uma empresa que funciona bem quando todas as coisas estão no caminho certo. Os “pequenos” problemas cotidianos (uma gravidez, uma região sem recepção telefônica, um cliente apressado) podem desencadear uma “tormenta perfeita” que acaba arruinando até mesmo a melhor reputação. É aí que os amigos da T<sup>2</sup> decidem introduzir métodos de controle e, aconselhados corretamente, começam a utilizar *Kanban*. Ao mesmo tempo, conseguem implantar sem muito esforço extra, processos do

<sup>3</sup> A comunidade ágil se reconhece no website: <http://www.agilemanifesto.org/>

<sup>4</sup> [http://www.softex.br/mpsbr/\\_guias/default.asp](http://www.softex.br/mpsbr/_guias/default.asp)

modelo MPS. Tentados pela facilidade com que implementaram os processos do Nível G, os sócios decidem realizar uma avaliação formal com um avaliador líder e passam com sucesso. A adoção do MPS pela empresa T<sup>2</sup> agora é um fato.

No Capítulo 6, os autores contam como os sócios, alentados pelo sucesso obtido por suas melhorias de processo, decidem aprofundar o caminho e utilizam o modelo MPS para isso. Os controles estabelecidos abrem espaço para a gerência de configuração que, de forma INICIAL, começou a ser implementada no nível G; para a medição que, devido à formação profissional dos sócios, é considerada uma atividade fundamental para controlar e dirigir a empresa; e para o controle da qualidade, imposto pela realidade.

A chegada de novos clientes com pedidos de projetos, que são às vezes de adaptação e às vezes de novos desenvolvimentos, faz com que os processos da gerência de portfólio de projetos sejam valiosos para entender como organizar o crescimento da demanda. Este crescimento faz com que dois caminhos sejam avaliados: crescer internamente, o que significaria ampliar o espaço físico para admitir mais desenvolvedores, com o conseqüente investimento fixo; ou subcontratar parte do trabalho. Para tomar a decisão, os sócios se apoiam no processo de aquisição e decidem a favor da subcontratação, o que os leva a implementar este processo. Ainda assim, a pequena equipe inicial deve se dividir para atender o novo volume de trabalho, e incorporam um pequeno número de desenvolvedores. Para organizar os projetos, os sócios decidem utilizar práticas de *Scrum*, que são compatíveis com o MPS. Para assegurar-se de que está no caminho certo, a T<sup>2</sup> passa por uma nova avaliação e alcança o Nível F.

No Capítulo 7 começa a terceira parte, que desenvolve os processos intermediários do modelo MPS. À medida que se expandem os negócios e novas oportunidades surgem, os sócios se veem obrigados a expandir, da mesma forma, seus escritórios. Uma reunião fundamental os coloca em uma encruzilhada: para onde queremos levar a T<sup>2</sup>? Finalmente, decidem criar uma visão ambiciosa: ser uma das dez melhores fábricas de *software* do mundo. Preparando-se para o crescimento, os sócios entendem que a visão deles é complementada com uma base de conhecimentos que pode ser compartilhada, usada e expandida por todos os engenheiros e demais funcionários da empresa. Seus processos de gestão evoluem da mesma forma, para aproveitar este conhecimento de maneira sistemática. De maneira natural surge, na base de conhecimentos, a definição dos processos organizacionais.

Quando os colaboradores da T<sup>2</sup> superam um número considerado crítico de maneira arbitrária, as reuniões de processo se sistematizam e se formalizam em um projeto para sua avaliação e melhoria permanente. As constantes incorporações forçam, da mesma maneira, a organização a tratar da identificação, captação, preparação e retenção dos recursos humanos. Em todas estas tarefas, o MPS acaba sendo fundamental e indispensável, ao dar um marco coerente e as pautas culturais para crescer. O resultado é uma organização que aprende, com colaboradores motivados, que continuamente fazem propostas de melhoria dos seus próprios processos e também dos demais. Uma iniciativa brilhante sai de uma destas propostas e a T<sup>2</sup> incorpora práticas de reuso para melhorar ainda mais a qualidade de seus produtos e o rendimento de seu pessoal.

No segundo capítulo da terceira parte, o Capítulo 8, apresentamos as técnicas e práticas de *Extreme Programming* (XP) que não se sobrepõem às anteriores já incorporadas. Uma discussão em retrospectiva leva a identificar a variação na interpretação das técnicas de desenvolvimento nas equipes como a fonte de diferenças entre as estimativas iniciais, agora desenvolvidas a partir da história e dos resultados reais. Discutindo na reunião do grupo de processos organizacionais vincula-se, da mesma forma, essa indefinição com um conjunto de defeitos que estão saindo repetidamente ao mercado. Uma proposta de adoção do XP é recebida sem muito entusiasmo, mas de todo modo é implementada, cuidando de que, ao fazer isso, sejam respeitadas as condições para continuar cumprindo com a implementação de processos de MPS. Isto leva a algumas variantes, como por exemplo que *pair programming*, a técnica por meio da qual dois programadores trabalham juntos em um mesmo programa, seja implementada com um *coach* que registra os defeitos encontrados para permitir a realização de suas análises. As equipes incorporam, igualmente, *software* de controle e introduzem variantes aos processos para continuar avançando e permanecer dentro do marco do MPS.

Confrontados com a realidade de seu crescimento e os riscos que ela traz, os sócios incorporam uma visão estratégica de seu negócio e, mais uma vez, fazem isso seguindo o modelo. No Capítulo 9, é introduzida a visão do risco como constante. A T<sup>2</sup> não define a si

mesma como uma empresa que quer evitar os riscos, mas conhecê-los e enfrentá-los. Desse modo, é capaz de se preparar melhor para afrontar o que a realidade colocar em seu caminho. Os riscos assim analisados são melhor enfrentados e a capacidade desenvolvida de melhoria de processos é, nesse sentido, uma ferramenta. Por exemplo, ao invés de aproveitar de forma oportuna o reuso quando aparece uma necessidade de reusar partes já existentes nos projetos concluídos, a T<sup>2</sup> organiza uma fábrica de componentes que podem ser articulados rapidamente para formar produtos, reduzindo ainda mais seus defeitos por elemento e aumentando a níveis muito altos sua produtividade. Cada decisão tem um custo e um benefício, mas até este momento na história da T<sup>2</sup> não se aprendia sistematicamente a partir das decisões já tomadas. Para evitar que esse conhecimento se perca, a T<sup>2</sup> incorpora métodos formais de decisão que incluem várias técnicas que aproveitam dados históricos. Logo, são usados por projetos para tomar decisões sobre a velocidade, a qualidade esperada, o reuso e a subcontratação de terceiros. A T<sup>2</sup> já é uma organização com centenas de funcionários e possui sólida reputação de qualidade. Os contatos e indicações obtidos por referência de outros clientes começam a ser internacionais. Devido a esse crescimento e ao conseqüente distanciamento físico dos clientes, a T<sup>2</sup> acrescentou a seu arsenal o método FDD, que permite planejar os *sprints* com maior precisão. A T<sup>2</sup> decide ser avaliada em relação ao Nível C do modelo MPS, em uma avaliação conjunta com o Nível Definido do modelo CMMI-DEV, o que consegue com incrível sucesso.

Ao ingressar na Quarta Parte do livro, encontramos a T<sup>2</sup> em seu apogeu. Abre escritórios em vários países centrais, tem centros de desenvolvimento em todas as regiões do Brasil e da América Latina, e desfruta de um merecido prestígio. No entanto, os sócios não descansam em seus louros. No Capítulo 10, vemos como aproveitam a base de dados histórica que construíram ao longo dos anos para utilizá-la a seu favor. Identificam os processos críticos que se relacionam com seus objetivos de negócio, analisam sua estabilidade relativa, constroem modelos que permitem prever resultados futuros em pontos iniciais de um projeto e ajudam a corrigir problemas. Estendem as técnicas que empregam na tomada de decisões para incluir fatores quantitativos e melhoram suas análises de causa-raiz para que se considere o custo-benefício das possíveis alternativas. A gerência de projetos passa a ser uma ciência com aspectos de arte e não mais uma arte com aspectos de engenharia.

O Capítulo 11 fecha o livro com os sócios discutindo a compra de duas linhas de negócios por uma mega empresa. Para que sua história não seja um caso único, o capítulo faz a contabilidade dos fatores que permitiram seu sucesso. Revisa então *Lean*, *Kanban*, *Scrum*, FDD e XP. Também, e fundamentalmente, o papel do MPS como a ferramenta de desenvolvimento organizacional que permitiu que realizasse este crescimento em tempo recorde e com mínimos inconvenientes.

---

---

## CAPÍTULO 2 - O MÉTODO DA MELHORIA CONTÍNUA

### 2.1 Melhoria de processos

Se concordamos com a literatura existente e a experiência pessoal dos autores, a melhoria de processos é a ferramenta que permite às organizações se entenderem profundamente, indo de um conhecimento intuitivo a um quantitativo, passando por etapas intermediárias que servem tanto para melhorar os resultados quanto para apoiar os passos seguintes. Uma história serve para deixar clara a diferença entre seguir e não seguir processos que foram acordados entre todas as partes. Os processos que as partes acordaram mudam quando elas assim decidem, enquanto que não seguir processos implica a inexistência de um padrão reconhecível pactuado entre os interessados.

Em uma organização desenvolvedora de *software*, Pedro, um dos melhores programadores, era um fervoroso defensor dos processos. Rapidamente convenceu seus colegas de equipe de trabalho a adotar processos na construção de diversos documentos e que, acima de tudo, definissem a sequência de passos e o critério de finalização, baseado na qualidade objetiva de cada produto. Os projetos nos quais Pedro tinha participação eram bem-sucedidos com mais frequência que todos os demais, e seus clientes elogiavam o produto gerado. Finalmente as diferenças chegaram aos ouvidos da direção e nosso jovem programador foi recebendo sucessivas promoções e, cada vez, para cargos de maior responsabilidade. Não havia passado muito tempo quando finalmente foi promovido a chefe técnico de desenvolvimento. Convocou seus até então colegas e mostrou que sua promoção obedecia ao reconhecimento que a alta gerência tinha do trabalho que seguia processos, por isso esperava que todos colaborassem com ele para ampliar sua utilização e contribuíssem para sua melhoria. Muitas cabeças assentiram e, depois de algumas perguntas e respostas, a reunião foi concluída. Só Pablo, um programador que, até a chegada de nosso herói era considerado o programador “estrela”, ficou para trás.

- “Pedro”, disse. *“Estou contente por sua nomeação, mas você bem sabe que não sigo processos de outros e acho que tentar que os demais entendam meus processos é uma perda de tempo, porque eles servem para mim e só para mim. Espero que não leve a mal, mas minha intenção é continuar fazendo o que sempre fiz.”*
- *“Não esperava outra coisa”,* disse Pedro.
- *“Que bom que aceite assim, fico feliz”,* respondeu Pablo.

Satisfeito, pegou suas anotações e se dirigiu à porta da sala. Pedro deixou que chegasse até a porta e o deteve:

- *“Bom, é importante que não fracasse. Que nunca fracasse. Os que seguem processos podem ter problemas, porque os problemas nos ensinam quais partes do processo precisam de mudanças. Mas se você não seguir processos, os fracassos não nos ensinam nada e você terá de arcar com toda a responsabilidade. Espero que não leve a mal, mas vou continuar fazendo assim, como sempre fiz”.*

Os processos que apresentamos a seguir nos permitem identificar defeitos desde cedo e detectar sua origem. Em certo sentido, seguir um processo é como comprar uma apólice de seguro: não queremos que nada ruim aconteça, mas investimos um pouco, caso algo ocorra. O mesmo acontece com os processos: se todos nós tivéssemos memória perfeita, nunca cometêssemos erros e as especificações na linguagem natural tivessem um significado único e inalterável, então se relativizaria muito a necessidade de seguir processos. Ainda seriam úteis para coordenar o trabalho, mas para pessoas com memória total seria suficiente fazer processos muito pequenos. A realidade é bem outra: os seres humanos erram, esquecem e mal interpretam as comunicações em linguagem natural. Como consequência, a única maneira de aprender como organização é capturar nossos processos para entender seu funcionamento (dados do processo) e a qualidade dos produtos que geram.

Nem todos os processos são iguais. Há processos administrativos que não são tratados neste livro. Há processos extraorganização que tampouco nos interessam. Os processos que queremos ressaltar e melhorar são aqueles vinculados ao desenvolvimento de *software* nas organizações. Ainda assim, é possível obter mais detalhes do que colocaremos neste livro em outras fontes. Os autores se limitarão a exibir o comportamento mínimo para organizar projetos que produzam sistemas de *software* de boa qualidade.

As organizações que entendem seus processos e tiram proveito são chamadas de “maduras”. Uma organização madura persegue seus objetivos com uso desse conhecimento. Sabe o que suas equipes são capazes de alcançar e não faz promessas que não pode cumprir. As equipes usam o conhecimento para iniciar o desenvolvimento com confiança em suas forças e em sua capacidade de cumprir compromissos. As organizações imaturas, entretanto, às vezes cumprem seus compromissos, mas nem sempre conseguem. Não conhecem sua capacidade e fazem promessas que nascem de seu desejo de ganhar o cliente. O que estamos propondo neste livro é um caminho para chegar à excelência, amadurecendo como organização e usando métodos ágeis, para o qual usaremos um caso de estudo e um modelo.

É o modelo MPS, em nosso caso, aquele que orienta a sequência de ações no que diz respeito ao crescimento da maturidade organizacional. O modelo MPS, que explicaremos com mais detalhes no Capítulo 4, é um modelo de desenvolvimento organizacional por níveis, que define os processos que a organização deve implementar e os resultados esperados dessa ação, para ir avançando de nível em nível. Mesmo quando todos os envolvidos têm a intenção de seguir o modelo, é impossível implementar todos os processos simultaneamente, mesmo se nos restringirmos a tomar os níveis um a um, coisa por outro lado muito saudável, porque os hábitos construídos em um são aplicados ao que o segue.

Existe a necessidade, portanto, de encontrar um método que nos ajude a organizar o crescimento em termos mais concretos do que faz o MPS e que, por sua vez, se compadeça das necessidades da organização em relação a seus negócios. Como o leitor pode imaginar, não há uma maneira única de fazer isto. Desta forma, decidimos antecipar a apresentação de um processo do MPS, não em detalhes, mas mostrando seu uso. Tomando emprestadas técnicas do MPS em seu processo Gerência de Decisões (GDE), vamos começar definindo o problema que estamos tentando resolver.

**Problema:** Apesar de que em um marco “macro”, os níveis do MPS servem para definir as pautas da melhoria contínua, em cada caso é necessário atender às necessidades da organização que pretende melhorar seus processos, levando em conta o seu “negócio”<sup>5</sup>. O problema então é encontrar um método de melhoria que brinde um bom balanço entre negócio e modelo.

**Atributos desejáveis de uma Solução:** A solução deve fornecer um mecanismo de melhoria contínua que permita identificar cada passo sucessivo de um programa de melhoria e este deve ter detalhes suficientes para que seja possível executá-lo sem muita ambiguidade, mas não tanto que implique um planejamento detalhado para cada seleção (DETALHE). Deve fornecer um marco teórico reconhecível que ajude a medir o impacto das decisões sobre o sistema como um todo, não só a otimização local (MARCO). Deve permitir deduzir as ações derivadas que otimizem o sistema (SISTEMA). Deve retroalimentar o mecanismo que permita introduzir melhorias a um bom ritmo, sem que interfiram com o trabalho nem com o desenvolvimento pessoal dos funcionários (NEGÓCIOS). Estes atributos desejáveis constituem os critérios sob os quais avaliaremos as alternativas de solução. A ordem de sua importância relativa é mostrada na Tabela 2.1 no final deste capítulo.

**Alternativas de Solução:** A literatura tem muitos exemplos destes métodos. Os seguintes foram incluídos em nosso conjunto de soluções: *Plan-Do-Check-Act* [SHEWHART, 1939], IDEAL [McFEELEY, 1996], *Focus-Improve-Sustain-Honor* [ARTHUR, 2004] e *Lean Simplified* [ARTHUR, 2006].

**Método de Avaliação:** Utilizaremos uma matriz de Pugh [PUGH, 1991] para avaliar alternativas quando os atributos são múltiplos. Usada por Pugh na General Motors e descrita por ele em seu livro já citado e previamente em um artigo [PUGH, 1981], a matriz de Pugh é um dos métodos de análise de alternativas mais difundidos entre engenheiros. Cada coluna representa uma solução e cada linha um atributo. Cada linha tem um peso específico que representa o valor relativo desse atributo frente aos outros. Em cada interseção linha/coluna, avalia-se a contribuição que a solução dessa coluna faz ao critério dessa linha. Previamente foi

---

<sup>5</sup> A referência a um negócio está entre aspas porque se trata dos motivos de existência da organização. Se essa organização é um hospital público que mede seu impacto na comunidade a partir da quantidade de curas que realiza ou do estado geral de saúde da população que atende, então para todos os efeitos este é seu “negócio”. Não se deve entender como se só pudesse ser aplicado a organizações com fins lucrativos.

estabelecido um mecanismo de avaliação que permite ajustar a objetividade a respeito da medição de cada atributo.

**Crítérios de Avaliação por Atributo:** DETALHE é mensurável como 3 (detalhe adequado), 2 (detalhe um pouco excessivo ou não suficiente), 1 (detalhe bastante excessivo, algo assim como trinta páginas para entendê-lo, ou muito superficial, algo que permite muitas interpretações) ou 0 (não tem nenhuma explicação clara associada). MARCO é medido como 3 (está reconhecido no sistema que é necessário atender outras coisas), 2 (não se analisa o sistema de maneira total, mas é bastante abrangente), 1 (há algumas pistas para analisar ações derivadas) ou 0 (não se apoia em nada na mudança sistêmica). NEGÓCIOS é medido como 3 (quando foca sobretudo nas necessidades do cliente como ponto de partida), 2 (há um alinhamento com o negócio, mas é externo ao método), 1 (pode-se alinhar ao negócio, mas o método é incerto e não é mencionado) ou 0 (não há nenhuma relação evidente entre o método e o negócio).

Descreveremos agora as quatro opções, tentando fazer com que o próprio leitor possa julgar a qualificação de cada uma em relação a cada atributo.

## 2.2 Plan-Do-Check-Act

*Plan-Do-Check-Act* (PDCA) foi proposto em [SHEWHART, 1939], e difundido sobretudo por Deming em várias ocasiões<sup>6</sup>. Deming se referia a este procedimento baseado no método científico como o “Ciclo de Shewhart”. A posteridade o lembra frequentemente como o “Ciclo de Deming”, uma das consequências da notoriedade deste que é ainda maior do que a daquele. Depois, Deming mudou o “*Check*” (verificar) por “*Study*” (estudar) para enfatizar que o passo deve ser mais de análise do que de inspeção.

Pode-se, justificadamente, considerar Shewhart como o pai da qualidade industrial. Foi ele quem introduziu os diagramas de controle estatístico para a análise da estabilidade de um processo a partir da medição de um atributo. Em função de sua data de origem, é difícil encontrar o material original, mas na maioria das citações<sup>7</sup> o primeiro passo é identificar o problema e depois analisá-lo. Não há uma menção explícita aos objetivos de negócio, embora seja difícil imaginar que Shewhart os tenha evitado, lendo seus outros materiais. Possivelmente o método foi tirado (mais uma vez) do contexto e ao fazer isso, perdeu-se parte de seu valor sistêmico. Portanto, sem julgar o método em si, mas julgando seu uso habitual, podemos dizer que PDCA é simples, fácil de aplicar, mas é possível que seja usado sem levar em conta o impacto no negócio. Há, em várias versões do método, referências a um processo desenvolvido por Deming para a melhoria contínua, o que daria uma melhor versão sistêmica desse método, assim como um possível vínculo com os objetivos de negócio. A seguir descrevemos os passos do PDCA.

**PLAN (Planejar) Passo 1: Identificar o Problema.** Atividades: Selecionar o problema a ser analisado; definir claramente o problema e redigir seu enunciado de forma precisa; fixar um objetivo mensurável para o esforço de resolução do problema; estabelecer um processo para a coordenação e conseguir a aprovação da direção.

**PLAN (Planejar) Passo 2: Analisar o Problema.** Atividades: Identificar os processos que impactam no problema e escolher um; listar os passos do processo como são executados neste momento; construir um mapa do processo; validar o mapa do processo; identificar possíveis causas do problema; juntar e analisar dados relacionados com o problema; verificar ou revisar o enunciado original do problema; identificar as causas-raiz do problema; juntar dados adicionais, se necessário, para verificar as causas-raiz.

**DO (Fazer) Passo 3: Desenvolver Soluções.** Atividades: Estabelecer critérios para escolher uma solução; gerar soluções potenciais que ataquem as causas-raiz do problema; escolher uma solução; conseguir aprovação e apoio para a solução escolhida; planejar a solução.

**DO (Fazer) Passo 4: Implementar a Solução.** Atividades: Implementar a solução escolhida em um piloto ou ensaio. Se o processo PDCA estiver sendo utilizado dentro do Processo de Melhoria Contínua, passar ao Passo 6 desse processo. Se estiver sendo utilizado de forma separada, continuar com o Passo 5.

<sup>6</sup> O livro [SHEWHART, 1939] foi compilado por Deming com um prefácio de sua autoria.

<sup>7</sup> Veja-se por exemplo <http://quality.enr.state.nc.us/tools/pdca.htm>

**CHECK (Verificar) Passo 5: Avaliar os Resultados.** *Atividades:* Juntar dados da solução; analisar os dados. O objetivo buscado foi alcançado? Se positivo, passar ao Passo 6. Se não, voltar ao Passo 1.

**ACT (Agir) Passo 6: Padronizar a Solução (e Capitalizar Novas Oportunidades).** *Atividades:* Identificar mudanças sistêmicas e necessidades de treinamento necessárias para uma implementação completa; adotar a solução; planejar e monitorar permanentemente a solução; continuar procurando oportunidades incrementais para refinar a solução; procurar novas oportunidades de melhoria.

O método PDCA é sólido, mas sua idade fez com que vários dos detalhes que seu autor defendia fossem deixados de lado em sua implementação corrente, que é o que um bom avaliador julga: seu uso, acima de sua definição. Isso não impede que, para o leitor avisado, os passos do PDCA continuem tendo utilidade. De fato, como veremos no que segue, o ciclo de Shewhart continua sendo utilizado em diferentes variantes. Tampouco é frequente que os usuários do PDCA o coloquem no marco adequado, simplesmente é utilizado como um ciclo cuja repetição produz os resultados esperados. Devemos lembrar que para Shewhart e, como consequência, para Deming, há um processo de melhoria contínua no qual o PDCA se encaixa. De outra maneira, perde-se parte de seu impacto. É nesse processo marco que várias das iniciativas sistêmicas e o vínculo com os objetivos de negócios estão imersos, de modo que mudar esse processo como foi definido e colocar outro em seu lugar pode ter como efeito a perda desse ambiente e, como consequência, a piora do processo de melhoria. Vejamos agora como McFeeley lida com esse problema, incorporando a seu método o detalhe necessário (de forma excessiva, segundo nosso ponto de vista).

### 2.3 O Método IDEAL

Por causa da enorme influência que Deming e, como consequência, Shewhart – a quem ele citava constantemente – têm sobre a comunidade de melhoria de processos, este método e os que descreveremos mais tarde estão muito influenciados pelo PDCA. A Figura 2.1 mostra uma descrição gráfica do método IDEAL. Ele tem cinco fases que correspondem a etapas importantes de uma iniciativa de melhoria de processos. Como a melhoria é contínua, espera-se que o ciclo continue depois de alcançada a 5ª fase. Mesmo que o autor do IDEAL [McFEELEY, 1996] esclareça que os limites entre fases são mais imprecisos do que os descritos na referência, é comum que as recomendações deste não sejam seguidas e acabem sendo executadas como uma sequência de atividades em um projeto, assim como outros desvios de alto impacto.

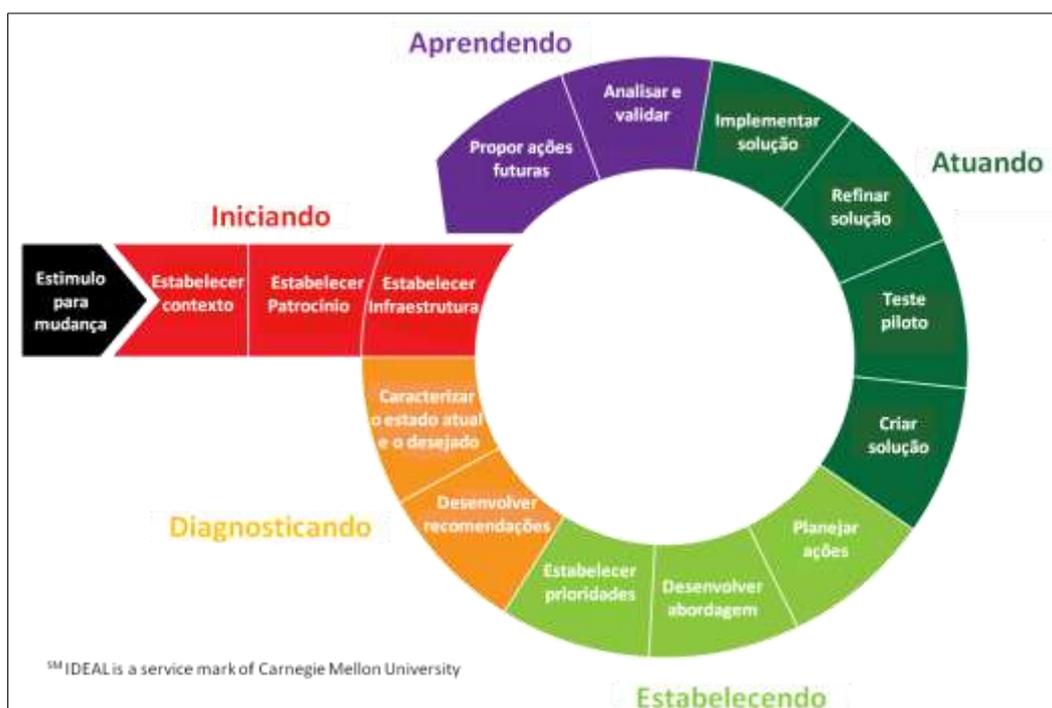


Figura 2.1: O Método IDEAL (adaptado de [McFEELEY, 1996])

A primeira fase é chamada, com certa lógica, de Iniciando (“*Initiating*”). Possui três blocos, mas na descrição detalhada estes não são considerados, sendo descritas no lugar 10 tarefas, algumas detalhadas até o nível de subtarefas. A mesma situação de desacoplamento entre a descrição textual e a gráfica é repetida em todas as fases. A fase Iniciando culmina quando a infraestrutura para a melhoria de processos foi construída e os objetivos de negócios foram vinculados ao programa de melhoria de processos, há um sistema de recompensas alinhado com as melhorias e um plano inicial para o projeto de melhorias, que contém o plano de comunicação dos avanços.

A fase seguinte é chamada de Diagnosticando (“*Diagnosing*”) e tem seis tarefas. Realmente, é a fase na qual se realiza a análise da lacuna existente entre as necessidades de processo e os processos em uso, tal como são realizados. O critério de entrada da fase não coincide totalmente com o critério de saída da primeira fase, por isso é difícil entender como se consegue chegar à sua execução. A fase tem como critério de finalização que as lacunas encontradas e as recomendações sejam entregues ao comitê de gestão, com a aceitação deste, assim como a necessidade de existir um esboço do plano estratégico de ação.

A terceira fase de IDEAL denomina-se Estabelecendo (“*Establishing*”), que aqui se refere ao plano e não aos processos. Apesar de ser confuso, cria uma boa sigla (IDEAL, por *Initiating, Diagnosing, Establishing, Acting e Learning*) que um nome mais lógico (Planejar ou Detalhar) não criaria. Nesta fase são ajustadas prioridades e formadas as equipes que levarão adiante a definição e difusão das mudanças e melhorias dos processos. O mais notável é que o método recomenda que o planejamento seja realizado pelo comitê de gestão, ou seja, são os gerentes que realizam a supervisão estratégica do plano de melhorias, e não o grupo de processos. Este excelente conselho é ignorado na prática. A fase tem 14 atividades. O critério de finalização é que o plano estratégico tenha sido completado e aprovado, e que a visão organizacional, o plano de negócios e o plano estratégico de melhoria de processos tenham sinergia entre si.

A fase seguinte é denominada Atuando (“*Acting*”). Esta fase é a mais interessante, embora o modelo tenha muitos componentes úteis, porque foca diretamente no negócio. É quando as melhorias são identificadas, construídas, aplicadas e colocadas em prática. Tem dez passos, dos quais destacaremos uma subtarefa do passo 2, Desenvolver Soluções: Analisar e Resolver o Problema. Ela nos interessa porque de muitas formas antecipa e se parece a *Lean*, no qual o enfoque está orientado puramente à dor e não à melhoria dos processos em si. Processos são modificados **porque** os atuais toleram defeitos e atrasos que não devem ser tolerados, as causas são atacadas, mas focando nos sintomas. Esta fase tem como critério de sucesso que a estratégia de realização e o plano sejam executados plenamente, ou estejam a ponto de serem iniciados. As soluções que se adotaram (ou testaram) foram documentadas corretamente, encontram-se sob o controle do grupo de processos, e está clara a forma de mantê-las. A melhoria de processos começou a ser institucionalizada na organização. Esta fase faz referência a muitas pequenas melhorias realizadas em paralelo, sob um único plano estratégico e múltiplos planos táticos. No entanto, a grande maioria das implementações de IDEAL sofre do mesmo defeito: um enorme plano tático que nunca fica pronto e que sofre da síndrome do correio: as cartas (pedidos de novas mudanças) continuam chegando e a tarefa de planejamento nunca é concluída.

A fase final, ou seja, a que inicia uma nova iteração do ciclo IDEAL, é denominada Aprendendo (“*Learning*”). Na realidade, é uma variante da primeira fase, já que teria pouco sentido começar de novo sem levar em conta o que já se avançou. Denomina-se assim porque, frente à alta gerência, é afirmado o patrocínio mediante a mostra dos avanços ocorridos pela aplicação do método.

Quando as organizações cometem os erros que ressaltamos antes, o resultado é que o projeto de melhorias tem pouco a mostrar. O exemplo mais comum é que são definidas soluções, mas não são implementadas nem em pilotos, o que é justificado pela síndrome de correio: já que foram aceitos novos pedidos de mudança, o grupo de melhorias focará na definição de soluções e não em sua implementação. Como consequência, uma longa lista de mudanças é lançada simultaneamente sem preparação suficiente, pela demanda de capacitação que isso acarreta, e o projeto fracassa. IDEAL não é um método ruim, mas é muito detalhado (é descrito em um documento de 236 páginas) e isso faz com que muita gente tenha tentado implementá-

lo sem ter lido todos esses detalhes<sup>8</sup>, com a consequente perda de vários elementos fundamentais, como o vínculo com os objetivos de negócio, o paralelismo na implementação<sup>9</sup> e a visão sistêmica (multicausal, com laços de retroalimentação e atrasos entre causa e efeito visível), que são indispensáveis para se estabelecer um programa de melhoria contínua.

O melhor do IDEAL é sua visão de melhoria de processos (Figura 2.2) baseada nos objetivos da organização e apoiada em alguns pilares: na visibilidade do projeto; na comunicação horizontal e vertical entre as partes; na captura, retenção e aproveitamento da experiência (lições aprendidas); e em uma rede de suporte de todo o projeto. Desse modo, os planos tático e estratégico são sustentados para se chegar ao sucesso.



Figura 2.2: Visão de Melhoria de Processos do IDEAL (adaptado de [McFEELEY, 1996])

#### 2.4 Focus-Improve-Sustain-Honor

*Focus-Improve-Sustain-Honor* (FISH) de [ARTHUR, 2004] é mais uma variante do PDCA, com ênfase nas ferramentas de Six-Sigma<sup>10</sup>. O ciclo de Arthur baseia-se na medição. O uso dos dados disponíveis e a busca estilo “inteligência de negócios” é o fundamento da análise, ao invés dos defeitos ou da distância a respeito de algum ideal. Isto, claro, não vai contra os preceitos do PDCA, mas influencia muito no impacto que tem cada um, porque em FISH é indispensável começar pela análise dos dados antes de entrar no ciclo. O ciclo tem quatro fases. A primeira fase, *Focus*, focar, que está baseada em um fato estatisticamente demonstrável pelas distribuições dos defeitos: uns poucos processos são responsáveis pela maioria dos defeitos. Encontrar essas “fábricas de defeitos” possibilita focar o processo de

<sup>8</sup> Se o leitor não se sente confortável com a afirmação de que um número muito grande de pessoas não lê os detalhes antes de implementar um modelo, os autores gostariam de remetê-lo ao trabalho de Royce, *Managing the Development of Large Software Systems* [ROYCE, 1970], que é considerado o pai do ciclo de vida em cascata por esse mesmo artigo, e que lamentavelmente está mal citado: o que Royce diz nesse trabalho é que essa visão do processo de desenvolvimento é infantil e cheia de problemas.

<sup>9</sup> Os usuários de IDEAL tendem a desenvolver um projeto sequencial com muitas iniciativas que atrasam a fase de aplicação, uma das maneiras de resistir à mudança.

<sup>10</sup> Six Sigma é uma estratégia de gestão criada na Motorola em 1986 e usada mundialmente (*SPC and TQM in Manufacturing and Services* [TENNANT, 2001]). Tenta melhorar a qualidade dos resultados dos processos identificando e eliminando as causas dos defeitos. Utiliza vários métodos, fundamentalmente estatísticos. O termo surgiu da análise estatística da ocorrência de defeitos na manufatura. A maturidade de um processo de fabricação pode ser expresso como a quantidade de desvios-padrão ( $\sigma$ ) que se distancia da média da população total, ou pela porcentagem de peças sem defeitos que produz. Um processo seis sigma ( $6\sigma$ ) é um que produz 99,99966% das peças sem defeitos, que foi o objetivo fixado pela Motorola e que deu nome aos métodos que foram juntados para alcançar este objetivo.

melhoria onde ele terá maior benefício. Para Arthur, a aplicação da lei de Pareto<sup>11</sup> prevê grandes lucros. Seu raciocínio é que se 80% dos defeitos são produzidos por 20% dos processos, aplicando-se novamente a regra temos que 64% (80% dos 80%) dos defeitos vem de 4% (20% do 20%) dos processos. Desse modo, focar a melhoria em um minúsculo grupo de processos permite eliminar um número bastante grande de defeitos. Por isso é necessário focar. A segunda fase, *Improve*, melhorar, é onde o defeito encontrado é eliminado. Esta é a fase onde são identificadas as causas profundas, são escolhidas as soluções possíveis e a implementação é definida e levada adiante. O MPS (ver o Capítulo 4) contém resultados esperados de processos que ajudam a entender a implementação dos passos de melhoria por identificação das causas-raiz. Utilizaremos ao longo deste livro esta capacidade de identificar as causas dos problemas para poder melhorá-las ou difundi-las<sup>12</sup>. Utilizar a análise da causa-raiz de forma sistemática é uma das ferramentas mais poderosas da melhoria de processos, e uma das três ferramentas intelectuais (junto com a análise e a gestão de riscos, e a visão sistêmica do processo) que mais rendimento consegue nos processos intelectuais que acompanham, ou devem acompanhar, a melhoria de processos.

A terceira fase, *Sustain*, sustentar, é onde devemos tentar consolidar e manter a melhoria, para a qual Arthur propõe utilizar ferramentas de “conhecimento profundo” como foram introduzidas por Shewhart e difundidas por Deming. Para começar, Arthur indica desenvolver o mapa do processo, utilizando sempre ferramentas simples. Em todos os casos, Arthur se inclina pela simplicidade, dedicando tempo ao processo intelectual e utilizando a ferramenta que melhor se adapta ao menor custo de aprendizagem possível. Por exemplo, neste caso ele sugere utilizar um diagrama de fluxo, sendo que há outras ferramentas possíveis<sup>13</sup> que são mais poderosas e igualmente difundidas.

Para poder afirmar que os resultados foram alcançados, é necessário saber se o processo está normalizado; porque, se não for assim, é impossível estabelecer comparações. Suponhamos que o problema é que uma receita culinária vem dando resultados diferentes. Ao analisar a causa profunda, percebemos que diferentes cozinheiros interpretam de forma distinta as instruções e que alguns passos estão faltando, porque o autor da receita assumiu que quem tentasse executá-la deveria ter conhecimentos culinários em grau suficiente, o que não resultou em uma previsão verdadeira. Além disso, há um erro na receita que sugere usar um tipo de farinha que não é o correto, digamos que diz “farinha” sem nada mais, que no contexto dos cozinheiros que têm problemas com a receita se entenda por “farinha de trigo”, quando quem criou a receita queria que fosse “farinha de milho”. A análise da causa detectou estes defeitos e foi sugerido que mudanças na receita sejam feitas, definindo com precisão os passos para que não haja diferentes interpretações, além de corrigir erros e ambiguidades.

Um grupo de processos sem a experiência suficiente consideraria que cumpriu com a obrigação: o processo, se executado corretamente, “deveria” funcionar. Jay Arthur, e os autores deste livro, não se conformam com essa definição, incompleta, da responsabilidade do grupo de processos: e se não funcionar? Lamentavelmente o resultado nesses casos é que o grupo de processos joga a responsabilidade aos que, devendo executar corretamente o processo, não o fazem, sem constatar que, necessariamente, são eles e não as mudanças que levam a perpetuar o problema ou introduzir outros novos.

Portanto, “sustentar” implica em medir e analisar os resultados. Isto obriga a entender quando, onde e o que medir. Um dos passos mais importantes na definição de processos e na mudança para a melhoria é identificar os processos-chave e os atributos a medir. Esta capacidade é exigida pelo MPS nos níveis mais altos de maturidade, a partir do Nível B; mas é uma das

---

<sup>11</sup> Pareto foi um estatístico francês do século XX que se destacou no estudo da distribuição da renda. Em 1906, ele fez a famosa observação de que vinte por cento da população possuía oitenta por cento da propriedade na Itália, posteriormente generalizada por Joseph M. Juran no princípio de Pareto (também conhecida como a regra do 80-20).

<sup>12</sup> Se o evento é um problema ou defeito, tentaremos localizar sua origem para eliminá-lo; mas se o evento é um resultado positivo não planejado, tentaremos entender o que o provocou para reproduzi-lo em outros ambientes.

<sup>13</sup> Os diagramas SADT e IDEF0, em sua versão norma internacional, são mais detalhados e de uso difundido. Também os diagramas de fluxo com pistas que permitem discernir responsabilidades. Da mesma forma, seria possível desenhar o processo seguindo técnicas de fluxo de dados da Análise Estruturada ou com ferramentas de UML.

qualidades mais valiosas de um gerente. Por exemplo, se nos preocupa que a maioria dos projetos termina depois da data prevista para sua entrega e realizamos mudanças em consequência disso para adiantar a produção de resultados, medir os atrasos que se produzem naqueles pontos é de suma importância. Medir só o resultado final, o desvio na data de entrega, é só parcialmente útil porque depois que entregamos atrasado, não se pode modificar o resultado. Arthur introduz aqui ferramentas de  $6\sigma$  que o MPS só exige a partir no Nível B, como já dissemos, e, portanto, complica um pouco além do necessário a análise de resultados.

A última fase do ciclo é *Honor*, honrar. Arthur se refere aqui à necessidade de respeitar e premiar os compromissos com a mudança, com a melhoria (nem toda mudança é uma melhoria, mas toda melhoria é uma mudança). Grande parte da mensagem sobre a melhoria de processos está contida na maneira que a organização ressalta e recompensa os esforços de seu pessoal em relação às mudanças e às melhorias. É importante destacar que nem todas as tentativas de melhoria, sobretudo nas etapas iniciais do processo de melhoria contínua, irão terminar igualmente bem-sucedidas. Algumas serão até negativas; mas é indispensável resgatá-las como esforços válidos, pois a organização aprende com elas.

## 2.5 *Lean Simplified*

O último método é *Lean Simplified* [ARTHUR, 2006]. Jay Arthur desenvolveu esse método como uma extensão de FISH, que vimos acima, com o propósito de tornar mais clara a aplicação deste, enfatizando a cadeia de valor que leva desde o pedido do cliente até sua satisfação pela entrega do produto. A palavra inglesa *Lean* significa enxuto, sem muita sobrecarga. Arthur escolhe chamá-lo *Simplified*, simplificado, porque reduziu a exigência estatística de seu método FISH. Chamaremos este método de LS, para evitar termos em inglês.

LS, como explica Jay Arthur em [ARTHUR, 2006], é um método para empresas de manufatura. No entanto, modificando ou deixando de lado o que não se aplica ao desenvolvimento de *software*, é um método poderoso para identificar e resolver problemas com o objetivo de realizar uma melhoria contínua. Apresentamos aqui nossa versão de LS adaptada à produção de *software*.

No coração de LS está o tema da velocidade de produção<sup>14</sup>. A velocidade não é pressa, é fazer melhor e sem interrupções, não é trabalhar os fins de semana ou até tarde à noite. A velocidade é produtividade posta a serviço do produto. Em um mundo conectado globalmente pela *Internet* os clientes esperam serviços quase imediatos sem perda de qualidade nem aumento de preços. O livro de [RODIN, 2010] é uma visão de como a demanda por serviços gratuitos entregues no momento e sem custo algum está revolucionando os negócios. Para as organizações que fabricam *software* o desafio está lançado: é preciso eliminar os defeitos e todas as demoras para entregar no prazo e com baixos custos.

Os atrasos não são justificáveis. O produto do *software* pode ser único e exclusivo, mas os processos pelos quais são produzidos não são. Cada projeto é composto pelas mesmas fases, que realizam as mesmas atividades. A resistência a esse conceito é notável em empresas de baixa maturidade e, no entanto, vemos várias vezes que a resistência a fazer as coisas de outra maneira é tão arraigada quanto a necessidade de sustentar que sempre são diferentes. A única que demora a mudar é a crença de que a organização está justificada em atuar como faz.

Em cada empresa há duas fábricas, a principal, que desenha, vende, fatura e entrega o produto, cuja fórmula é: Velocidade com Qualidade + Lucros = Benefício. A segunda é a fábrica de consertos, que corrige todos os erros que vão sendo cometidos à medida que o produto é desenhado, vendido e faturado. Há sempre uma fábrica de consertos visível, que é medida e controlada, mas há outra que é oculta, o que faz com que os defeitos sejam corrigidos sem que

---

<sup>14</sup> A empresa Toyota inventou o método de “produção enxuta” (*lean production*) tomando como referência os supermercados dos EUA; onde perceberam que, quando as prateleiras dos supermercados alcançam o ponto mínimo do inventário, são reabastecidas tão rapidamente quanto os clientes “tiram” os produtos da gôndola. Em um sistema de tração, o processo anterior deve sempre fazer o que o processo subsequente pede. Para ver que o estoque está baixo e, como consequência, repô-lo, coloca-se um cartão que marca o ponto exato. O nome em japonês desse cartão é *Kanban*, palavra que hoje identifica tanto o cartão quanto o sistema.

haja atribuição nem contabilização. Essa fábrica tem outra fórmula: Defeitos + Atrasos = Perdas.

No fundo, LS se concentra na velocidade da produção. A relação entre as etapas de um processo é fundamental. As etapas e atividades que não agregam valor devem ser eliminadas do processo. Por isso, a primeira atividade em LS é fazer um mapa da “cadeia de valor”, a sequência de atividades que vai da recepção do pedido do cliente à satisfação de suas necessidades<sup>15</sup>. Mais uma vez, o mapa precisa ser simples, mas não tanto que acabe sendo facilmente mal interpretado. Além disso, como se trata de um sistema de tração no qual a saída força o processo anterior e assim sucessivamente, este método atende principalmente a voz do cliente. Feito corretamente, isto gera valor para o cliente e, como consequência, para a organização.

### Foco nos Defeitos

Ao tentar reduzir o “atrito” que atrasa os processos, a Toyota descobriu que há muitas formas de desperdício (“*muda*”).

1. Excesso de produção (em *software*, incluir código que não foi solicitado “caso venhamos a necessitar”).
2. Inventário excessivo (em *software*, os processos que são gerados ao redor de funcionalidade não exigida, como testes-extra, volume de manuais, etc.).
3. Esperas (em *software* isso se manifesta especialmente quando é preciso esperar que o pessoal ou as instalações estejam disponíveis, ou quando o cliente precisa dar uma resposta).
4. Movimentos desnecessários do produto (em *software* a ubiquidade dos produtos em formato eletrônico faz disto um problema inexistente, mas se os planos são mantidos no papel ou nas lousas, isso pode chegar a ocorrer).
5. Movimentos desnecessários de pessoal (tipicamente na instalação ou na validação, frequentemente na etapa de gerar requisitos).
6. Processamento desnecessário ou inadequado (não é muito comum, mas em certas organizações burocráticas isso ocorre).
7. Defeitos que obrigam a reparações e retrabalho (não há porque explicar isto em nossa profissão).

Se a organização trabalha com prazos curtos e se concentra em manter a flexibilidade, são obtidos benefícios com qualidade e satisfação do cliente. No Capítulo 3 veremos como um grupo de desenvolvedores de *software* construiu métodos que permitem que estes conhecimentos sejam aproveitados. O movimento que iniciaram é conhecido como o *Agile Manifesto*<sup>16</sup> e marcou o desenvolvimento do *software* desde sua aparição.

LS continua com a organização do trabalho para eliminar o desperdício. São cinco as tarefas a realizar: *Sort*, ordenar; *Straighten*, endereçar; *Shine*, polir; *Standardize*, padronizar; e *Sustain*, manter. Damos aqui nossa interpretação dessas tarefas em atividades de engenharia de *software*. Ordenar significa decidir o que é útil e o que não é, e eliminá-lo. Esta é a tarefa das pessoas ou papéis que identificam as melhorias do processo. Frequentemente, os pedidos de mudança de processos são originados nas equipes, e um papel em particular, chamado garantia da qualidade é quem deveria detectar a necessidade de mudança e transmiti-la. Endereçar é colocar cada coisa em seu lugar e ter um lugar para cada coisa. Esse é o papel da gestão de configuração na engenharia de *software*. Polir é manter a área limpa para expor defeitos e perdas. Em *software* isto se manifesta na aplicação de formas e padrões que permitem a análise e a inspeção por terceiros, por exemplo o uso de padrões de programação que ajudam a ler um programa. Padronizar é definir sistemas, processos e procedimentos que ajudem a manter as três regras anteriores. Este é novamente o papel da área de melhoria de processos. Manter, por último, é conseguir que o fluxo de trabalho seja estável e as regras sejam respeitadas. Entre a área de melhoria de processos e o grupo de garantia da qualidade isto deveria ser realizável.

Outra das normas que regem a LS é a preeminência da demanda sobre a produção. Ao invés de produzir com antecipação ao que será exigido, só se produz a partir do que é pedido. Em

<sup>15</sup> Ouvir e reagir não é o mesmo que escutar e satisfazer.

<sup>16</sup> <http://www.agilemanifesto.org/>

nossa tradução ao mundo dos processos, isto significa que não tentaremos melhorar o que não se sente que não funciona, ou está com problema. O vocábulo inglês “*pull*”, que significa puxar, representa este pensamento, contra o vocábulo inglês “*push*” que significa empurrar. É comum na disciplina de melhoria de processos que se tente “empurrar” melhorias do centro para fora, ou de cima para baixo. Em nossa experiência, a resistência assim criada exige muito esforço e não justifica o retorno sobre o investimento. Pelo contrário, um enfoque de “puxar” faz com que a mudança seja vista como algo positivo já que, efetivamente, resolve um problema. Quando uma melhoria de processos é percebida como uma eliminação de um obstáculo à produtividade, ganha-se um aliado para as futuras mudanças que, além disso, agora conta com tempo extra para apoiar a criação e difusão de novas mudanças.

LS tem mais a contribuir, mas no essencial o exposto já alcança o objetivo de fazer entender as vantagens e desvantagens do método. Nossa matriz de Pugh para os quatro métodos fica assim:

atributos	peso	PDCA	IDEAL	FISH	LS
NEGÓCIO	5	1	3	3	3
DETALHE	4	1	1	2	3
SISTEMA	3	2	1	1	3
MARCO	2	2	0	0	3
soma		19	22	26	42

**Tabela 2.1: Seleção do Método de Melhoria de Processos**

Com estes valores fica evidente que nos decidimos pelo LS. Claro, pode-se criticar esta decisão, porque os valores colocados na interseção são arbitrários até certo ponto. Em uma decisão de maior impacto econômico, seria desejável que a pontuação estivesse melhor detalhada para conseguir maior objetividade.

Como vamos utilizar LS em nossas análises e nossas propostas para a empresa que tomamos como caso de estudo, é bom ressaltar alguns valores e crenças que se associam a LS.

1. O processo justo produzirá os resultados justos.
2. Desenvolver o pessoal e os fornecedores agrega valor.
3. A resolução contínua de problemas raiz conduz à aprendizagem organizacional.
4. O fluxo de uma peça aumenta a produtividade, o lucro e a qualidade.
5. Os produtos não gostam de fazer fila esperando atenção. Os materiais, peças e produtos são impacientes.
6. O único que agrega valor a um processo é a transformação física ou informacional da matéria-prima em algo que o cliente quer.
7. Os erros são oportunidades para a aprendizagem.
8. A resolução de problemas é 20% de ferramentas e 80% de uso da inteligência.

Nosso enfoque de melhoria de processos vai adotar muitas destas máximas aplicadas às áreas de desenvolvimento de *software*. Não vamos nos limitar a seguir o modelo MPS em sua aplicação, vamos procurar identificar e resolver os problemas que são comuns nas empresas desenvolvedoras de *software*.

## CAPÍTULO 3 - OS MÉTODOS ÁGEIS: *KANBAN*, *SCRUM*, *XP* E *FDD*

### 3.1 O que são os métodos ágeis?

No capítulo anterior, apresentamos a melhoria contínua de processos e decidimos tomar como referência a LS. As vantagens de um método leve, desprovido de burocracia, que enfoca diretamente as necessidades do negócio não passaram despercebidas para os “metodologistas” de Engenharia de *Software*. Já no século passado nasceram vários métodos de desenvolvimento que se apoiavam nas ideias de produção da Toyota, notavelmente o *Extreme Programming* [BECK, 2000], *Scrum* [SCHWABER, 2002], *DSDM* [STAPLETON, 1997], *Adaptive Software Development* [HIGHSMITH, 1999], *Crystal* [COCKBURN, 2005], *Feature-Driven Development* [COAD, 1999], *Pragmatic Programming* [HUNT, 1999] e outros. Em uma tentativa de encontrar formas comuns, dezessete destes criadores se reuniram em fevereiro de 2001 em um centro de esqui em Utah. O que surgiu foi um manifesto que marcou a engenharia de *software* de modo único, o *Agile Software Development Manifesto*<sup>17</sup>.

O conteúdo do manifesto pode ser lido *online*, mas consideramos que sua influência é tão importante, e suas coincidências com o método da *Toyota* TPS são tantas, que o incluímos aqui.

#### Manifesto para o Desenvolvimento Ágil de *Software*

**Estamos evidenciando maneiras melhores de desenvolver software, fazendo-o nós mesmos e ajudando outros a fazê-lo.**

**Por meio desse trabalho, passamos a valorizar:**

**Indivíduos e interação MAIS QUE processos e ferramentas**

**Software em funcionamento MAIS QUE documentação abrangente**

**Colaboração com o cliente MAIS QUE negociação de contratos**

**Responder a mudanças MAIS QUE seguir um plano**

**Ou seja, mesmo tendo valor os itens à direita, valorizamos mais os itens à esquerda.**

**(assinam) Kent Beck, Mike Beedle, Arie van Bennekum, Alistair Cockburn, Ward Cunningham, Martin Fowler, James Grenning, Jim Highsmith, Andrew Hunt, Ron Jeffries, Jon Kern, Brian Marick, Robert C. Martin, Steve Mellor, Ken Schwaber, Jeff Sutherland, Dave Thomas.**

© 2001, the above authors  
this declaration may be freely copied in any form, but only in its entirety through this notice.

Embora o manifesto descreva as ideias mais básicas, há entre os autores mais acordos que aqueles ali expostos. Muitas das coincidências vêm da mesma fonte: o foco na qualidade, com as regras da *Toyota* que já mencionamos no capítulo anterior, na seção “Foco nos Defeitos”. Assim como a *Toyota* tem seu método TPS (*Toyota Production System*), que é uma forma de “Kaizen”, os métodos ágeis se apoiam em períodos de produção curtos e muita colaboração dentro da equipe de projeto, apoiado na sinergia gerada pelo trabalho em equipe e na estrutura formada para alcançar as metas estabelecidas pela direção da empresa. Em grande parte, sua popularidade entre desenvolvedores deriva da independência que as equipes sentem ao tomar suas decisões em conjunto e bastante livres das redes burocráticas que são geradas nas grandes empresas, o que traz consigo resultados concretos, tanto qualitativos como quantitativos.

Ao deixar que a equipe tome as decisões para o próximo período de trabalho, chamado na gíria dos agilistas<sup>18</sup> um “*Sprint*”<sup>19</sup>, os métodos ágeis conseguem motivar a equipe dos projetos e comprometê-los com o resultado ao permitir que tomem decisões de peso. Pela curta duração

<sup>17</sup> <http://agilemanifesto.org/>

<sup>18</sup> Usaremos este neologismo para designar aqueles que aderem aos métodos ágeis e os praticam.

<sup>19</sup> Os dicionários definem “*Sprint*” como a maior velocidade alcançada por um participante em uma corrida, especialmente no final. As corridas de até 200 metros são consideradas todas *Sprints*, do princípio ao fim. Por analogia, cada parte de um projeto ágil é considerado um *Sprint*.

de um *Sprint*, normalmente menos de quatro semanas, as equipes podem ver seus resultados de imediato. Também é importante a participação do cliente: junto a um representante dele, que deve estar comprometido, por sua vez, com o resultado, define-se o objetivo de cada *Sprint*. É um dever das equipes ágeis definir uma parte do produto que, em si mesmo, tenha valor para a organização do cliente. Desse modo, o produto parcial é concreto e mantém a concentração e o foco no negócio.

Os métodos ágeis nasceram como resposta às burocracias que ignoram as particularidades do desenvolvimento de *software* e, em sua ignorância, pressionam as equipes para levar adiante projetos com compromissos irracionais assumidos de forma indevida. Ao colocar metas curtas e priorizar a participação do cliente nas decisões de negócio, além de colocar um freio concreto às mudanças arbitrárias, os métodos ágeis resgatam o valor da engenharia de *software* frente aos embates burocráticos de certos níveis nas corporações. Entre outras virtudes, a decisão pela equipe é visível para todos: DEVE ser visível para todos. Em consequência, a transparência dos projetos ágeis é um de seus atributos mais valiosos e mais revolucionários.

Os agilistas se consideram um movimento pró-métodos. Os que acreditam que a agilidade é contrária aos processos e às ferramentas, à documentação ou aos planos, estão equivocados. O que procuram é que estas entidades estejam a serviço das atividades da equipe e não ao contrário. Se o leitor acha que ser ágil é não planejar, não seguir processos por mais ligeiros que sejam, não ter mais ferramentas que o ambiente de desenvolvimento interativo e algumas linguagens e não documentar as decisões, está totalmente equivocado. Quem pensa assim é um *hacker*, não um agilista. Os agilistas pensam que o planejamento detalhado não pode levar mais que umas poucas horas e deve envolver a equipe, que os processos são flexíveis, mas devem ter o acordo de todos aqueles que os colocam em prática, que a documentação deve ser fácil de manter e responder a uma necessidade orgânica do projeto e deve ser feita quando essa necessidade se manifesta e não como uma condição contratual depois que o produto estiver concluído. Quanto às ferramentas, basta recordar que a integração contínua, um dos alicerces dos métodos ágeis, requer excelentes ferramentas de gestão de configuração com testes integrados, portanto é claro que os agilistas trabalham em prol da eficiência e não contra ela.

Os quatro métodos ágeis que consideramos mais úteis em diferentes etapas de uma empresa são *Kanban*<sup>20,21</sup>, *Scrum*<sup>22</sup>, XP<sup>23</sup> e FDD<sup>24</sup>. A ordem na qual os descreveremos vai do mais simples (*Kanban*) ao mais complexo (FDD). *Scrum* e XP se apoiam em *Kanban* e, em particular, descreveremos XP em sua versão XBREED, que mistura os conceitos de XP com as práticas de *Scrum* para obter um processo mais sólido e confiável.

### 3.2 *Kanban*: medindo o fluxo

Quem introduziu *Kanban* como método ágil foi [ANDERSON, 2011]. O método *Kanban* é parte de uma família de sistemas que se denominam “*pull*”, ou de puxar, contra o enfoque tradicional de sistemas “*push*” ou de empurrar. Outra maneira de ver a diferença entre uns e outros sistemas é que o sistema “*pull*” é guiado pela demanda enquanto que o sistema “*push*” é guiado pela produção. Em um sistema “*pull*” o processo espera que haja demanda de seu produto para iniciar a produção, tentando fazer com que ele chegue bem a tempo, de maneira que não haja estoque<sup>25</sup>. O estoque é característico dos sistemas “*push*”, já que é o amortecedor que permite o desacoplamento entre processos consecutivos. O problema é que o estoque consome muito capital e tem um alto custo, sendo que, além disso, não se sabe se o produto final tem comprador ou não. Desse modo, muito trabalho e material é desperdiçado.

O método *Kanban* permite alcançar um ritmo de produção sustentável e introduzir mudanças nos processos com baixíssima resistência. É por isso que o usamos preferencialmente

---

<sup>20</sup> [KNIBERG e SKARIN, 2010]

<sup>21</sup> Quando usamos *Kanban* com maiúscula nos referimos ao método *Kanban* desenvolvido por Anderson, quando é usado em minúscula, *kanban*, faz referência a qualquer outro uso, como o sistema *kanban* da *Toyota* ou a tabela *kanban* que veremos mais adiante.

<sup>22</sup> [KNIBERG, 2007]

<sup>23</sup> [KNIBERG, 2007]

<sup>24</sup> [PALMER e FELSING, 2002]

<sup>25</sup> Isto também é conhecido como sistema “*Just in Time*” ou com as siglas JIT.

naquelas organizações de baixa maturidade institucional. Como vimos, o método *Kanban* é um dos mecanismos que suportam o TPS<sup>26</sup>, mas a adaptação para engenharia de *software* é de 2004 e foi realizada por Anderson na *Microsoft*. Anderson o apresentou na conferência *Agile 2007* de Washington e, desde então, começou o entusiasmo pelo método, já que os resultados eram muito animadores.

O método é, em si, muito simples, mas ao usá-lo de determinada maneira, traz consigo múltiplas vantagens que mostraremos ao explicá-lo. Apesar de que o texto [ANDERSON, 2011] é a base que permite entendê-lo profundamente, para o leitor que busca uma gestão mais pragmática aconselhamos [KNIBERG e SKARIN, 2010] que remete ao essencial da implementação, sem deixar de lado o anedótico que permite a compreensão, ou como se diz no México, a “aterrissagem” do material.

Um elemento central no método é o uso do painel *kanban*. Não se deve confundir o uso do painel com a aplicação do método; é possível usar o painel sem segui-lo, como de fato se faz em muitas adaptações de *Scrum* e XP, porque o painel é um excelente meio para comunicar o progresso de um projeto.

O painel *kanban* é simplesmente um registro do avanço do projeto materializado segundo a conveniência da equipe. O mais frequente é usar um painel onde se possa colar notas autoadesivas, como na Figura 3.1, dividida em colunas verticais. Cada coluna indica um estado das tarefas. As notas autoadesivas contêm os nomes das tarefas. A primeira coluna da esquerda tipicamente contém o “pendente”, quer dizer, a lista das tarefas que devem ser feitas e que ainda não foram iniciadas. Quando um membro da equipe tem disponibilidade para começar uma delas, pega da primeira coluna a tarefa correspondente, seja por prioridade pré-estabelecida ou por alguma outra razão que a equipe tenha decidido, e a passa para a coluna seguinte à direita. Em alguns métodos que usam o painel *kanban*, o membro da equipe que faz isto coloca a data e hora do início, seu nome e data estimada de finalização nos cantos da nota, desenhados para esse uso, como é mostrado na Figura 3.2<sup>27</sup>. Quando termina de realizar sua tarefa, o membro da equipe retira a nota do lugar onde a colocou e a move para a próxima coluna à direita, onde por sua vez a mesma será tomada por outra pessoa que dará continuidade ao processo até que a tarefa chegue ao ponto onde foi combinado que será considerada completa, ponto no qual chega à coluna da extrema direita do painel.

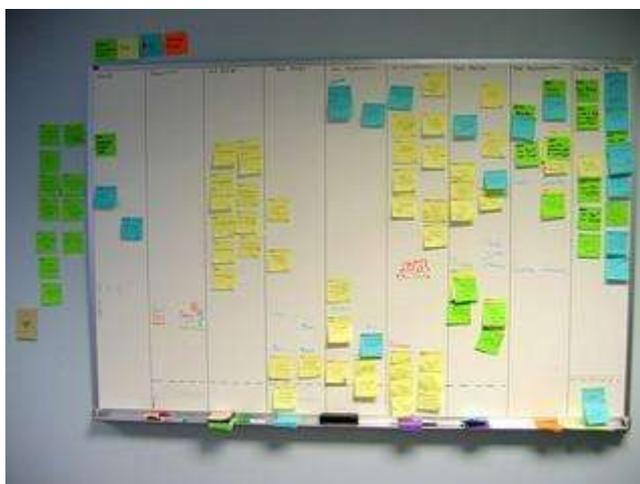


Figura 3.1: Painel *kanban*

Não há nenhum motivo especial para utilizar cartões autoadesivos ou as lousas em si. Podem ser utilizados papelões nos quais são presos cartões de cartolina, podem ser usadas filas horizontais ao invés de colunas ou pode ser utilizado um painel virtual dos muitos que são

<sup>26</sup> *Toyota Production System*.

<sup>27</sup> No exemplo mostrado, no canto superior esquerdo está o nome da pessoa responsável, no superior direito, a data e hora de abertura do item, no inferior direito a estimativa de finalização e no inferior esquerdo (vazio no exemplo) a data real de finalização. Quando se usa esta convenção frequentemente as notas autoadesivas são copiadas e coladas umas sobre as outras para ter a história de seu desenvolvimento.

oferecidos pela *internet*. O objetivo é o mesmo: dar clareza às tarefas pendentes de resolução e entender tanto o estado atual do projeto quanto a ocupação do pessoal.

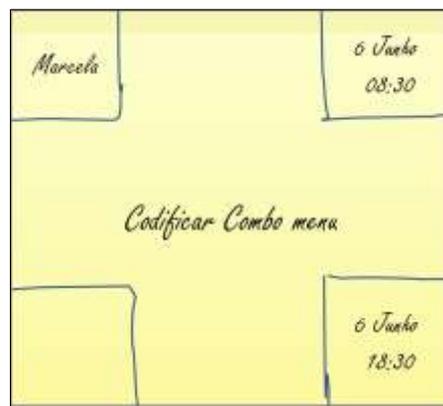


Figura 3.2: Nota do Painel *kanban*

No método *Kanban*, há um limite para o número de tarefas em cada coluna. O objetivo é identificar claramente a capacidade do sistema e equilibrar a demanda em relação à competência da equipe. O método *Kanban* permite compreender o tempo de circulação de cada tarefa, desde o ingresso no sistema pela esquerda, até a finalização na coluna da direita (há algo de satisfação pessoal para cada membro da equipe em mover a tarefa para a coluna “concluído” que motiva a usar o *kanban*). Depois que se ajustou os parâmetros de produção, a equipe alcança um ritmo sustentável, quer dizer, que pode ser mantido indefinidamente, conseguindo na verdade uma certa previsibilidade que outros métodos demoram muito para alcançar.

Como essa regularidade se torna presente muito rapidamente, toda obstrução a essa regularidade é rapidamente visível, potencializando a equipe para detectá-las e, em consequência, resolvê-las. O impacto que tem na regularidade os defeitos, as demoras, os gargalos e a má estimativa do tamanho e da complexidade do produto aparecem rapidamente no painel. É fácil relacionar estes problemas com o custo do projeto, pois ao restringir o número de tarefas em cada coluna as pessoas devem atender os gargalos para ajudar a resolvê-los. Desta maneira, o método *Kanban* vincula rapidamente os problemas técnicos do projeto ao resultado de negócio, sem precisar estabelecer complicados mecanismos de análise. Este é um resultado que não se vislumbrava ao introduzir o método, mas que é um dos pilares de sua adoção.

Uma das vantagens do *Kanban* é que, ao produzir com qualidade e tempo, gera-se confiança nos clientes, que recebem produtos com regularidade e com a qualidade esperada. Outra vantagem é que, ao fazer com que a equipe melhore constantemente seus processos para evitar demoras, as entregas são feitas com mais frequência e englobando maior número de funcionalidades. Por sua vez, esta situação faz com que a equipe se sinta mais à vontade e se dedique com mais afinco a melhorar o fluxo de trabalho.

Para implementar *Kanban* o primeiro passo é identificar o fluxo de trabalho, o que se conhece como a “cadeia de valor” que já tínhamos visto no capítulo anterior, nas discussões sobre o método de melhoria de processos a ser empregado. Pela origem comum desses métodos (as diferentes versões de qualidade total) e o fato de que o método *Kanban* é uma adaptação ao desenvolvimento de *software* de uma técnica com o mesmo nome (*kanban*) derivada do TPS, por sua vez da mesma origem que as anteriores, as coincidências não devem nos surpreender. *Kanban* usa cinco princípios para criar comportamentos nas organizações onde é aplicado: Visualizar o Fluxo de Trabalho; Limitar o Trabalho em Realização; Medir e Administrar o Fluxo; Explicitar Políticas de Processo; e Utilizar Modelos<sup>28</sup> para Reconhecer Oportunidades de Melhorias.

O primeiro ponto a destacar na construção de um painel *kanban* para o fluxo de trabalho é que a coluna da direita deve corresponder ao estado de tarefa “pronta”. Antes que se possa

<sup>28</sup> Os modelos a que Anderson faz referência são mais gerais que os que apresentaremos no próximo capítulo; mas, dada a abertura que sugere o texto já citado, os autores não acharam contraditório utilizar o MPS para introduzir melhorias de processo compatíveis com *Kanban*.

prosseguir com a implementação do método, é imprescindível que a equipe, junto com o fornecedor de informação (mais adiante, chamaremos este papel de “Dono do Produto” seguindo o costume introduzido por [SCHWABER e BEEDLE, 2002]), defina os atributos necessários de uma tarefa para que seja considerada “pronta”. Por exemplo, a densidade de defeitos remanescentes conhecidos, ou os estados pelos quais teve de passar (inspeções, teste unitários, etc.).

A coluna da direita está assim bem identificada e seu sentido é claro. A coluna da esquerda é onde se colocam as tarefas pendentes. No meio, a quantidade de colunas deverá ser estabelecida de acordo com o que a equipe quiser, desde que tenham significado para ela. Por exemplo, pode ser que a coluna seguinte à direita de “Pendentes” seja “Em análise”, a que segue esta seja “Analisado”, a seguinte “Em Desenvolvimento”, a seguinte “Pronto para Teste”, a seguinte “Pronto para Integração”, e assim sucessivamente. Por outro lado, pode ser que uma equipe determine que tudo o que necessita saber está contido em três colunas: “Pendente”, “Em Desenvolvimento”, “Pronto para Entregar”. As decisões que assim são tomadas têm repercussões muito grandes. Por exemplo, a primeira escolha sugere que, dentro da equipe, há especialidades que vão passando a tarefa de uma à outra. São abertas pelos analistas, que a passam aos desenvolvedores, que a passam aos testadores. A segunda sugere que a equipe trabalha em células integradas, onde todos esses papéis são cumpridos dentro da célula. Um caso extremo pouco recomendável é o da célula unipessoal.

Recomendamos a adoção de um painel mais complexo, com divisões técnicas do trabalho, pelo menos enquanto não se incorporarem técnicas especiais, como programação por duplas e *design* dirigido por testes. O motivo é que as diferentes etapas dentro do processo permitem visualizar melhor os estados intermediários, de modo que os atrasos potenciais e os gargalos possam ser rapidamente detectados e a duração das tarefas, melhor estimadas. Além disso, este esquema de trabalho é muito flexível e permite crescer com facilidade para outros métodos, em particular o *Feature Driven Development*, que recomendamos mais adiante como um método vantajoso para projetos grandes com equipes numerosas. Se estas duas características não forem suficientes, outra vantagem consiste em que os estados intermediários que são gerados ao acontecer um repasse<sup>29</sup> permitem que o projeto conte com o apoio de grupos organizacionais, como Gestão da Qualidade, que podem tomar essas transferências como referências e intervir sem violência na qualidade do processo.

Até aqui foram descritas generalidades do uso de um painel *kanban*. Para que seja uma aplicação do método *Kanban*, é necessário limitar o número de notas em cada coluna. Se em uma coluna há tantas notas quanto indica o limite, geralmente anotado no alto de cada coluna junto a seu nome, não se pode passar uma nota a mais para essa coluna. Isto implica que, embora o passo anterior tenha sido terminado para uma tarefa, a coluna está bloqueada e não é possível avançar a nota correspondente. Claramente as pessoas que ficam ociosas notam isto, as pessoas que estão trabalhando nas atividades da coluna saturada também percebem isto, e a cadeia de produção fica detida. Nesta situação, é detectado um gargalo, que deve ser resolvido pelos próprios membros da equipe. Ao contrário da grande maioria dos métodos existentes, *Kanban* não é prescritivo. Tal característica também é seguida pelo *Scrum*, mas o *Kanban* é ainda menos prescritivo do que o *Scrum*. A equipe escolhe, adapta e adota seus processos segundo suas necessidades.

O que diferencia notavelmente o *Kanban* dos outros métodos ágeis é sua flexibilidade; mas, sobretudo, é a limitação do volume de trabalho em cada etapa. É esta restrição que coloca em jogo os mecanismos de adaptação e, em consequência, os mecanismos de melhoria que, em outros métodos, ficam a cargo de reuniões especiais chamadas “retrospectivas”. O que nos demais métodos é uma visão do que aconteceu, no *Kanban* é a necessidade lógica de operar sobre o que está acontecendo.

Seguindo nossa opção de melhoria de processos, que definimos no capítulo anterior, utilizaremos os mesmos preceitos que Anderson usa para *Kanban*, pois são compatíveis: Foco na Qualidade, Redução do Trabalho em Desenvolvimento, Entregas Frequentes, Equilíbrio da Demanda contra a Produção, Fixação de Prioridades e Ataque às Fontes de Variação para Melhorar a Previsibilidade. A receita de Anderson não só é compatível com a de LS, que

---

<sup>29</sup> Em inglês, “*hand-off*” entre um papel e o outro.

escolhemos anteriormente: é totalmente compatível com o MPS! No Capítulo 5 expandiremos as técnicas de *Kanban* utilizando o exemplo da Tahini-Tahini.

### 3.3 *Scrum*: Organizando o sistema para um estado de equilíbrio orgânico

*Scrum* não deve ser considerado um método, apesar de ter regras claras que devem ser seguidas, porque deixa muitas questões abertas para resolução da equipe do projeto. Ao descrever o *Kanban* dissemos que, depois dele, *Scrum* é o enfoque ágil menos prescritivo. Isto é certo, mas a grande diferença entre o número de regras de um e outro (3 contra mais de 10) faz com que estejam próximos, mas não muito.

Para implementar *Scrum* em uma organização é preciso antes realizar mudanças profundas. Com *Kanban* as mudanças originais são apenas três: refletir o fluxo em um painel *kanban*, limitar o número de tarefas por etapa e melhorar o fluxo total fazendo as alterações exigidas pelo processo. *Kanban* se adapta facilmente a qualquer processo subjacente, porque as entregas rápidas podem ser internas à equipe e a participação dos envolvidos externos é desejável, mas não indispensável. Por outro lado, em *Scrum* é indispensável reestruturar a organização em vários sentidos: primeiro é preciso contar com uma pessoa que conheça o produto, ou tenha a visão do produto, e que esteja disponível para trabalhar com a equipe durante a duração do projeto. A dedicação exigida não é de tempo integral, mas esta pessoa deve permanecer acessível para que a equipe possa tomar decisões relacionadas ao negócio conjuntamente com ela. Da mesma forma, esta pessoa deve possuir autoridade suficiente para que suas decisões não sejam revistas<sup>30</sup>. Na linguagem de *Scrum* esta pessoa é conhecida como o “Dono do Produto”.

Em segundo lugar, o pessoal deve ser dividido em equipes interdisciplinares pequenas, auto-organizadas, que contem com a supervisão e colaboração para facilitar a resolução de problemas por um especialista, chamado em *Scrum* de *Scrum Master*. O *Scrum Master* se encarrega de que os processos do *Scrum* sejam seguidos e de manter a relação da equipe com o meio que a rodeia. Nesse sentido, é muito mais um cão-pastor que cuida do gado do que um gerente que dirige as operações. A auto-organização da equipe é fundamental no *Scrum*. O *Scrum Master* não dita o que deve ser feito, mas facilita o caminho para que possa ser feito. Desse modo, é a própria equipe que fixa as regras de colaboração e produção, e estas são flexíveis e modificáveis. Esta é a base para que não existam filas de espera na equipe e que, quando forem abertas oportunidades de trabalhar juntos, estas sejam aproveitadas em benefício de melhor produtividade e qualidade.

Terceiro, o trabalho a realizar, os requisitos, devem ser divididos em um conjunto de entregas concretas e pequenas, não uma funcionalidade extensa, mas pequenas parcelas de trabalho que tenham sentido para o negócio e possam ser organizadas de acordo com sua prioridade, fixada em conjunto pelo usuário Dono do Produto e o *Scrum Master*. Objetivos são fixados, chamados de *release*, que estabelecem prazos de longa duração (meses) para a entrega do produto completo. Depois, o tempo de duração do projeto é parcelado em pequenos marcos, chamados *Sprints*. Cada *Sprint* terá uma duração fixa, usualmente entre 1 e 4 semanas. As equipes de trabalho escolherão da lista de requisitos priorizados (*Backlog*), aqueles que entram no *Sprint* (*Sprint Backlog*). A equipe investe um dia de trabalho no planejamento do *Sprint*. Há regras claras sobre como planejar e vários métodos que foram experimentados e adotados por muitas equipes. Fundamentalmente, a estimativa se baseia na história de trabalho da equipe e na velocidade com que se constrói em geral o produto.

A equipe se reúne todos os dias por um período curto, não mais do que quinze minutos, para revisar as tarefas do dia anterior e o plano do dia. O *Scrum Master* dirige e facilita esta reunião. Estas reuniões são denominadas *Scrum* e dão nome ao método.

O objetivo de um *Sprint* é entregar uma parte do produto ao término do *Sprint*, o que se manifesta por uma demonstração feita ao cliente. O fragmento de funcionalidade que se entrega ao cliente se chama “*sashimi*”, por analogia com o prato de peixe cru japonês em que cada pedaço é um prato em si mesmo, mas também um componente do prato total. Cada demonstração permite ao cliente entender melhor o produto que pediu e fazer os ajustes

---

<sup>30</sup> Isto é o que [BOEHM e TURNER, 2003] chamam de um usuário “CRACK” (*collaborative, representative, authorized, committed and knowledgeable*) que traduzido é colaborativo, representativo, autorizado, comprometido e com conhecimento.

necessários para obter o melhor retorno do investimento no menor prazo possível, mudando as prioridades do *Backlog* se isto for útil.

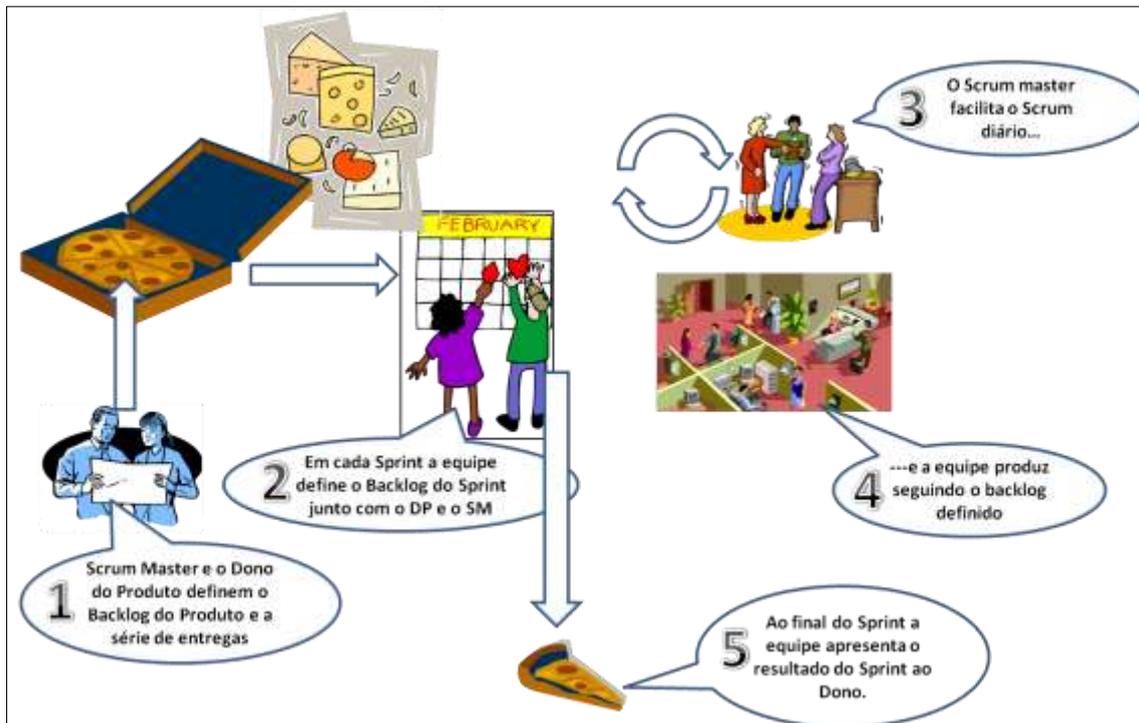


Figura 3.3: Processo Scrum

Como há muita liberdade para realizar as tarefas durante um *Sprint*, é possível que tenhamos que realizar ajustes. No geral, as mudanças nos processos são introduzidas entre o término de um *Sprint* e o início do seguinte. Para decidir sobre as mudanças, a equipe realiza reuniões chamadas “retrospectivas” para analisar o comportamento dos processos e tentar melhorá-lo.

### Momentos da Verdade em um Scrum

Há muitas oportunidades de errar tentando implementar o *Scrum*. Chamamos esses pontos críticos de “momentos da verdade”, por analogia com o mesmo conceito em *Marketing*.

O primeiro sintoma de que não devemos implementar *Scrum* é o do Dono Desconhecido. Isto aparece quando se inicia o projeto usando as ferramentas tradicionais de planejamento e não se identifica o Dono do Produto. Isto tem consequências imediatas. Ao não ter o guia de um usuário CRACK<sup>31</sup>, o projeto sofrerá com falta de visão do produto, falta de clareza nas prioridades, má condução do “jogo do planejamento” ao iniciar os *Sprints*, que serão definidos com falta de objetivos claros, contribuindo para que sejam solicitadas mudanças de requisitos no meio dos *Sprints*. Nesse caso fica desvirtuada a validade do *Scrum*, motivo pelo qual é inaceitável continuar utilizando-o no projeto. É preciso voltar a métodos tradicionais, como o ciclo em cascata ou usar outros, como o *Rational Unified Process*. Se isto não for o desejado, é preciso esperar para começar o projeto quando já houver um Dono do Produto. Se isto for impossível, mas existir a possibilidade de gerar um “sósia” interno na empresa, sempre fora da equipe, esta se torna uma solução aceitável. NUNCA um membro da equipe pode ser o Dono do Produto, já que isto cria conflitos de interesses.

Mesmo quando há um Dono do Produto, este pode sofrer de diferentes problemas: falta de visão do produto, prioridades pouco claras, ser muito inflexível ou ter os objetivos pouco claros. Em todos os casos é desejável tentar educar o Dono do Produto para conseguir o comportamento desejado, ou se isto não for possível, mudar de Dono do Produto. Sem um Dono do Produto CRACK, a equipe fica à deriva e a qualidade do produto fica comprometida.

<sup>31</sup> Usuário “CRACK” (*collaborative, representative, authorized, committed and knowledgeable*) que traduzido é colaborativo, representativo, autorizado, comprometido e com conhecimento [BOEHM e TURNER, 2003].

Não é só o Dono do Produto que pode ser a causa do fracasso do projeto. Também é necessário que o *Scrum Master* conheça os processos e saiba como atuar em cada caso. O *Scrum Master* é o responsável pela escolha correta do jogo do planejamento, evitando problemas quando faltar história à equipe. Apesar de que o Dono do Produto e o *Scrum Master* têm participação nesse jogo, suas intervenções são limitadas. O Dono do Produto pode pedir explicações ou explicar o que está esperando como resultado, mas não pode fixar o valor das estimativas. O *Scrum Master* facilita a reunião, podendo pedir precisões (por exemplo, a definição de “pronto”), mas não deve influenciar na escolha de valores. De qualquer modo, é útil lembrar que a estimativa da equipe é sobre “tempo ideal”, quer dizer, sem interrupções nem outras tarefas. O *Scrum Master* deve ajustar esse tempo ideal ao “tempo real”, multiplicando pelo fator correspondente.

O *Scrum Master* é responsável pela adoção, desde cedo, da definição correta de quando um produto está “pronto”. Um *Scrum Master* muito flexível pode ceder a pressões do Dono do Produto, como mudanças no meio de um *Sprint*, que não são benéficas para ninguém. Um *Scrum Master* muito rígido pode degenerar a direção dos *Scrums*, tratando-os realmente como reuniões de avanço. O *Scrum Master* precisa impulsionar, da mesma forma, a melhoria de processos, por meio de retrospectivas frequentes ou quando a ocasião exigir. No começo das atividades com *Scrum*, é muito possível que se escolham as durações dos *Sprints* muito longas ou muito curtas. Isto não é problema, pois a duração dos *Sprints* pode ser ajustada durante uma retrospectiva.

Um problema comum é entender “ágil” como codificar e consertar. Não há sequer a mínima documentação, faltam métodos de engenharia de *software* na confecção de modelos ou no teste dos programas. Todas as boas práticas aprendidas são esquecidas ao adotar “ágil”. Estes defeitos estão associados à declaração que se adotou “algo como” *Scrum*, mas, ao contrário do que ocorre no *Scrum*, não é conhecido realmente o *status* de um requisito individual. Se a organização mente para si mesma sobre o que constitui adotar *Scrum*, a estrutura inflexível é substituída pelo caos, perde-se o controle do projeto, as iterações são de qualidade muito baixa e o sistema é frágil frente às mudanças.

Outra crítica, com fundamento, do método *Scrum* é que este não faz referência a métodos ou técnicas de engenharia utilizadas no meio do *Sprint*. É claro, isto é assim pela concepção do *scrum*, mas continua sendo certo que devem ser adotadas ferramentas de engenharia de *software* que apoiem as tarefas da equipe. Uma equipe ágil reconhece táticas ágeis, como *Test Driven Development* (TDD), *refactoring*, requisitos iterados e outras igualmente úteis e as aplica. Em seguida veremos algumas destas técnicas, de uso comum em equipes de *Scrum*.

### 3.4 XP: Inspeções, *Design*, Cooperação e Muitas Provas

*Extreme Programming*, normalmente conhecida como XP, é um conjunto de técnicas condensadas da engenharia de *software* tradicional, adaptadas para seu uso em equipes pequenas que repetem rapidamente seus ciclos de desenvolvimento. De fato, Kent Beck<sup>32</sup>, o pioneiro de XP, foi um dos primeiros a sugerir iterações curtas com participação intensa do usuário durante elas. No livro citado, Beck explica sua posição “extrema”, dizendo que se as técnicas da engenharia de *software* são boas, realizá-las todo o tempo é ainda melhor.

XP se apoia em cinco valores: Comunicação, Simplicidade, Retroalimentação, Coragem e Respeito. Desses cinco valores, Beck deriva cinco princípios, mais práticos e próximos à produção que os valores: Retroalimentação rápida, para acelerar a aprendizagem; Busca contínua pela Simplicidade, para procurar a solução mais simples; Mudança Incremental, para reduzir o impacto da mudança nas organizações; Estar aberto à Mudança, contrariamente a tentar controlá-la; e Trabalho com Qualidade, para que tanto o cliente quanto os desenvolvedores se sintam gratificados pela experiência.

XP não tenta resolver todos os aspectos da engenharia de *software*, centrando-se fundamentalmente em quatro atividades: Codificar, Testar, Escutar e Desenhar. Diz Beck: “Codifica-se porque se não codificarmos, não se fez nada. Testa-se porque se não testarmos, não se sabe se terminou a codificação. Escuta-se porque se não escutarmos, não se sabe o que codificar ou o que testar. E se desenha para poder seguir codificando, testando e

---

<sup>32</sup> [BECK, 2000], *Extreme Programming Explained*, Addison-Wesley.

escutando indefinidamente”<sup>33</sup>. Beck oferece um repertório de técnicas que funcionam bem em equipes pequenas e foram amplamente adotadas, muitas vezes com sucesso, mas também às vezes criticadas.

As técnicas de Beck são: O Jogo do Planejamento; Pequenas Entregas (de produto); Metáfora; *Design* Simples; Desenvolvimento Dirigido por Testes; Refatoração; Programação em Duplas (ou em Pares); Propriedade Coletiva (dos produtos pela equipe); Integração Contínua; Semana de 40 Horas (hoje chamada Passo Sustentável); Cliente Presente; e Padrões de Código. Como são de uso comum em muitas aplicações de métodos ágeis, entre as que adotam conscientemente ou não o XP, daremos aqui uma síntese de seu conteúdo, tentando explicar o suficiente para que aquelas que despertem o interesse do leitor possam ser investigadas na bibliografia oferecida. Recomendamos ao leitor [KNIBERG, 2007], *Scrum and XP from the Trenches: How we do Scrum*, por seu estilo prático e abrangente.

### **O Jogo do Planejamento**

Beck tenta equilibrar as necessidades do negócio com as necessidades (técnicas) da equipe. Nenhuma, para ele, deve dominar a outra. Propõe um diálogo onde a organização cliente define o escopo, prioridades e a programação e composição das entregas (*releases*) de código. Michael Cohn, em [COHN, 2006], e Anderson em [ANDERSON, 2011], descrevem em detalhe uma variante que se chama o jogo do planejamento. Basicamente é similar ao que Barry Boehm tornou popular como “*Wide Band Delphi*” em [BOEHM, 1981], quer dizer, uma estimativa feita por especialistas, neste caso, os membros da equipe, que converge a partir da discussão do primeiro resultado vendo a categoria e a média da estimativa. A iteração é feita até que a diferença entre os extremos seja aceitável, reduzindo-a em cada iteração mediante a justificativa de cada um sobre seu prognóstico. É importante que as equipes discutam com o cliente suas estimativas, mas não para que o cliente arbitrariamente fixe a duração das tarefas. A equipe técnica tem a responsabilidade de alertar sobre as consequências de escolher certos *designs* ou elementos técnicos.

### **Pequenas Entregas**

Todas as entregas em XP são feitas em períodos curtos, idealmente a cada duas semanas. Isto mantém o cliente interessado e comprometido, já que o *feedback* assim obtido aumenta o valor do produto.

### **Metáfora**

Ao invés de esperar que a arquitetura ofereça coesão à estrutura do produto, XP utiliza uma “metáfora”, uma explicação de muito alto nível do que se espera produzir. Por exemplo, ao invés de documentar em um diagrama as interfaces com os usuários e os componentes esperados, em XP se fala do comportamento “como se fosse um terminal de Banco 24 Horas”. Isto deveria ser suficiente para se deduzir propriedades e atributos do produto, por exemplo, que haverá um administrador e diferentes tipos de clientes. Desse modo, é reduzida a necessidade de refrescar a memória de cada programador quando cria o *design*.

### **Design Simples**

Segundo Beck, o melhor *design* é o que melhor se ajusta aos casos de teste, executa todos, não tem lógica duplicada, contém todos os atributos e intenções que os programadores querem do produto e faz isso com a menor quantidade de classes possível.

### **Desenvolvimento Dirigido por Testes**

No XP qualquer característica do programa que não tenha um teste associado, não existe. Os programadores escrevem seus testes para ganhar e manter a confiança no código que geram, e os clientes fazem isso pelo mesmo motivo. Ao acrescentar a quantidade de testes associados com o código é impossível rompê-lo ao introduzir mudanças sem que os defeitos saltem à vista. De fato, o teste de regressão é realizado permanentemente. No XP, o programador primeiro escreve o caso de teste da mudança que quer implementar e o executa sobre o código atual, ou seja, sem modificá-lo, assegurando-se de que não funcione. Desse

---

<sup>33</sup> [BECK, 2000], op. cit. Capítulo 9, *Back to Basics*, p. 49. Tradução dos autores.

modo verifica-se o caso de teste, que deve achar defeitos. Depois escreve o código correspondente até que o caso de teste não encontre defeitos.

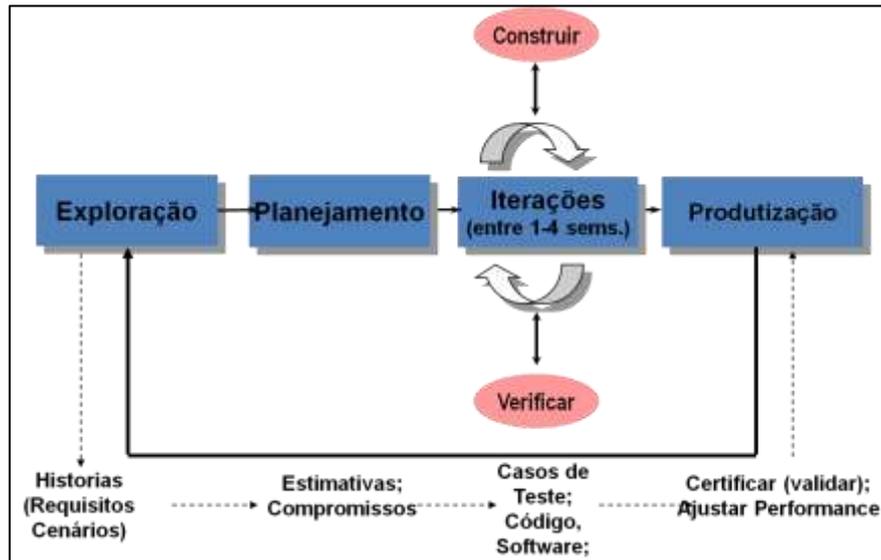


Figura 3.4: Ciclo de Desenvolvimento de XP

### Refatoração

Este enfoque de mudanças pequenas pode ter um efeito indesejado, já que otimiza localmente o código, podendo anular outras otimizações locais e perdendo de vista a otimização global. Às vezes, é necessário realizar mudanças no *design* porque surgem certos padrões de comportamento indesejáveis. Uma das tarefas a realizar ao mudar o código é investigar a existência de padrões que não são considerados “boa programação”, como código duplicado, dados que passam entre métodos sem ser processados e outros mais que podem ser encontrados nas referências<sup>34</sup>.

### Programação em Duplas (ou em Pares)

A programação em duplas ou em pares, como seu nome indica, é uma técnica na qual duas ou às vezes três pessoas trabalham juntas no desenvolvimento de um segmento do código. São vários os *sites* que discutem os prós e os contras da programação em duplas. Os autores se tendem a acreditar que a socialização da tarefa entre dois evita interrupções e permite aos dois programadores entrar em “*flow*” segundo tal definição em [CSIKSZENTMIHALYI’S, 1991], de modo que a própria tarefa “flui” (daí o nome) e há prazer em levá-la adiante. É desse modo que justifica o sucesso da produtividade de sua técnica particular de programação em duplas<sup>35,36</sup>. Estudos realizados há muitos anos por [DE MARCO e LISTER, 1987] mostram que uma pessoa em um ambiente com baixa quantidade de interrupções é até 3,8 vezes mais produtiva que outra da mesma competência em um ambiente com as interrupções habituais. Este estudo é anterior à dispersão dos sentidos propiciada pela *internet*, de modo que o ambiente normal de hoje em dia é muito mais “ruidoso” do que era no final dos anos 80.

É pouco provável que uma dupla de pessoas trabalhando juntas aceite interrupções, enquanto que uma pessoa sozinha em seu escritório é vítima do correio eletrônico, do “*Messenger*” e até do celular, em suas distintas variantes. Como se isto fosse pouco, as empresas fabricantes de telefones estão acrescentando mais “inteligência” a seus produtos, e receberemos ainda mais interrupções: “o voo de Dorita, a Exploradora, está saindo com atraso” ou “minha tia Rosinha comprou sapatos novos e publicou no *Facebook*”, e por aí vai. A programação em duplas nos

<sup>34</sup> Vejam *code smells*, <http://c2.com/cgi/wiki?CodeSmell>

<sup>35</sup> Os autores argumentam, além disso, que ao trabalhar em uma equipe de duas pessoas (ou três, se há um *coach*), as interrupções comuns de correios eletrônicos, conversas *on-line* e outras distrações que podem ser notadas nos cubículos dos programadores, desaparecem por completo.

<sup>36</sup> <http://www.codinghorror.com/blog/2006/09/the-multi-tasking-myth.html> cita várias fontes que atacam a crença de que fazer muitas coisas ao mesmo tempo serve para ganhar tempo.

poupa da humilhação de sucumbir a todas essas tentações e permite que aproveitemos o “*flow*”.

Uma variante importante da programação em duplas é quando deixa de ser em dupla, ao ter um terceiro participante. Este costuma ser o líder técnico, ou o programador mais respeitado pela equipe. Sua participação tem intenções formativas. Geralmente participa quando um ou os dois programadores carecem de experiência nesta técnica. A seguir aproveitaremos para juntar estatísticas que podem ser usadas na coleta e análise de dados de revisões por duplas.

### **Propriedade Coletiva (dos produtos pela equipe)**

Se duas pessoas podem participar do evento de programação e estas duplas podem mudar: quem é o dono do produto? Obviamente, o programa pertence à equipe em sua totalidade. Cada um participa da responsabilidade de melhorar o produto e ajudar a equipe a se apropriar do produto.

### **Integração Contínua**

Parte da possibilidade de ter propriedade coletiva do código é o fato de que a qualidade dele é colocada à prova o tempo todo, deixando em evidência qualquer defeito quase tão rapidamente quanto se incorre nele. Para começar, o *design* guiado pelos testes inicia um caminho baseado na qualidade como meta, e para seguir, a integração contínua aproveita a geração dos casos de teste para verificar cada mudança contra a base de dados de teste, assegurando que não há regressão de defeitos anteriores. Isto implica que cada novo acréscimo de código é integrado o mais rápido possível ao produto em evolução. Para concluir, a equipe apresenta com regularidade, em períodos curtos, seu produto parcial ao cliente, de modo que o *feedback* é frequente e é possível senti-lo. É frequente que se gerem regras nas equipes relacionadas com esta integração contínua, que requer um *software* especializado e uma disciplina de gestão da configuração e de controle de mudanças muito bem definida.

### **Semana de 40 Horas (hoje chamada Passo Sustentável)**

Apesar de ter muitas regras que a diferenciam das equipes comuns, uma equipe XP é suscetível aos mesmos problemas derivados das pressões do ambiente. É possível que se capitule ante um cliente loquaz e persistente, deixando a equipe com uma tarefa de maior envergadura do que pode fazer razoavelmente. A regra é que não se trabalhe fora do horário, mas se isso for necessário, é importante que se estenda por mais de uma semana. Algumas equipes levam isto a um *Sprint*, outras a duas semanas, mas todas têm claro que o pessoal cansado introduz muitos defeitos e estes introduzem muita desconfiança, retrabalho e interrupções.

### **Cliente Presente**

Grande parte do sucesso de um projeto, seja ágil ou não, baseia-se na participação do cliente. Já vimos (nota 30 deste capítulo) que um usuário colaborativo, representativo, autorizado, comprometido e com conhecimento é primordial para o sucesso de um projeto. Mas se isto é certo para os projetos tradicionais, é indispensável para XP, ainda mais que para *Scrum*. Durante o *Sprint*, em *Scrum*, o Dono do Produto está ausente e só participa convocado pela equipe. Em XP é parte da equipe, convive com ela. Isto permite validar as ideias rapidamente e contar com uma voz que permita seguir caminhos alternativos quando for necessário.

### **Padrões do Código**

O último ponto que tocamos aqui é sobre padrões de código. Se os programadores escolhem o código que querem modificar, trabalham com distintos colegas várias vezes por dia, e o código deve ser lido e entendido por todos, melhor que haja um único estilo de programação. Há muitos anos que [KERNIGHAN e PLAUGER, 1974] falam do estilo de programação. Nós aconselhamos muito mais que um padrão. A equipe teria de ler o livro sobre estilos de programação e adotar suas próprias regras. A maneira de introduzi-las é sendo fiel à programação em duplas, com ou sem o “*coach*” presente.

## Escalonamento

Quase desde o princípio de sua existência, a engenharia de *software* se ocupa de dois problemas ligados entre si: “*Programming in the Large*” e “*Programming in the Many*”<sup>37</sup>. O primeiro problema é o de atacar projetos grandes. O conceito de “grande” é relativo à média que produz a organização. Se uma organização habitualmente gera 100.000 linhas de código por ano, um incremento de 10.000 linhas para um ano é um problema logístico, mas não muda radicalmente a natureza do assunto. Por outro lado, para a organização que habitualmente gera 6.000 linhas de código anuais, 10.000 linhas a mais pode ser um salto quantitativo impossível de absorver. É claro, uma parte da solução consiste em acrescentar pessoal, o que se conhece como “*Programming in the Many*”. Como é conhecido, a aprendizagem da programação é um processo individual. Portanto, aprender a programar com muitos e entre muitos é um exercício disciplinar importante. No caso dos métodos ágeis existem autores que se dedicaram a resolver estes problemas, notavelmente [COCKBURN, 2005] e [PALMER e FELSING, 2002], estes últimos sobre as ideias originais de Peter Coad [COAD *et al.*, 1999]. No caso de Cockburn, seu método *Crystal Clear* é, na realidade, uma família de métodos cada vez mais complexos. As ideias de Coad, por outro lado, estão baseadas em sólidos argumentos arquitetônicos; nós as usaremos porque o que foi apresentado até aqui não é útil para sistemas grandes e complexos, a serem desenvolvidos por equipes numerosas.

Nenhum dos três métodos esboçados neste capítulo “escala” bem, no sentido de que, à medida que é necessária uma equipe de maior tamanho para poder entregar um sistema maior e complexo em prazos aceitáveis pelo mercado, todos eles começam a perder propriedades que são manifestas quando a equipe é pequena. *Kanban* é aquele que, por ter menos regras, também tem menores expectativas; mas ocorre que seu próprio objetivo, o de reduzir o número de componentes em desenvolvimento ao mesmo tempo, conspira contra seu uso em sistemas grandes e complexos.

### 3.5 Feature Driven Development

*Feature Driven Development*, ou FDD, é uma alternativa interessante porque combina a velocidade e flexibilidade dos métodos ágeis com alta escalabilidade. Utilizando algumas técnicas de alto impacto, extraídas das boas práticas de engenharia de *software*, FDD consegue harmonizar o uso de modelos e planejamento com práticas ágeis. FDD, na verdade, consegue se colocar no meio das facções que apoiam um ou outro lado na “guerra religiosa” entre os agilistas e os tradicionalistas.

Contado em poucas palavras, FDD consiste em desenvolver um modelo do domínio entre a equipe de desenvolvimento e os especialistas no contexto. Tirando a informação do desenvolvimento do modelo e outras atividades que poderiam ter acontecido em relação à identificação de requisitos, a equipe constrói uma lista de particularidades e características do produto (“*features*”) expressando-a como funções formuladas sob um padrão <ação> <resultado> <objeto>. Por exemplo, “Calcular o total de horas trabalhadas pelos consultores”, ou “Devolver o valor da hora média de esforço por ponto de função”.

Cada uma destas funções é suficientemente pequena e uma equipe pode implementá-la em duas semanas de trabalho ou menos. No entanto, tampouco é desejável que sejam muito fragmentadas. É desejável que as funções só se dividam em outras menores quando se intui que em duas semanas não poderão ser implementadas, de outro modo são mantidas nesse nível de abstração.

Agora, é possível seguir o modelo para fazer um planejamento rápido e colocar para trabalhar em paralelo várias equipes que coordenem suas interfaces utilizando um mesmo modelo, atribuindo as funções da lista. Mediante este simples procedimento se evitam muitas dores de cabeça posteriores, muitas horas de refatoração e muitos defeitos entregues ao cliente.

O ciclo de desenvolvimento de FDD é muito formal. De todos que vimos, é o que tem a definição mais parecida aos processos típicos das organizações grandes. Pode ser porque os autores querem uma clara definição dos passos e dos papéis, pode ter sido uma exigência do ambiente para conseguir sua adoção nas primeiras implementações. O caso é que os cinco

---

<sup>37</sup> [BORIA, 1987] *Ingeniería de Software*, Kapelusz

processos que mostraremos na Figura 3.5 estão perfeitamente definidos, assim como os papéis dos atores que os executam.

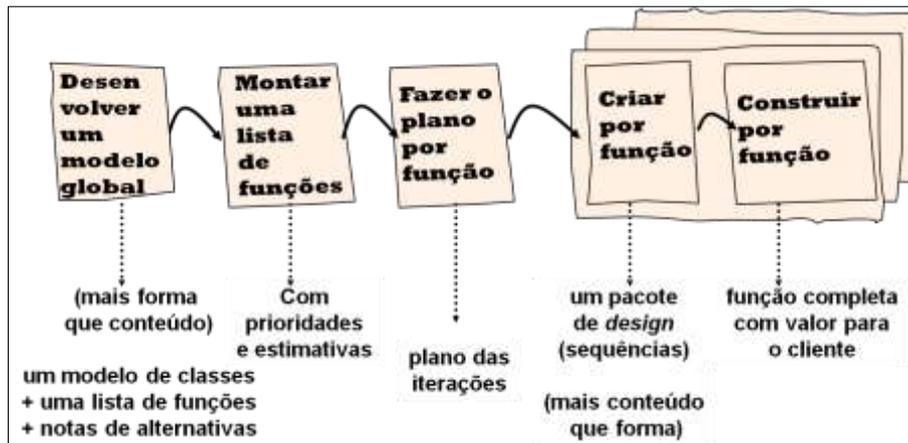


Figura 3.5: Ciclo de Desenvolvimento de FDD

Vamos detalhar um pouco os cinco processos.

O primeiro processo começa quando se determinam os principais atores para todos os papéis requeridos. FDD possui seis papéis chave. O principal papel é o de Gerente de Projeto. É o responsável administrativo do projeto e deve manter o “sistema” projeto andando. Bastante parecido com o *Scrum Master*, o Gerente de Projeto FDD se ocupa de que o processo seja seguido e que sua equipe possa funcionar livre de interrupções. Diferente do *Scrum Master*, o Gerente de Projetos FDD tem a última palavra em relação ao escopo, orçamentos mensais e dotação do projeto.

O segundo papel fundamental é o de Arquiteto-Chefe. Este papel é a contrapartida técnica do Gerente. Apesar do Arquiteto ser o responsável pelo resultado do *design*, suas tarefas incluem facilitar reuniões que permitam aos especialistas no domínio e membros escolhidos da equipe desenhar as características do produto. No entanto, o cargo implica em ter a experiência necessária para poder tomar a decisão definitiva sobre o *design* arquitetônico (ou projeto da arquitetura). Se o projeto é muito simples e a pessoa está capacitada, os papéis de Gerente de Projeto e Arquiteto-Chefe podem ser acumulados por uma só pessoa. Se o projeto é muito complexo tecnicamente e requer muita experiência no domínio ou muito tempo com o cliente, o papel de Arquiteto-Chefe pode ser dividido entre duas pessoas.

Um terceiro papel exigido é o de Gerente de Desenvolvimento. Pela natureza do FDD, várias equipes desenvolvem simultaneamente, o que pode trazer conflitos sobre o uso de recursos. O Gerente de Desenvolvimento tem como principal tarefa resolver esses conflitos, utilizando critérios técnicos e conhecimento das necessidades do cliente. Frequentemente, o papel é atribuído ao Gerente de Projeto ou ao Arquiteto-Chefe. Pode-se notar que o papel exige conhecimento técnico, conhecimento do domínio do cliente e muita capacidade de mediar, buscar o “ganha-ganha”, resolver conflitos, ou até evitar que eles aconteçam.

O quarto papel é atribuído a múltiplos profissionais, um para cada equipe formada, e seu nome é Programador-Chefe<sup>38</sup>. O Programador-Chefe é uma pessoa de grandes habilidades técnicas que passou por várias iterações do ciclo de vida de desenvolvimento repetidas vezes e é capaz de liderar um grupo pequeno. Responde ao plano geral do Arquiteto e trabalha de forma coordenada com o Gerente de Desenvolvimento e os demais Programadores-Chefe.

O quinto papel é o dos que realizam o desenvolvimento cotidiano, chamados de Dono da Classe em FDD. Cada Dono da Classe é um programador de mérito (no FDD não há lugar para a mediocridade) e tem como responsabilidade as atividades de desenvolvimento, teste e documentação de uma parte do produto final, seguindo padrões de programação definidos e, às vezes, colaborando com o Programador-Chefe.

<sup>38</sup> As equipes de “Programador-Chefe” estão descritas por [BROOKS, 1995] seguindo a implementação que fez Harlan Mills na IBM.

O sexto e último dos papéis exigidos é o de Especialista no Domínio. O fato de ser o último em nossa lista não o torna menos importante e, de fato, pode ser o que mais impacto causa na qualidade final do produto e no sucesso do projeto. Além de contar com a óbvia experiência no domínio, que pode estar dividida entre vários protagonistas, é necessário que os Especialistas no Domínio tenham as outras características de CRACK que já mencionamos (ver a nota 31). Sem disponibilidade destes usuários, é muito difícil que a equipe concretize um produto de sucesso.

Depois que os protagonistas são determinados para todos os papéis, com a possível exceção dos Donos de Classe que não precisam participar, o processo pode começar. No primeiro processo se desenvolve um modelo global do produto a ser implementado. É importante assinalar que, igual à citação atribuída a Eisenhower, “os planos não servem, mas o planejamento é tudo”, o modelo não é o que importa nesta etapa. É muito mais importante usar a construção do modelo para estabelecer uma relação com o cliente e investigar as áreas cinzentas do conhecimento do domínio. Esta etapa é importante não porque o resultado é um diagrama que defina com exatidão a natureza e os objetivos do produto, mas porque ilumina a equipe a respeito de quem são os especialistas no domínio, com que apoio pode contar e quais são as “grandes ideias” que guiam o cliente.

De qualquer modo, se não há um produto, não há um critério de finalização, e como já dissemos, FDD tem claramente definidos cada um desses critérios para os cinco processos, ao contrário dos métodos anteriores, que deixam a definição do critério de finalização de cada elemento à equipe. Para dar por concluído o primeiro processo, o Arquiteto-Chefe do projeto deve estar de acordo com o modelo resultante. O modelo deve ter sido construído com o auxílio de todas as partes envolvidas. Isto inclui claramente a equipe de Especialistas no Domínio, mas se for possível a participação dos Donos da Classe, estes deveriam se revezar entre as equipes para se embeber com os conhecimentos dos especialistas e resolvê-los pessoalmente.

O FDD tem sua própria descrição das tarefas a serem realizadas, o que não impede que os autores recomendem muito uma leitura dos livros de [ANDRIOLE, 1993] e [WOOD e SILVER, 1995] e, sobretudo, do artigo de [ZAHNISER, 1995] em seu *site* de *internet*, para ter uma ideia clara das opções, técnicas a empregar e resultados esperados.

Depois que se chegou ao modelo que representa bem o produto, a equipe passa ao segundo processo, construir a lista de funções<sup>39</sup>. Neste passo, a equipe é reduzida aos Programadores-Chefe. Partindo da divisão arquitetônica inicial resultante do primeiro passo, os Programadores-Chefe, ou a equipe da lista de funções, constrói a lista das características que serão implementadas. O processo é iterativo. Juntam-se ao redor do *design* arquitetônico, para o qual já existe uma lista preliminar de tarefas designadas e estas são redistribuídas em áreas. Cada área é um conjunto afim de funcionalidades, possivelmente com seus próprios requisitos não funcionais (segurança, velocidade de resposta, disponibilidade, legibilidade, manutenibilidade e outros do tipo). As áreas, depois de estabilizadas, voltam a se separar em atividades, onde cada uma delas é um conjunto mais detalhado e reduzido da funcionalidade de uma área.

Por exemplo, no nível mais alto, a arquitetura tem um componente com o nome “Gerenciamento”, debaixo da qual são listadas certas atividades: gerenciar clientes, contas, revisões, consertos, etc. Ao definir áreas, a gerência global pode ser uma delas, contendo uma lista de funções relacionadas ao mesmo tempo com todas as outras áreas. Teria, da mesma forma, áreas para clientes, contas e ajustes, a última categoria sendo uma síntese das duas atividades, clientes e contas, listadas na arquitetura como separadas. Quer dizer, a criação de áreas não está de modo algum limitada pela lista inicial da arquitetura, mas é o resultado da experiência dos Programadores-Chefe com domínios semelhantes.

Depois que as atividades foram definidas nas áreas, cada passo dentro delas é uma função. Por exemplo, a atividade de gerenciar clientes na área de gerência pode incluir os passos: Criar Cliente, Ajustar dados do Cliente, Equilibrar dados do Cliente, Dar baixa de Cliente. O resultado deste processo é uma lista derivada da arquitetura e vinculada a ela, que adota a forma de uma árvore. Percorrer a árvore a partir da raiz até o nó mais profundo de um galho implica passar pelos três níveis: Arquitetura Base, Áreas, Atividades e Passos (Figura 3.6). A

<sup>39</sup> Estamos usando funções, ou características, para traduzir o original em inglês “*features*”.

lista então não só contém todas as funções a serem implementadas, mas, ao ser uma árvore, permite designar requisitos não-funcionais a nós intermediários, de modo que é criada realmente uma boa descrição arquitetônica do produto. Seguindo [HOFMEISTER *et al.*, 2000] o nível base é o nível Conceitual, as Áreas constituem o Nível Modular e as funções são a base para os dois níveis seguintes: Execução e Código.

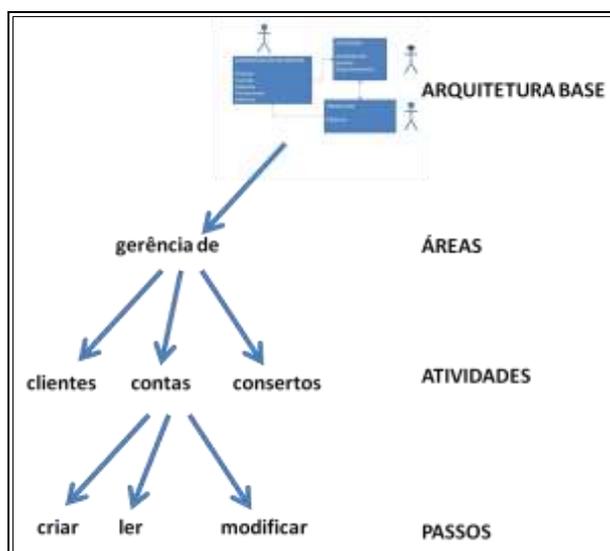


Figura 3.6: Árvore de Funções derivada da Arquitetura

O leitor interessado pode combinar as técnicas do FDD com as sugeridas no livro “*Applied Software Architecture*” [HOFMEISTER, 1999] para obter resultados ainda melhores em aplicações críticas.

O terceiro passo do FDD é a aplicação do processo de realizar o plano do projeto completo contemplando cada função. O Gerente de Projeto, o Gerente de Desenvolvimento e os Programadores-Chefe se reúnem com a lista de funções diante deles e atribuem cada uma das funções aos programadores, que se convertem assim em Donos da Classe. Cada Programador pode ser dono de muitas classes, dependendo de sua experiência, da complexidade e granularidade dessas classes e do grau de conhecimento que tenha do domínio. Os Programadores-Chefe são considerados, para este efeito, programadores e, portanto, acumulam o papel de Programador-Chefe e de Donos da Classe.

A atribuição de funções leva em conta muitas variáveis, tais como a prioridade do cliente para a implementação de áreas, a interdependência das funções, sua complexidade e seu tamanho, a facilidade com que podem ser testadas dependendo da ordem de implementação e da disponibilidade do pessoal. O processo se repete sobre áreas e vai reatribuindo tarefas à medida que as prioridades que são analisadas vão sendo mudadas. Ao finalizar o passo, cada Programador-Chefe tem tarefas atribuídas e um calendário para realizá-las.

Com suas tarefas na mão, cada Programador-Chefe organiza sua equipe de desenvolvimento. No quarto passo, a partir de uma visão associada a cada tarefa da lista, o Programador-Chefe procura as classes associadas a cada uma e busca identificar os programadores que vão participar. Se existirem novas classes associadas, ele as atribui. Este processo se aplica a uma ou mais tarefas a cada vez, considerando todas aquelas que valem a pena implementar em um ciclo de, no máximo, duas semanas para uma equipe. Se o domínio associado é simples e bem conhecido, o Programador-Chefe pode iniciar o *design* do trabalho. Isto implica no desenvolvimento da documentação que permita ao Dono da Classe modificar, estender ou criar o código correspondente sem intervenção de especialistas no domínio. Caso contrário, o Programador-Chefe pede ajuda aos Especialistas para realizar uma revisão do domínio correspondente ao conjunto de funções a implementar.

O Especialista, ou os Especialistas, percorrem o domínio realizando uma apresentação à equipe, onde são tratadas as regras do negócio, os algoritmos da aplicação e os dados necessários para realizar as funções. Se existirem documentos associados às funções, estes são mencionados e explicados. A equipe então os estuda para desenvolver, em primeiro lugar, o diagrama de sequência que define a funcionalidade. Após a definição dos itens de configuração, a equipe se dedica a refinar as classes para acomodar a funcionalidade, o que

permite ao Programador-Chefe publicar um modelo atualizado a ser compartilhado por toda a equipe. Usando ferramentas CASE, o Programador-Chefe atualiza o esqueleto do programa para que se ajuste ao modelo. Depois, cada Dono da Classe toma suas classes e as atualiza para refletir as mudanças no modelo e acrescenta os comentários, o prólogo e, se foi acordado, o invariante resultado da execução dos métodos da classe<sup>40</sup>. Finalmente, a equipe revisa o *design* resultante para garantir sua totalidade, consistência e o que pode ser realizado segundo o plano. Este é o critério de saída da fase<sup>41</sup>.

O último passo é a fase de construção de cada pacote, que deve ter sido definido no passo anterior. Nesta fase o código necessário é escrito ou modificado, é feita uma inspeção do código e é realizado o teste de unidades, para comprovar que o produto que está sendo construído não seja incluído na *baseline* do código com defeitos conhecidos. Ao se aprovar o produto, ele é integrado seguindo os procedimentos habituais de gestão de configuração.

### 3.6 Resumo

Este capítulo cobriu quatro dos melhores métodos ágeis. É comum selecionar práticas de um método e utilizá-las em outro. Tanto que existiu um nome próprio para a combinação de *Scrum* com XP, XBREED, que já não significa nada (o *site* na *internet* foi ocupado por um advogado especialista em danos por acidente). *Kanban* é especialmente útil para começar um processo de melhorias. A visibilidade que oferece e a baixíssima exigência de mudanças fazem com que o peso da responsabilidade pela qualidade fique na equipe de desenvolvimento e não em uma equipe organizacional de processos. Em nossa experiência, é a melhor forma de conseguir a adoção de melhorias. *Scrum* aumenta as exigências, mas sua reputação está justificada: é um método que acelera dramaticamente a produção de *software* e gera sistemas de boa qualidade. Como também deixa muita liberdade à equipe a respeito das técnicas de engenharia, é muito frequente que seja adotado junto com o XP, RUP ou outra (assim chamada) “metodologia”. Todas estas práticas são bastante úteis, mas se escalam relativamente mal. Como os limites ao crescimento são muito rígidos, a adoção de um método mais formal, mas que tenta aproveitar os melhores conselhos dos agilistas, se faz necessária para esses projetos que, ou atacam um volume grande de código ou se ocupam de manter um produto de grande porte com muita complexidade. Escolhemos FDD por nossa afinidade com a corrente de Cutter Consortium, mas igualmente poderíamos ter escolhido *Crystal (op cit.)*. FDD é, para nós, autores, o formato ideal para ser usado com os modelos de maturidade inspirados em CMM<sup>42</sup>, tais como o MPS ou o CMMI.

---

<sup>40</sup> Para ver o uso do invariante da classe em engenharia de *software*, ver [BORIA, 1987] ou para um tratamento mais profundo e sistemático, [GRIES, 1987].

<sup>41</sup> Usamos indistintamente fase, passo e processo para nos referirmos às cinco fases do FDD.

<sup>42</sup> [PAULK *et al.*, 1994] seguindo a [HUMPHREY, 1989].

## CAPÍTULO 4 - O MODELO DE MELHORIA DE PROCESSOS DE SOFTWARE MPS-SW

### 4.1 Competir com a excelência

Entre as muitas variáveis que compõem o sucesso de uma empresa de desenvolvimento de *software*, a qualidade é uma das poucas que se contam dentro da classe do alcançável. Poucas são as empresas que obtêm seus lucros com produtos realmente inovadores; para a maioria, o negócio é produzir melhor que a concorrência e a um menor custo. Utilizam seu conhecimento do domínio para melhorar a satisfação de seus clientes e ampliar sua carteira, tornando cada vez mais caro o custo de transferência para produtos dos concorrentes, restando e ampliando sua cota de mercado. Nesse contexto, os custos de desenvolvimento são mais importantes que os preços, porque a falta de monopólio faz com que a concorrência tente entrar no mercado, diminuindo os preços.

Para uma empresa pequena ou média que luta para manter seu lugar no mercado, a qualidade é primordial, porque os custos de desenvolvimento são mascarados pelos custos de refazer o trabalho. Em [DIAZ e KING, 2002], os autores mostram que o retrabalho supera a quinta parte do esforço total de um projeto (Figura 4.1). Colocado em termos que uma pessoa comum pode entender, o engenheiro de *software* joga fora uma em cada cinco coisas que faz. Se fosse uma lavanderia, uma em cada cinco peças de roupa lavada teria de voltar ao lava-roupas.

As organizações pequenas e médias não podem suportar um custo tão excessivo. Como já explicamos no Capítulo 2, quando fizemos a proposta do uso de LS (*Lean Simplified*) como o método de tratamento do problema, é necessário identificar as fontes de defeitos e removê-las. Apesar de ser possível fazer uso do LS sem nenhum outro apoio, a procura dos defeitos é, em si, uma arte. A maioria das pessoas se sente mais cômoda seguindo um guia. A grande contribuição de Watts Humphrey [HUMPHREY, 1989] consiste na criação de um modelo de estados que oferece precisamente esse guia<sup>43</sup>.

**Tabela 1: *Sistema de Decisão General Dynamics*  
Desempenho do Projeto vs. Nível CMM**

Nível CMM	Porcentagem Retrabalho	Eficácia da Fase de Contenção	Densidade CRUD por KSLOC	Produtividade (x Fator Relativo)
2	23.2%	25.5%	3.20	1 x
3	14.2%	41.5%	0.90	2 x
4	9.5%	62.3%	0.22	1.9 x
5	6.8%	87.3%	0.19	2.9 x

**Figura 4.1: Relação da Refatoração vs. Nível do CMM [DIAZ e KING, 2002]**

De fato, os modelos de estados são anteriores ao trabalho de Humphrey. Em um famoso trabalho<sup>44</sup>, Richard Nolan introduz um modelo de estados para a gestão de recursos computacionais. Como prelúdio, Nolan descreve diferentes usos dos modelos de estados, em campos tão diversos como a astronomia (a formação de estrelas, galáxias e sistemas solares), biologia, ecologia e, desde o século XIX, a economia dos países. Em particular faz referência a

<sup>43</sup> Uma história que circulava na TeraQuest, no final da última década do século passado, relatava que Watts Humphrey teve sua “Eureka a bordo de um avião que o levava de volta a Pittsburgh depois de um seminário no Instituto Juran. Fervoroso admirador das teorias de Deming, Juran e Crosby, Humphrey definiu como seu objetivo eliminar a dificuldade de aplicar as técnicas da qualidade total, incluindo os gráficos de comportamento de processos (cartas de controle), à engenharia de *software*. Teve sua inspiração ao compreender a necessidade de estabelecer processos comuns, que possam ser analisados estatisticamente. Daí seu modelo de estados.

<sup>44</sup> [NOLAN, 1973]. Note-se a homenagem implícita a este artigo como fonte de inspiração no nome do livro de Humphrey op. cit.

dois critérios para a 'boa formação' de modelos de estados, descritos por Simon Kuznets<sup>45</sup>: os estados devem estar bem definidos e devem ser claramente diferentes e demonstráveis empiricamente; e a relação analítica de qualquer estado com seu predecessor ou sucessor deve estar bem definida, tornando possível identificar os processos que fazem um objeto passar de um estado a outro. Continuando nossa rota, tentaremos demonstrar que o modelo de referência MPS-SW é um sólido modelo de estados.

#### 4.2 Um caminho para a excelência organizacional

O Modelo de Referência para *Software* MPS (MR-MPS-SW) [SOFTEX, 2012a] é um modelo de maturidade de processos, inspirado e compatível com o CMMI-DEV [SEI, 2010] e aderente às normas internacionais ISO/IEC 12207 [ISO/IEC, 2008] e ISO/IEC 15504 [ISO/IEC, 2003]. Historicamente, a maioria dos enfoques sobre qualidade desenvolveu 'normas' de boas práticas a serem aplicadas. Apesar de cumprirem uma função importante ao difundir métodos testados e ajudarem a focar esforços em qualidade, elas se centram em seu cumprimento e não no desenvolvimento da organização.

AS NORMAS EXIGEM CUMPRIMENTO OS MODELOS PROCURAM DESENVOLVIMENTO
---

A vantagem de um modelo de melhoria de processos é que indica um caminho desejável para alcançar a excelência. As normas podem ser cumpridas; mas, se são vistas como uma lista de regras a seguir, podem não ser mais do que a soma das partes. Por outro lado, um modelo, quando é bem interpretado, além de indicar as melhores práticas, também mostra uma possível ordenação ao apresentar a sequência mais lógica de implementação. Os modelos mais úteis possuem formas de avaliar o grau de implementação em que se encontra uma empresa. O MR-MPS-SW não é uma exceção, e esta valiosa ferramenta permite à empresa interessada na melhoria de seus processos entender o que já tem implementado e o que falta, além de sugerir os próximos passos. A desvantagem de um modelo é que os caminhos de implementação são múltiplos e dependem da empresa em questão.

De certo modo, a articulação de *Lean Simplified* com um modelo como o MPS é a ferramenta ideal, já que permite identificar as mudanças mais significativas com LS e utilizar as recomendações do modelo para realizá-las. Apesar de que as normas são muito mais fáceis de interpretar e implementar, o efeito sinérgico que possuem é muito escasso e difícil de encontrar, mesmo com muita experiência.

A concorrente de uma empresa que procura a excelência é ela mesma. Os concorrentes que se centram no que "faz o outro" não podem concorrer a longo prazo com aqueles que perseguem a excelência por meio da melhoria contínua. Em particular, "tenta-se que o modelo MPS seja adequado ao perfil de empresas com diferentes tamanhos e características, públicas e privadas, com especial atenção às micro, pequenas e médias empresas. Também espera-se que o modelo MPS seja compatível com os padrões de qualidade aceitos internacionalmente e que tenha como pré-requisito o aproveitamento de toda competência existente nas normas e modelos de melhoria de processos já disponíveis. Dessa forma, tem como base os requisitos de processos definidos nos modelos de melhoria de processos e responde à necessidade de implantar os princípios de engenharia de *software* de forma adequada ao contexto das empresas, estando em consonância com as principais abordagens internacionais para definição, avaliação e melhoria de processos de *software*"<sup>46</sup>.

---

<sup>45</sup> [KUZNETS, 1966]

<sup>46</sup> [SOFTEX, 2012a], página 6

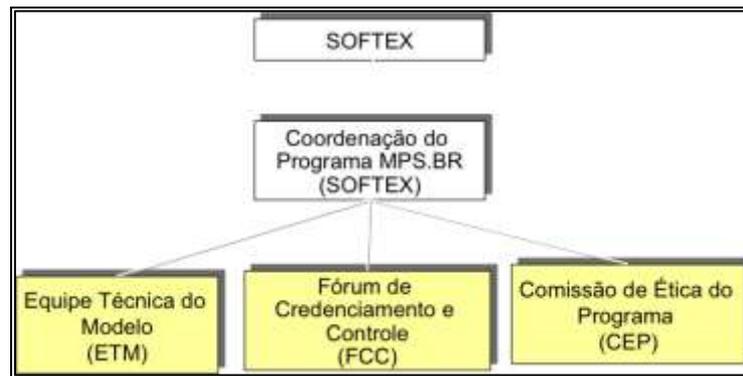


Figura 4.2: Organização do MPS.BR [SOFTEX, 2011]

Em particular, o MPS se ajusta à nomenclatura internacional de guias. Há um Guia Geral MPS de *Software*, MR-MPS-SW [SOFTEX, 2012a], já citado, cuja leitura é indispensável para entender a gênese<sup>47</sup>, os objetivos do modelo e o modelo de negócios. Há um Guia Geral MPS de *Serviços* [SOFTEX, 2012b], um Guia de Aquisição [SOFTEX, 2011a] e um Guia de Avaliação [SOFTEX, 2012c]. Voltaremos a este último antes de fechar este capítulo. Há também um Guia de Implementação para cada nível [SOFTEX 2011b; SOFTEX, 2011c; SOFTEX, 2011d; SOFTEX, 2011e; SOFTEX, 2011f; SOFTEX, 2011g; SOFTEX, 2011h;], além de guias de implementação para Fábrica de *Software* [SOFTEX, 2011j], para Fábrica de Testes [SOFTEX, 2011k] e para empresas que fazem aquisição de *software* [SOFTEX, 2011i], mais um com orientações para a implementação e avaliação do MR-MPS-SW em conjunto com o CMMI-DEV [SOFTEX, 2012d], outro [SOFTEX, 2012e] com uma análise de aderência do MR-MPS-SW em relação à NBR ISO/IEC 29110-4-1 [ABNT, 2012] e um outro [SOFTEX, 2012f] com o mapeamento e sistemas de equivalências entre o MR-MPS-SW e o MoProSoft [OKTABÁ *et al.*, 2005]. O modelo de negócios do MPS divide claramente os papéis e responsabilidades, definindo nos extremos os clientes, usuários finais do modelo, de um lado, e do outro, o Programa MPS.BR coordenado pela SOFTEX, o organismo que dirige o modelo e as instituições habilitadas para implementá-lo e avaliá-lo. Estes dois grupos, não excludentes, devem ser autorizados pela SOFTEX para atuar dentro dos limites do modelo.

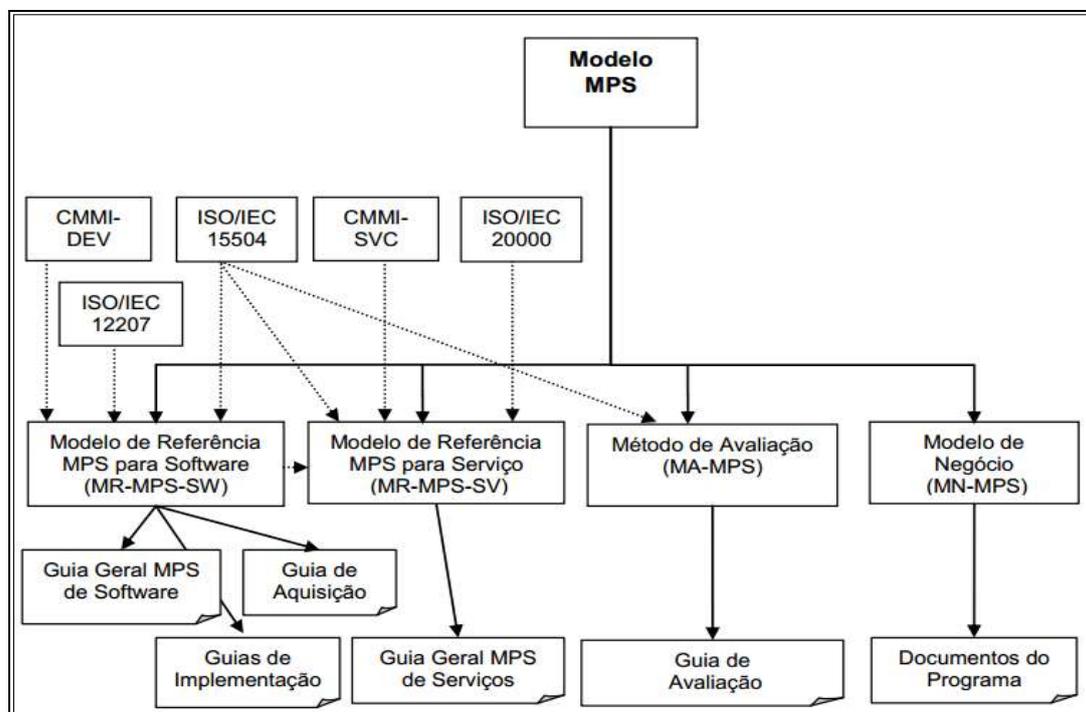


Figura 4.3: Componentes do Modelo MPS [SOFTEX, 2012a]

<sup>47</sup> [SOFTEX, 2012a], página 15, Seção 7, Base técnica para a definição do modelo MPS

### 4.3 Arquitetura do MPS

O MPS tem uma arquitetura muito simples. Por um lado, são descritos os processos que compõem o modelo. Cada processo tem seu propósito e seus resultados esperados. É possível entender cada processo de forma separada, mas não reside aí o valor do modelo: como modelo de estados de maturidade, o MPS organiza o desenvolvimento utilizando grupos de processos. No MPS, os níveis de maturidade são sete, do G ao A. Para alcançar um certo nível de maturidade, a organização deve cumprir com todos os resultados esperados dos processos definidos até (e incluindo) o nível de maturidade em questão. Os níveis de maturidade são cumulativos, quer dizer, para alcançar o F deve-se também atender o G. Para alcançar o E deve-se atender o F, o que implica em atender também o G.

Assim, para alcançar um nível, é preciso mostrar que todos os resultados de todos os processos definidos para ele foram alcançados e também todos os que estão abaixo. O leitor pode imaginar os níveis como bonecas russas, uma dentro da outra. Além disso, é preciso mostrar que os Atributos de Processo correspondentes ao nível em questão também estão cobertos. Estes Atributos de Processo são mostrados na Figura 4.4 nas colunas da direita e são denominados AP (Atributos de Processo). Os Atributos de Processo definem a Capacidade do Processo e também estão descritos em termos de resultados esperados, como os processos em si. A Capacidade do Processo expressa o grau de refinamento e a institucionalização com que o processo é executado na organização.

Nível	Processos	Atributos de Processo
A		AP 1.1, AP 2.1, AP 2.2, AP 3.1, AP 3.2, AP 4.1, AP 4.2, AP 5.1 e AP 5.2
B	Gerência de Projetos – GPR (evolução)	AP 1.1, AP 2.1, AP 2.2, AP 3.1, AP 3.2, AP 4.1 e AP 4.2
C	Gerência de Riscos – GRI	AP 1.1, AP 2.1, AP 2.2, AP 3.1 e AP 3.2
	Desenvolvimento para Reutilização – DRU	
	Gerência de Decisões – GDE	
D	Verificação – VER	AP 1.1, AP 2.1, AP 2.2, AP 3.1 e AP 3.2
	Validação – VAL	
	Projeto e Construção do Produto – PCP	
	Integração do Produto – ITP	
E	Desenvolvimento de Requisitos – DRE	AP 1.1, AP 2.1, AP 2.2, AP 3.1 e AP 3.2
	Gerência de Projetos – GPR (evolução)	
	Gerência de Reutilização – GRU	
	Gerência de Recursos Humanos – GRH	
	Definição do Processo Organizacional – DFP	
F	Avaliação e Melhoria do Processo Organizacional – AMP	AP 1.1, AP 2.1 e AP 2.2
	Medição – MED	
	Garantia da Qualidade – GQA	
	Gerência de Portfólio de Projetos – GPP	
	Gerência de Configuração – GCO	
G	Aquisição – AQU	AP 1.1 e AP 2.1
	Gerência de Requisitos – GRE	
	Gerência de Projetos – GPR	

Figura 4.4: Níveis de maturidade do MR-MPS-SW [SOFTEX, 2012a]

No Modelo de Referência MPS, à medida que a organização evolui pelos níveis de maturidade, a capacidade para realizar os processos deve melhorar da mesma forma. No entanto, os benefícios em termos de desempenho não surgem do rigor com que estes são implementados, mas da cultura que é gerada por sua aplicação correta e consistente. A capacidade para realizar os processos é manifestada pela implementação dos Atributos de Processo (AP), que por sua vez é manifestada pela implementação dos Resultados esperados dos Atributos de Processo (RAP). A Figura 4.5 descreve a arquitetura graficamente.

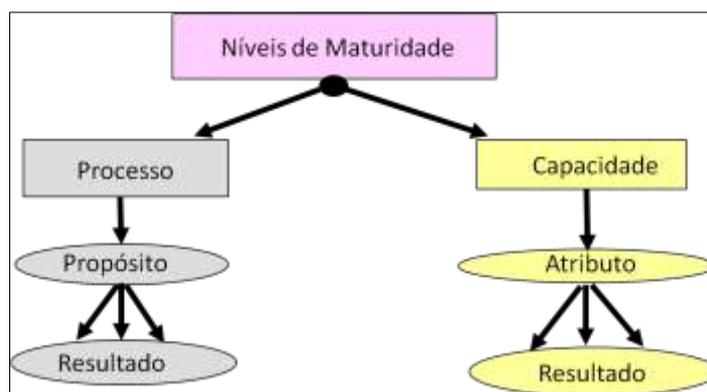


Figura 4.5: Estrutura do MPS [SOFTEX, 2011]

Os detalhes de cada processo e os Atributos de Processo de cada nível serão discutidos em maior detalhe nos respectivos capítulos que tratam, um a um, o desenvolvimento organizacional da empresa Tahini-Tahini. Enquanto isso, focaremos na promessa que fizemos ao começar este Capítulo.

#### 4.4 Os Níveis de Maturidade do MPS

Os dois critérios para a ‘boa formação’ de modelos de estados definidos por Kuznets são que os estados devem estar bem definidos, ser claramente distintos e demonstráveis empiricamente; e que a relação analítica de qualquer estado com seu predecessor ou sucessor deve estar bem definida, tornando possível identificar os processos que fazem com que um objeto se mova de um estado ao outro. Para ver se o MPS cumpre com a primeira condição basta revisar com detalhes a Figura 4.4. Cada um dos diferentes níveis têm perfeitamente definidos os processos que os constituem e estes processos são diferentes, mesmo quando tiverem o mesmo nome, já que esses são evoluções dos homônimos anteriores e, sendo assim, são diferentes. Os níveis, além disso, estão bem definidos, diferenciando-se pelos Atributos de Processo que devem ser implementados em cada caso.

Vejam os que caracteriza cada nível. Imaginemos uma empresa que não tem processos bem definidos e façamos com que amadureça por meio dos níveis do MPS, começando pelo nível G: a empresa precisa, primeiro, ter o controle das tarefas, a partir dos requisitos, para poder cumprir seus compromissos. Ao invés de correr para implementar o que o cliente quer, a equipe faz uma pausa para planejar e entre as tarefas planejadas está o monitoramento. Surge a cultura básica de projetos. Os benefícios do Nível G se manifestam em um melhor foco no negócio, porque os compromissos assumidos são compreendidos e honrados. Há entendimento suficiente do estado do projeto para administrar as expectativas dos clientes. Há, da mesma maneira, uma maior transparência. O estado do projeto é comunicado e compartilhado. As expectativas são documentadas e explicadas. A alta gerência trabalha com informação verdadeira. O outro resultado importante deste nível é a instituição de uma nova disciplina, pela qual as equipes revisam e atualizam os compromissos. Estes são respeitados e todos os esforços são feitos para que isso seja realizado com transparência.

Para passar do nível G ao F, a organização precisa tomar consciência do valor de fazer as coisas de forma repetível e deve começar a cuidar de seus ativos organizacionais. É preciso começar a medir para entender e poder melhorar seu sistema de decisão. É no Nível F que começa a cultura da objetividade. Isto se manifesta em um Sistema de Medições, pelo qual se distingue entre medir e usar indicadores de gestão. O Sistema de Medições está alinhado com as necessidades de informação dos vários níveis e fornece retroalimentação verdadeira aos que tomam decisões. Além disso, a organização cuida de proteger de seus ativos: os produtos de trabalho são reconhecidos como ativos organizacionais e protegidos. Controlam-se as mudanças e são criadas versões dos produtos da equipe como parte integral do processo. No Nível F é, também, adotada uma melhor estratégia de relacionamento com os fornecedores. Os acordos são estabelecidos para beneficiar todas as partes, não só o adquirente. Começam a integrar os fornecedores com a linha de produção, com o propósito de eliminar esperas e diminuir custos. Começa-se a entender que os funcionários não são um custo irre recuperável, mas um ativo que pode ser utilizado de diferentes maneiras, e estas podem ser medidas e comparadas. A gerência do portfólio de projetos permite aproveitar ao máximo os recursos e esforços da organização de forma alinhada com seus objetivos estratégicos.

Do F ao E: O valor do que é repetível passa a ser o valor do que pode ser compartilhado. A organização foca no aprendizado e cria os ativos para que os projetos possam trabalhar sobre processos comuns. Nasce a cultura do aprendizado. A partir da ênfase nos processos organizacionais, discute-se “o que”, e não “quem é responsável pelos problemas e o processo é melhorado para incrementar as margens do negócio. Estes processos-padrão da organização permitem compartilhar as melhores práticas em favor de melhorar a eficiência e a efetividade dos projetos. Acompanhados dos ativos de processo correspondentes, cultivam a melhoria a partir da experimentação controlada e do compartilhamento de experiências. É facilitado e estimulado o ajuste dos processos às necessidades individuais dos projetos. Como isto requer novas competências básicas, forma-se o pessoal de maneira sistemática para que utilizem os processos. Os atores são, então, efetivos e eficientes. Os processos compartilhados facilitam as melhorias ao facilitar a comunicação e o compartilhamento de experiências. Trata-se de uma verdadeira organização que aprende; esta, como um todo, começa a fazer as coisas bem desde o começo, melhorando a qualidade e eliminando retrabalhos custosos. Estas mudanças permitem, da mesma forma, uma melhor integração. Atende-se ao ciclo de vida completo do colaborador, da definição dos cargos à seleção, contratação e preparação do pessoal. São criadas equipes levando em conta a eficiência de seu funcionamento futuro. É estabelecida e mantida a coordenação entre equipes. O trabalho a realizar é identificado, assim como os ativos que podem ser reutilizados para incrementar o reuso e baixar os custos. Ainda antes de pensar em *design* detalhado, são pensadas as arquiteturas de linhas de produto.

Do E ao D: A organização passa a se centrar na engenharia de *software*. Culturalmente, começa a visão dos grupos de interesse e das especializações funcionais baseadas nas experiências conjuntas dos projetos. O rendimento individual e coletivo aumenta, e com isso o sentido de posse. Diminui a rotatividade de pessoal e aumenta a produtividade. As equipes se integram ainda mais e é comum o apoio ao outro em sua tarefa, mesmo esta não sendo sua especialidade (engenheiros de teste integrando equipes de inspeções, engenheiros de *software* entrevistando o cliente).

Do D ao C: A organização se orienta à pró-atividade. Começa a cultura de previsão e da qualidade total. É possível começar a falar de zero defeitos e da busca da excelência. A mentalidade de “isso aqui não pode acontecer” dá lugar a “o que vamos fazer para evitar que aconteça” e “o que podemos fazer se acontecer de qualquer forma”. Enquanto, no nível E, são identificados ativos que merecem ser considerados para sua reutilização, baseando-se em critérios de oportunidade e qualidade, neste nível, dada sua característica de olhar para frente, são identificadas oportunidades de gerar ativos para reutilização. Desse modo, a organização volta-se, cada vez mais, para uma linha de produção de *software*.

Do C ao B: Nasce a cultura do conhecimento profundo a partir do entendimento da variação e da estabilidade. Aparece a cultura da previsão, mediante conhecimento estatístico. A institucionalização da gestão quantitativa faz com que todos pensem em como é que podiam viver sem este conhecimento antes.

Do B ao A: Da previsão se passa à competência com a excelência. A organização procura otimizar cada custo, melhorar cada dia mais.

É importante entender o porquê desta sequência. O que as empresas não possuem quando iniciam seu caminho de melhoria de processos é a disciplina de projetos, o que realmente têm é a engenharia. De nada vale a engenharia sem disciplina. É por isso que *Scrum* é tão reverenciado, porque consegue prender a imaginação das pessoas que acreditam que, ao fazer *Scrum*, não seguem processos. De fato, é sumamente disciplinado e tem claros processos, só que alguns são metaprocessos e a própria equipe pode modificá-los.

Para resumir o exposto:

- Nível G: Disciplina de Projetos (organizar o trabalho)
- Nível F: Disciplina de Qualidade Inicial (organizar a empresa/organização)
- Nível E: Disciplina de Conhecimento Compartilhado (aprender as boas práticas e compartilhá-las)
- Nível D: Disciplina de Engenharia (organizar o desenvolvimento em maior nível de detalhe)
- Nível C: Disciplina de Previsão (cultivar o pensamento proativo)
- Nível B: Disciplina de Qualidade Total (entender os processos críticos quantitativamente e prever seu comportamento em um projeto)

Nível A: Disciplina da Excelência (procurar o panteão da disciplina).

Tomando isto com uma granularidade um pouco maior, os níveis G e F ‘estabilizam o paciente’ para que possa ser tratado e os níveis E, D e C montam a fábrica para que seja possível entender os processos quantitativamente.

#### 4.5 Para que a Mudança Aconteça

As organizações que sobrevivem são aquelas que melhor se adaptam a mudanças. A melhoria de processos não é a modificação dos documentos que descrevem os processos, é a modificação da conduta das pessoas que devem segui-los. Isto não é um tema simples, pelo contrário, é algo muito difícil de conseguir e requer profunda atenção. Dependendo do escopo e da profundidade de uma mudança, é mais difícil conseguir que ela resulte em um bom resultado. Se o escopo é individual, como quando há uma mudança menor em um procedimento, a difusão e implantação da mudança é simples e pode ser realizada em nível individual. Quando o escopo supõe a modificação do comportamento da equipe, a mudança não é trivial. A mais difícil de se conseguir é a mudança da cultura. Poucas vezes isto tem como resultado uma situação de fácil adoção, sendo que, na maioria das vezes, as organizações que tentam sem o adequado planejamento, acabam fracassando. Do ponto de vista do desenvolvimento organizacional, o amadurecimento de uma organização pode ou não supor a mudança de sua cultura.

Na realidade, não é que a mudança em si seja difícil, é a mudança orientada a certos objetivos que é complicada. Se olharmos ao nosso redor nada permanece estático, imutável, tudo muda permanentemente. Mas essa mudança espontânea não é equivalente à melhoria. O que procuramos é orientar a mudança para a melhoria do desempenho. Quando queremos que as pessoas realizem algo de maneira diferente do que estão realizando na atualidade, a mudança produz uma ruptura com o conhecido. Apesar de serem evidentes as vantagens da mudança, o familiar é o presente, o agora. Mesmo quando as pessoas estão de acordo com a necessidade de mudança, não quer dizer que estão de acordo com a direção que queremos dar à mudança<sup>48</sup>.

O desconhecido causa temor ou, pelo menos, ressentimento. Esperar que todos entendam a mudança desde o início e a adotem por suas vantagens é ilusório, a maioria ignorará ou resistirá à mudança. Elizabeth Kubler Ross [KUBLER-ROSS, 1997] estudou a sequência de comportamentos que são seguidos normalmente (mas não sempre) ao enfrentar uma mudança de profundo impacto em nossas vidas.

---

<sup>48</sup> A referência bibliográfica mais antiga sobre mudança é de Maquiavel, em 1513, no seu célebre livro *O Príncipe*: “Deve-se considerar não haver coisa mais difícil para cuidar, nem mais duvidosa a conseguir, nem mais perigosa de manejar, que tornar-se chefe e introduzir novas ordens. Isso porque o introdutor tem por inimigos todos aqueles que obtinham vantagens com as velhas instituições e encontra fracos defensores naqueles que das novas ordens se beneficiam. Esta fraqueza nasce, parte por medo dos adversários que ainda têm as leis conformes a seus interesses, parte pela incredulidade dos homens: estes, em verdade, não creem nas inovações se não as veem resultar de uma firme experiência. Donde decorre que a qualquer momento em que os inimigos tenham oportunidade de atacar, o fazem com calor de sectários, enquanto os outros defendem fracamente, de forma que ao lado deles se corre sério perigo...” *O Príncipe*, Nicolau Maquiavel, Capítulo VI, DOS PRINCIPADOS NOVOS QUE SE CONQUISTAM COM AS ARMAS PRÓPRIAS E VIRTUOSAMENTE.

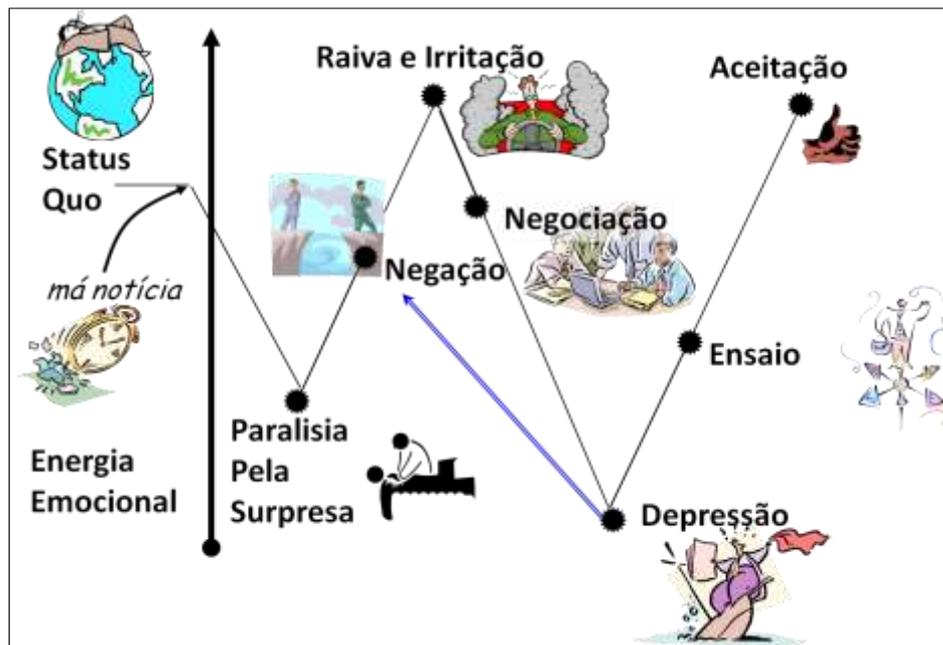


Figura 4.6: Sequencia da Resistência à Mudança (adaptado de [KUBLER-ROSS, 1997])

A seguir evitaremos discutir mudanças culturais profundas, às quais reservamos uma seção ao final deste Capítulo.

Para que uma mudança planejada tenha sucesso é útil contar com todos os elementos de partida. É preciso ter um patrocinador da mudança em uma posição que permita às pessoas alterarem suas prioridades sem violentar sua posição na organização. Esse patrocinador de alto nível precisa contar com o apoio, ou ganhar-lo, dos gerentes abaixo dele. A estes chamaremos de patrocinadores reforçadores. É preciso que aqueles que conduzam efetivamente as atividades da mudança existam. Estes são nossos agentes de mudança. Às vezes, contamos com pessoas de muito prestígio que entendem e apoiam a mudança. Estes são chamados de 'campeões da mudança' e são muito importantes para acelerar a transição.

Ao tratar-se a transição, fala-se muito da mudança, mas é indispensável entender que esta não é um objeto, mas um processo, um *devenir*<sup>49</sup>, algo que ocorre ao longo do tempo. Há um estado inicial real e um estado final desejado, que deve obrigatoriamente ser diferente do atual, pois se não for assim, não há nenhuma mudança. Este estado final não pode ser acessado de imediato e sem esforço, ou não estaríamos descrevendo-o aqui. É um estado que requer mudanças de conduta de várias pessoas e que leva um tempo para implementar e implantar. A passagem do estado atual ao estado final é chamada de 'transição' e é o que causa todos os problemas, em geral fruto de más interpretações sobre o que é a transição.

A Figura 4.7 mostra uma visão muito simples da transição que é desenhada como uma linha reta entre o estado inicial e o desejado. Supõe-se, então, que a introdução da mudança seja gradual e não ofereça maiores problemas.

<sup>49</sup> Movimento pelo qual as coisas se transformam.

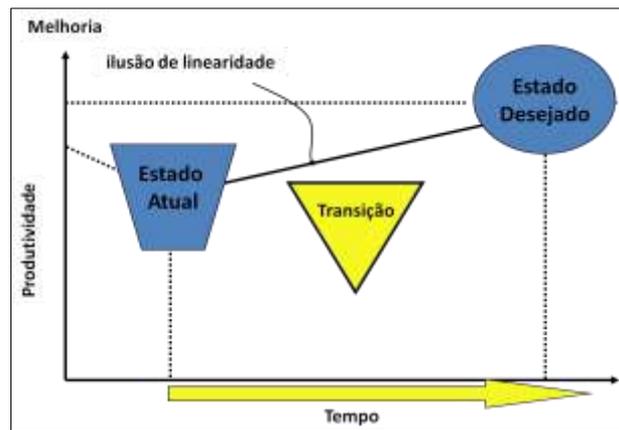


Figura 4.7: Modelo de Transição Ilusória (1)

A realidade é que não é possível conseguir a mudança dessa maneira. É necessário, pelo menos, levar em consideração a curva de aprendizagem, como é mostrado na Figura 4.8.

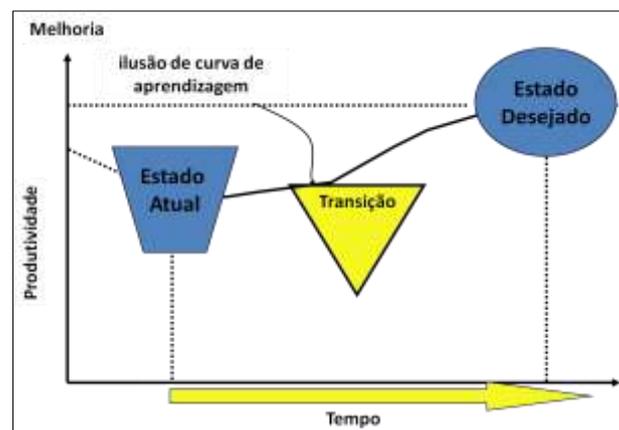


Figura 4.8: Modelo de Transição Ilusória (2)

Esta visão está menos equivocada do que a anterior, mas continua sendo uma visão altamente otimista da realidade. Quando há uma transição importante, o novo é desconhecido e deve ser aprendido. A aprendizagem segue a curva S, mas a produtividade cai durante o período de aprendizagem. De fato, é preciso planejar uma estratégia que faça com que o período de transição seja o mais breve possível e tenha tanto apoio quanto possível para que o projeto de mudança não morra. A Figura 4.9 ilustra o verdadeiro comportamento da transição quando é planejada e realizada com uma estratégia adequada.

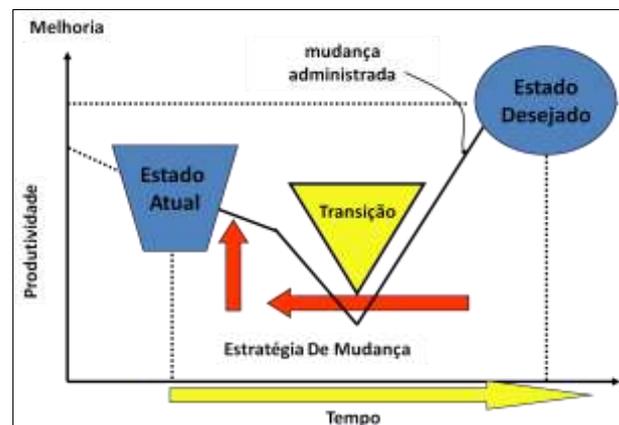


Figura 4.9: Modelo de Transição Administrada

Se a mudança não é planejada, considera-se que a adoção está assegurada pela auto-evidência de sua necessidade. O que vamos obter, provavelmente, é somente um gradualismo que vai manter a produtividade em baixa durante um período, o que fará com que a mudança

seja insustentável e o resultado seja o cancelamento do projeto de melhoria e a conformidade com um novo *status* quo de menor produtividade, como mostra a Figura 4.10.

Duas são as estratégias principais para reduzir o impacto da mudança: uma é dividir a mudança em etapas tão pequenas quanto forem sustentáveis. A outra é dar o apoio necessário para a instalação das novas condutas às equipes que precisam realizá-las. Isto se traduz em treinamento no trabalho, sobre o trabalho, no momento que seja necessário e com os processos reais<sup>50</sup>.

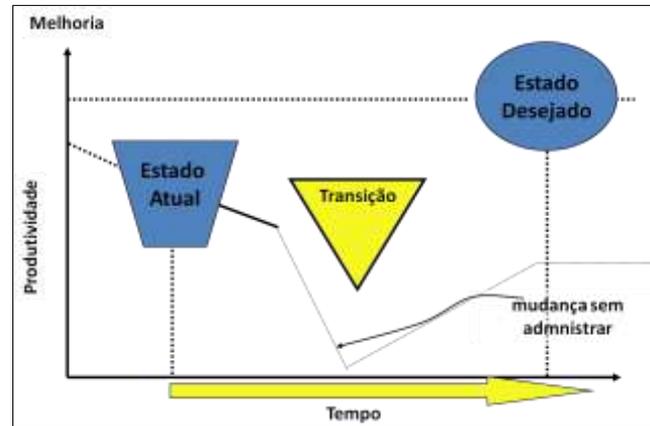


Figura 4.10: Modelo de Transição Sem Administrar

Durante a transição, cada pessoa precisa ser ajudada para construir seu próprio compromisso pessoal com a mudança. Conner e Patterson estudaram isto em [CONNER e PATTERSON, 1982]. Precisamos destacar que o processo individual não é suficiente para conseguir a mudança da organização. Como diz Peter Senge em [SENGE, 2006]<sup>51</sup>, “Quando pensamos na organização que aprende, são as equipes que aprendem”. Dito de outra maneira, só quando as equipes modificam o seu comportamento é que os processos se institucionalizam na organização. A Figura 4.11 mostra os diferentes passos, limiares e estados no processo de construção de comprometimento pessoal com a mudança. Mais abaixo, elaboraremos como se combinam o comprometimento individual com a aprendizagem da equipe.

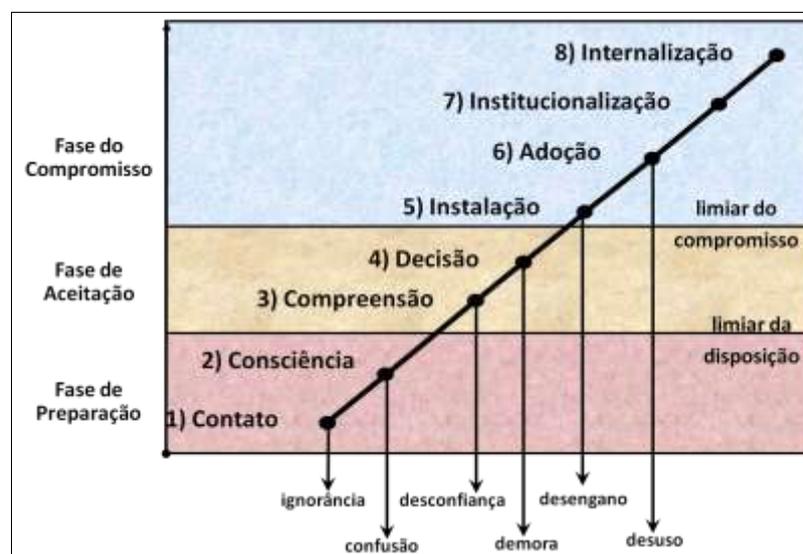


Figura 4.11: Passos do Comprometimento (adaptado de [CONNER 1982])

A primeira fase é a preparação para a aceitação. Só quando o indivíduo aceita a mudança, pode tentar, e só tentando, pode construir seu compromisso. O primeiro passo é sempre o contato. Este contato precisa ser repetido e variado, por exemplo: comunicações orais,

<sup>50</sup> Na TeraQuest chamávamos isto pelas siglas JIT, OTJ, JET (*Just in Time, On the Job, Just Enough Training*)

<sup>51</sup> *When we think of a learning organization, it is the teams that learn.* Peter Senge, “*The fifth Discipline*”

escritas, em boletins da organização, em reuniões mensais com a alta gerência, em reuniões de avanço do projeto, onde for possível, para que não possa ser ignorado. É fácil ignorar uma mudança: basta pensar que não se aplica a você. Se não há o recurso da ignorância, o próximo passo é a tomada de consciência. Quando se percebe que a mudança é inevitável, é aí que se aflora a resistência. Não deveria ser um tema para a confrontação, pois a resistência pode ser inútil, mas não por isso ser ilegítima. Confrontado com a resistência do pessoal, um agente da mudança deve ser paciente e tentar entender os motivos que a geram. Este é um desses momentos nos quais é bom lembrar que a percepção do fato é igual ao fato para quem percebe. Em outras palavras: não importa quem tem razão, o percebido é verdadeiro para quem percebe. Aceitando que não há mudança, por melhor que seja, que não tenha seu lado ruim, o agente deve escutar e negociar. Por exemplo, se a dificuldade acontece porque não há tempo para a aprendizagem dos novos processos ou ferramentas, o agente deve responder fazendo com que esse tempo apareça. Desse modo, colabora com o indivíduo para que este possa cruzar o limiar da disposição, evitando cair na confusão e levando-o à compreensão.

A compreensão do indivíduo com relação à mudança não quer dizer que ele se sinta favorecido por ela. Se não continua o trabalho com a pessoa, esta cairá na desconfiança. Para ajudá-lo a avançar até o próximo passo, a decisão, é imprescindível aplainar o caminho, escutando suas objeções e reduzindo-as com ações concretas. Não há substituto para o sucesso, quando se abandona o indivíduo a seus próprios meios. Neste caso, a mudança é muito improvável, de modo que é necessário continuar com o processo para influenciar o resultado. Neste passo, é provável que se comece com a primeira parte do treinamento nos novos processos, em um alto nível para gerar o vocabulário comum. Tomada a decisão, a pessoa está pronta para cruzar o limiar do compromisso.

Estar disposto a passar à instalação não é o mesmo que realmente realizá-la. Este é o momento em que JIT-OTJ-JET (ver nota 50) é indispensável. Um “coach” com profundos conhecimentos deve se unir à equipe no momento exato para que a experiência de instalação seja positiva e não traumática. Desse modo, evita-se cair no desengano e, assim, as equipes passam a ter permissão de avançar do compromisso até a adoção da mudança. Esta pode cair no desuso ou seguir até a institucionalização, mas do ponto de vista da estratégia da mudança, a adoção é o ponto de chegada.

#### **4.6 Quando a mudança é cultural**

Até aqui consideramos as mudanças estritamente como mudanças de conduta individuais ou de comportamento de equipe. Se adotamos a definição de [CAMERON e QUINN, 2011] para falar de tipos de cultura, encontramos que são quatro dimensões, como mostra a Figura 4.12, derivadas dos dois eixos sobre os quais se apoia uma organização<sup>52</sup>:

---

<sup>52</sup> A afirmação de que são dois os eixos vem de um estudo de mais de 1.500 empresas que responderam com um total de mais de 50.000 dados que, analisados estatisticamente, mostraram que a cultura pode ser explicada por estes quatro quadrantes em relação aos dois eixos.

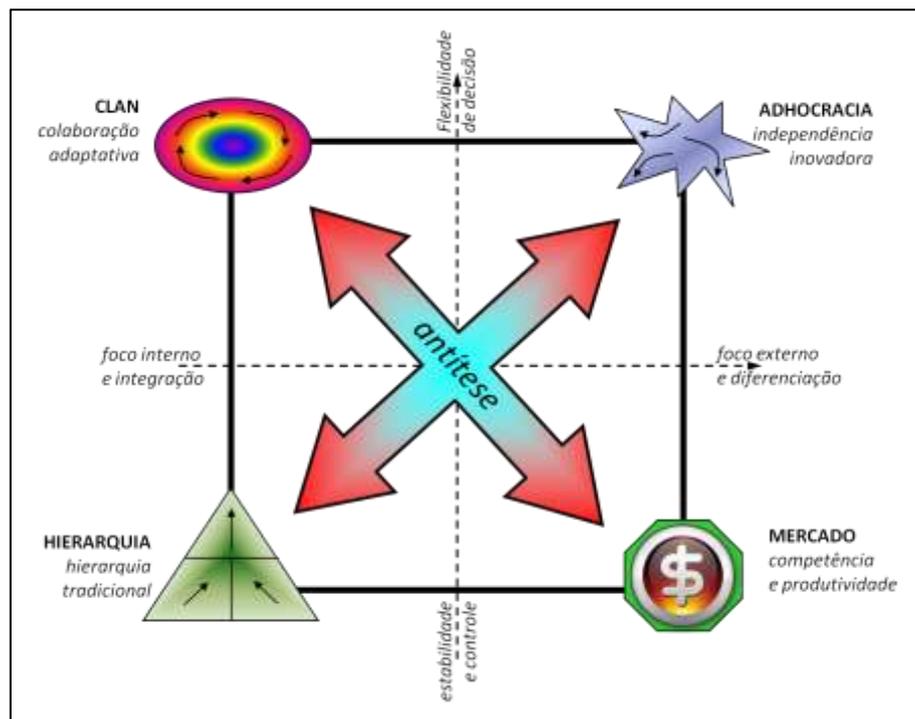


Figura 4.12: Valores em Competência (adaptado de [CAMERON e QUINN, 2011])

Na direção horizontal, o eixo se move de dentro para fora, segundo a ênfase que a organização colocar, para dentro ou para fora de si. Para a esquerda, a empresa foca seus esforços para dentro, enquanto que para a direita a atenção é para seu ambiente, seus clientes e fornecedores. Um enfoque para dentro serve quando há uma crença firme de que os processos internos são a maneira de se aproximar do cliente e isto dá resultado.

O eixo vertical mostra como se tomam as decisões na organização. A estabilidade ou flexibilidade da cultura depende se as decisões dentro da organização são tomadas no ponto mais alto possível, neste caso representado pela parte inferior do eixo; ou se são tomadas no ponto mais baixo possível, neste caso representado pela parte superior do gráfico. As organizações do segundo tipo têm o cuidado de dar a seus colaboradores ordens claras para garantir que a tomada de decisões seja em prol do conjunto. [CAMERON e QUINN, 2011] denominaram esse eixo como o de estabilidade/flexibilidade.

Os quatro quadrantes definidos assim são os tipos culturais básicos. Toda organização mostra até certo ponto variantes e integrações destes, mas para os efeitos da análise é importante entender os tipos “puros”.

O “Clã” é fruto de uma cultura onde o foco é interno e as decisões são delegadas às equipes. É capaz de se adaptar muito rapidamente a mudanças e há muita participação coletiva. São capazes de manter a qualidade de um serviço por muito tempo e melhorá-la indefinidamente. É difícil para o clã construir produtos muito grandes e complexos. Este é o arquétipo de cultura que tenta desenvolver o movimento dos agilistas.

A “Hierarquia” é a estrutura que todos conhecemos melhor. O foco é no respeito ao estabelecido e até quem está propondo resiste às mudanças. São organizações muito estáveis que podem criar produtos muito complexos e com altíssima qualidade, como as fábricas de aviões, transportadores espaciais ou certas instituições governamentais. Têm uma tendência a degenerar em burocracia.

O “Mercado” não é uma referência ao mercado externo da organização, embora haja uma relação, mas à empresa em si, que, em todos os seus níveis, opera como um mercado, realizando transações internas e externas para satisfazer o cliente. Em um mercado não há privilégios para amigos e a competência é tudo, aí o nome. As empresas financeiras costumam mostrar exemplos desta cultura.

Por último, uma “Ad-hocracia” é uma cultura que favorece a diferenciação de seu pessoal. Em uma Ad-hocracia é dado mais mérito às invenções e às patentes do que às ascensões e promoções.

Mudar de quadrante, ou mover-se significativamente na direção da mudança, é muito custoso. Para fazer isso de forma consciente, é necessário um diagnóstico profundo e o planejamento das atividades com ainda mais cuidado do que enunciamos no capítulo anterior. É conveniente contratar um consultor que tenha experiência no tema e procurar intensamente a participação de todos os envolvidos.

#### 4.7 Avaliação do Estado de Maturidade

Durante o vir-a-ser do projeto de melhoria de processos, é necessário conhecer os avanços realizados, tanto para poder julgar seu sucesso parcial como para apoiar a mudança com retroalimentação positiva. Esta tarefa é difícil e ingrata. O responsável, ou responsáveis, de levar adiante o projeto de melhorias é o encarregado natural de realizar esta atividade, mas a autoavaliação não é um caminho fácil. Assim como se desaconselha aos médicos sua própria automedicação (e ainda mais aos não médicos) e aos programadores verificar seu próprio trabalho, é necessário contar com uma visão externa, objetiva, para verificar o grau de implementação dos resultados esperados.

O MPS tem um Guia [SOFTEX, 2012c] para a avaliação dos processos de uma organização. Este guia tem como objetivo “permitir a avaliação objetiva dos processos de *software* de uma organização/unidade organizacional<sup>53</sup>; permitir a atribuição de um nível de maturidade do MR-MPS-SW baseando-se no resultado da avaliação; ser aplicável a qualquer domínio na indústria de *software*; ser aplicável a organizações e unidades organizacionais de qualquer tamanho”<sup>54</sup>. O documento esclarece que está destinado fundamentalmente às instituições que realizam avaliações ou implementações do MPS autorizadas pela SOFTEX. Talvez, como em algumas publicidades de televisão, o público deveria ser avisado de que este material é para uso profissional e que a autoavaliação é uma má ideia.

Para sustentar esta afirmação, vejamos o trabalho que a equipe de avaliação precisa realizar: examinar uma planilha, construída pela organização que está sendo avaliada, para verificar a evidência de implementação dos resultados esperados para cada processo e dos Atributos de Processo, segundo sejam designados pelo avaliador líder. Entre outras considerações, o Guia não permite que sócios da organização à qual pertence a unidade organizacional, parentes em algum grau dos sócios da empresa ou da direção participem da equipe de avaliação. O motivo é claro: os conflitos de interesse impedem de exercer a tarefa de julgar o grau de implementação dos resultados esperados. Em nossa experiência, é difícil que os colaboradores da organização sejam totalmente objetivos sem com isso pressupor má vontade da parte deles. Frequentemente, eles são mais exigentes do que os avaliadores externos.

Portanto, o monitoramento e o controle de um plano de melhorias deveria incluir visitas frequentes e periódicas de uma pessoa especialista em avaliações para garantir que não haja surpresas no momento da avaliação formal. Estas avaliações são aconselháveis desde o primeiro momento. Conhecidas como análise de lacunas (*gap analysis*), são avaliações curtas que permitem à organização contar com uma visão objetiva de seu programa de implementação de melhorias de processos e, ao mesmo tempo, receber conselhos de uma pessoa com muito mais conhecimento e diferentes experiências.

---

<sup>53</sup> A frase “unidade organizacional” denota a área de uma organização que representa o escopo da avaliação. Pode ser parte da empresa ou ela toda, uma área particular (por exemplo, a fábrica de testes) ou um setor (novos produtos). O avaliador líder e o patrocinador da avaliação por parte da organização são as pessoas que definem o escopo ao contratar a avaliação.

<sup>54</sup> SOFTEX, 2012i.

## PARTE II – Primeiros Passos

---

### CAPÍTULO 5 - UMA ORGANIZAÇÃO COM PROBLEMAS (NÍVEL G do MPS-SW)

#### 5.1 A Pequena História de Tahini-Tahini

A empresa Tahini-Tahini, internamente conhecida como T<sup>2</sup>, como a chamaremos também a partir de agora, é uma empresa muito pequena de uma cidade da América Latina. Trabalham nela sete profissionais que se conheceram no Instituto Politécnico local. Todos os empregados são sócios com maior ou menor participação e as tarefas administrativas, assim como os cargos representativos (Presidência, Secretárias, Gerências) são exercidos rotativamente. A empresa foi fundada com base no trabalho final de graduação de dois dos engenheiros e, fundamentalmente, oferece certos serviços de folha de pagamento de salários para empresas muito pequenas por meio da *internet*, na modalidade *Software as a Service* (SaaS).

No começo quiseram se chamar ISP, por "*Internet Salários Provider*", que é o nome original do sistema, concebido como uma brincadeira, e que ainda é usado internamente, mas como não puderam obviamente registrar a sigla, mudaram para Tahini-Tahini, que é um jogo de palavras com o nome da farinha com que se fabrica a comida árabe húmus (uma pasta feita com gergelim, óleo e água) e que soa homófono ao inglês "*tiny*" (pequeno). A anedota foi inventada por uma das sócias, Marcela, quando disse juntando o dedo indicador com o médio da mão direita no símbolo universal de algo pequeno: "*Somos uma empresa tiny tiny*".

Os sete profissionais são: Alfredo e Ana, casados entre si, e que idealizaram o serviço; Claudio, irmão mais velho de Ana; Mariano, amigo da infância de Alfredo; Marcela, a amiga do casal desde o ensino médio; e os gêmeos Alberto e Armando Galarraga. Os sete se conhecem muito bem, pelo menos desde o primeiro ano da universidade de cada um. Alguns foram monitores em matérias nas quais os outros cursavam, outros foram colegas de classe. No momento em que os conhecemos, a T<sup>2</sup> está alcançando seus vinte meses de funcionamento contínuo e os gêmeos estão para prestar o último exame para se formar.

No começo, a T<sup>2</sup> tinha um mercado único, o dos negócios de roupa feminina com uma ou duas lojas no máximo. Nesse mercado continuam sendo hegemônicos, mas prevendo que possa surgir alguma concorrência, começaram a expandir o negócio para outros ramos de empresas. Por causa das diferenças entre contratos de trabalho, cada novo ramo que acrescentam significa incorporar mudanças no sistema.

Até aqui, o modelo de negócios da organização consiste em conseguir um cliente para que este pague o desenvolvimento. O cliente fica associado a um "responsável pela conta" designado, em geral, na reunião semanal, normalmente aquele que atendeu a ligação ou fez o contato inicial. Esta pessoa se encarrega de elaborar os requisitos de mudanças e gera uma lista de funcionalidades que devem ser atendidas.

Depois que se confirmam os requisitos com o cliente, o responsável pela conta escolhe uma pessoa da equipe para trabalhar com ele ou ela. Juntam-se na sala de *design* em frente ao mapa do sistema ISP e, usando cartões autoadesivos do tipo *Post-It Notes*, vinculam os requisitos a diferentes módulos. Quando se esgota a lista, revisam os módulos para avaliar o trabalho necessário para modificá-los, a fim de ajustá-los aos novos requisitos. Prepara-se um orçamento muito superficial, que é enviado ao cliente e que, se for aprovado, inicia o que na T<sup>2</sup> passa a ser um projeto.

O projeto, na realidade, só tem de projeto o nome. Há um responsável nominal, mas subentende-se que, se houver problemas ou emergências, todos estarão obrigados a atuar. O responsável define prioridades "na hora". Este processo é assim: quando uma pessoa termina de implementar um requisito, anuncia ao grupo em voz alta. O responsável pela conta, que é o proprietário da tarefa, pega o código do ambiente da pessoa que o desenvolveu e o copia no arquivo próprio, onde realizará os testes do programa. Quem desenvolveu solicita então outra tarefa. Como consequência, as vezes, dois ou três "projetos" disputam os serviços de algum

dos membros da equipe, o que é conhecido internamente como um “leilão”. Quem ganha o leilão vincula um ou mais requisitos à pessoa que ficou livre. O ganhador do leilão é escolhido pela pessoa que vai receber a tarefa.

Quando o responsável pela conta sente que o produto está desenvolvido de forma suficiente, começa a fazer testes *ad-hoc*. Quando não encontra falhas, libera o produto, subindo-o ao *site* da T<sup>2</sup>, de onde o cliente pode usá-lo.

No momento em que nossa equipe é chamada para colaborar com eles para realizar melhorias em seus processos, a T<sup>2</sup> está a ponto de concretizar um lançamento importante para dentistas.

## 5.2 Quem assume a responsabilidade?

Hoje é um grande dia na T<sup>2</sup>. Depois de assistir a uma aula no Politécnico sobre a qualidade total, os gêmeos induziram todos a terem uma reunião plenária com nossa empresa de consultoria, Vitaleria, devidamente autorizada pela SOFTEX como instituição implementadora do MPS (II/MPS). Nossa equipe de consultoria chega cedo à reunião, como de costume, e vai se informando das novidades internas: em vinte e cinco dias deve ser concretizada a entrega do sistema ajustado para dentistas. É sexta-feira e os gêmeos, apesar de terem organizado a reunião, tiraram o dia de folga porque na segunda-feira têm seu último exame. Ana acaba de descobrir que está grávida e ficou em casa, um pouco enjoada. Não virá à tarde, tem uma consulta com sua ginecologista. Alfredo está, mas ficou muito nervoso com a notícia. Marcela vai chegar tarde, avisou que vai passar na casa de Ana e ajudá-la. Claudio está viajando. Por sorte, Mariano não tem problemas e está sentado em sua mesa, trabalhando.

Estamos sentados na mesa da sala de *design*, tomando café com Mariano e Alfredo e a ponto de começar a falar de nossa proposta, quando liga o contato com os clientes dentistas, o Doutor Molar Coroa Ponte, Presidente da Sociedade de Odontologia local. A Associação recebeu uma oferta de uma importante empresa de consultoria internacional para instalar um sistema centralizado de ERP que resolveria também alguns dos problemas que os associados querem resolver com o ISP para dentistas. O Doutor Molar não está muito disposto a ceder à pressão de alguns membros da Sociedade para cancelar o contrato com a T<sup>2</sup>, mas precisa de nossa ajuda para resistir aos ataques. A consultora está fazendo promessas de entrega em tempos impossíveis, ignorando a necessidade de adaptação do ERP em questão, o que, além do mais, certamente obrigará algumas mudanças importantes nas interfaces dos futuros clientes. Por tudo isso, o Doutor Molar pede à T<sup>2</sup> que:

1. Informe o estado atual do projeto de adaptação do ISP aos problemas dos dentistas;
2. Dedique algumas horas, no meio da semana, a fazer uma demonstração do produto no estágio atual, com o objetivo de convencer os associados a esperar o lançamento antes de romper com a T<sup>2</sup>;
3. Estime que porcentagem do produto estará pronta para a demonstração.

Alfredo dá garantias ao cliente, dizendo “*em algumas horas volto a ligar, Doutor, não se preocupe*”, se despede e desliga. Levanta-se e vai até a zona de trabalho. Pergunta a Mariano: “*Você está com o programa de dentistas?*”

Mariano responde que não, e que isso é de exclusiva responsabilidade de Marcela e dos gêmeos. Alfredo pede desculpas por interromper a reunião e liga para o celular de Marcela. Responde um aviso que diz que o celular está desligado. Com certa preocupação, liga para sua casa, onde é atendido pela secretária eletrônica. Deixa uma mensagem pedindo que retornem urgente a ligação. Liga para o celular da Ana e descobre que está redirecionando para o telefone de sua casa. Começa a ficar desesperado.

Este é o momento onde um consultor precisa mostrar para que serve: um cliente está com problemas e sua fisionomia mostra ansiedade, um estado negativo. A ansiedade se manifesta fisicamente, é fácil de perceber e nossos consultores sabem reconhecer bem. Uma fina linha de transpiração aparece no lábio superior de Alfredo que respira com maior dificuldade, suas mandíbulas estão tensas, há um leve tremor em suas mãos ao levantar a xícara de café, que engole com dificuldade, fazendo ruído e pede uma aspirina para sua incipiente dor de cabeça. Ansiedade, sem dúvida.

Nossos consultores sabem que primeiramente deve-se mudar o pensamento negativo, identificar a fonte de preocupação, eliminar o temor que provoca, criar um plano que o leve da insegurança a uma situação criativa, em que haja esperança. Também sabem que precisam tomar a decisão sobre o plano, porque as pessoas ansiosas têm, nesse estado, dificuldade

para decidir, pois o medo que sentem é paralisante, gera pensamentos negativos sobre si mesmos. Devem ser firmes, mas ao mesmo tempo com muita empatia, enfatizando o “ganha-ganha” porque, quem sente ansiedade, tem medo de que os outros percebam suas dificuldades, temor de perder o controle, apesar de ter consciência de estar com dificuldades para pensar.

O primeiro passo para resolução de um problema é identificá-lo com clareza. Nosso consultor-líder, Máximo, faz um resumo de seu entendimento do problema: há um cliente importante (o Doutor Molar) em situação crítica (a pressão externa da consultora que oferece o ERP) e é preciso dar uma resposta sobre o estado do projeto (as duas perguntas). Esse é o detonante. O problema é que não há nenhuma forma objetiva de entender o estado, porque as pessoas que estão a cargo, por uma série de circunstâncias totalmente fortuitas, não estão presentes. Alfredo concorda levemente com a análise apresentada. Mariano concorda com mais interesse do que Alfredo.

O segundo passo é encontrar as causas-raiz do problema. Para garantir a participação e compromisso de Alfredo e Mariano, Máximo se levanta e vai à lousa. Com um marcador divide a lousa em dois e, em um dos lados, desenha um diagrama de espinha de peixe (Figura 5.1) com quatro “espinhas” e coloca na cabeça do peixe o nome que dá ao problema: “não há forma objetiva de entender o estado do projeto”. As espinhas recebem seus nomes: “Processo”, “Pessoal”, “Ferramentas” e “Capacitação”. Os nomes e a quantidade de espinhas podem mudar, sendo esta escolha em parte dependente da percepção do problema ao enfrentá-lo. De qualquer modo, esta é uma emergência e é mais importante mostrar iniciativa do que precisão total. Agora que já conquistou a audiência, Máximo começa a perguntar aos dois sócios usando outras técnicas.

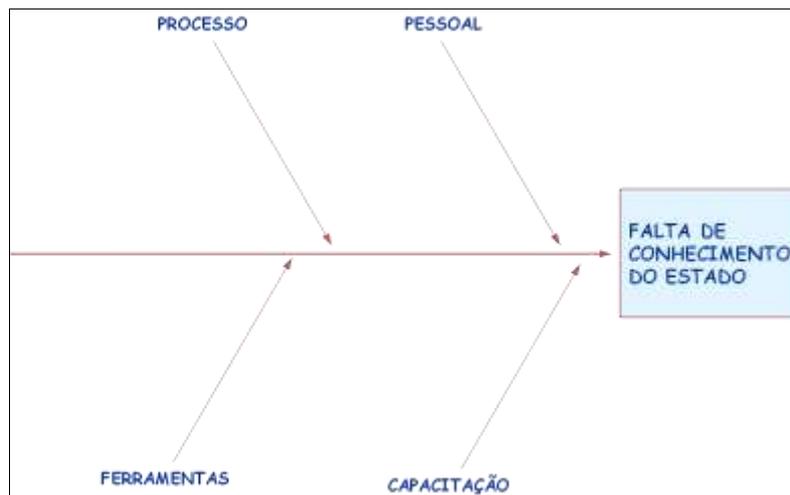


Figura 5.1: Causas da Falta de Conhecimento do Estado do Projeto

Primeiro começa com os “Cinco Porquês”.

“Falta Conhecimento do Estado das Tarefas. Por quê?”, pergunta Máximo.

Não se trata de identificar mais sintomas, ou se a existência do problema é reconhecida. Trata-se de indagar sobre a natureza das causas. À pergunta “Por que falta conhecimento do estado do projeto?” é incorreto responder com “porque não sabemos responder aos clientes quando nos perguntam sobre isto”. Se a resposta obtida também responde à pergunta “Como sei que falta conhecimento do estado do projeto?” é preciso descartá-la. O que se procura é a origem do problema, não outras manifestações dele mesmo. Por isso Máximo escolheu as categorias correspondentes no diagrama. As respostas às perguntas são orientadas a esses eixos, as “espinhas” do peixe no diagrama.

Como o pessoal da empresa é um grupo de engenheiros, no máximo faltando uma matéria para se formar, nem o pessoal nem a capacitação merecem grande atenção. O pequeno grupo centra-se em processos e ferramentas. A primeira resposta é que os processos atuais não registram o estado. Continuam os questionamentos sobre os seguintes porquês até se detectar a causa-raiz. Pode ser que seja necessário perguntar cinco vezes “Por quê?” ou pode ser que se chegue antes à causa-raiz. É fácil detectar uma causa-raiz porque ela geralmente oferece imediatamente uma solução. Por exemplo, se eles respondem ao segundo porquê dizendo que

os processos existentes não estão preparados para identificar tarefas, responsáveis, nem o estado do projeto, a solução é óbvia, apesar de não ser fácil de implementar: modificar os processos para que se possa identificar as tarefas, os responsáveis e o estado do projeto. Continuando com as perguntas, Máximo passa à espinha das ferramentas. Também aqui é fácil ver que as ferramentas com que se conta não dão o suporte necessário para o processo que querem implementar.

Depois que uma das espinhas gera o imediato reconhecimento da solução (por exemplo, “falta um processo” gera a resposta “coloquemos um processo” e o mesmo com as ferramentas), o exercício é suspenso e passamos à discussão das soluções. A aplicação dos cinco porquês não é o único caminho possível, já que parte de nosso repertório é uma alternativa chamada as Oito Disciplinas<sup>55</sup>:

- D1: Formação de uma equipe de especialistas que em seu conjunto consigam atender todas as funções;
- D2: Identificação e definição completa do problema;
- D3: Implementação e verificação de uma ação de contingência provisória para que o dano não se propague;
- D4: Identificação e verificação da causa-raiz;
- D5: Determinação e verificação de ações corretivas permanentes;
- D6: Implementação e verificação das ações corretivas permanentes;
- D7: Prevenção da recorrência do problema e/ou sua causa-raiz; e
- D8: Reconhecimento dos esforços da equipe.

Como nossos consultores são ecléticos, não desdenham nenhum ensinamento que seja útil. Neste caso, Máximo reconhece a necessidade de aplicar D3: é necessário deter a propagação do efeito do problema, antes de começar a escrever o processo que impossibilite sua recorrência, pois senão será tarde demais, uma vez que já terão perdido o cliente. Máximo se coloca a trabalhar com os sócios para fazer um plano de emergência. Coordena com Alfredo. para que vá buscar imediatamente sua mulher e Marcela. Mariano e Máximo se comunicam com os gêmeos e decidem fazer uma reunião de investigação, assim que terminarem de festejar, terça-feira à primeira hora. Mariano liga para o Doutor Molar e anuncia que na quarta-feira, antes do meio-dia, ele receberá uma proposta completa para chamar os interessados a participarem de uma demonstração, proposta que incluirá a proporção do sistema que será demonstrada.

Depois de suficientemente detida a hemorragia, e já identificadas claramente as causas, Mariano e Máximo definem um processo que vai solucionar a causa e evitar que se repita. Nosso homem em T<sup>2</sup> parte para definir, seguindo o método, as características desejáveis do processo:

- Evitar que o problema se repita no futuro
- Ser muito fácil de implementar (dois ou três dias na T<sup>2</sup>)
- Contar com suporte em *software* existente na empresa, ou fácil de conseguir (*freeware*, *Open software* ou semelhante) ou desenvolvido internamente (sobretudo na nuvem).

Nosso consultor considera que Mariano é uma pessoa inteligente e com conhecimentos, não o subestima nem é condescendente. Tampouco dita os resultados, mas quando aparece a oportunidade (um problema para o qual conhece uma resposta) faz propostas para que ele as avalie. Com a análise inicial que realizaram com a espinha de peixe, sabe que a base de todas as propostas sempre precisa ter em seu centro um sistema que permita ingressar tarefas e indicá-las, ao mesmo tempo em que vai atualizando permanentemente seu estado. O método de funcionamento é simples e, o que é muito importante, pode ser considerado uma evolução do que fazem hoje em dia. Quando Mariano descreve a “conclusão” pelo qual se designam prioridades, Máximo associa com o livro de *Kanban* e *Scrum* e sugere dar uma lida, juntos, em [KNIBERG e SKARIN, 2010]. Uma rápida leitura dos materiais de *Kanban* faz com que tomem conjuntamente a decisão de procurar na rede ferramentas de *software* que se ajustem às necessidades da T<sup>2</sup> para colocar o painel *kanban* na nuvem ou em servidores locais, gratuitamente. Há elementos suficientes para que Mariano se entretenha vários dias

---

<sup>55</sup> [http://en.wikipedia.org/wiki/Eight\\_Disciplines\\_Problem\\_Solving](http://en.wikipedia.org/wiki/Eight_Disciplines_Problem_Solving)

comparando-os. Revisam juntos vários deles e, apesar de não tomarem a decisão, avançam muito no tema.

Três horas mais tarde, Mariano e nosso consultor se despedem. Receberam uma ligação tranquilizadora de Alfredo: Marcela e Ana adiantaram a visita à ginecologista, onde está proibido o uso de celulares, por isso não havia resposta de nenhuma das duas. Tudo está bem no universo da T<sup>2</sup>. Máximo se despede dizendo a Mariano que, se quiserem continuar com um contrato permanente, as horas que trabalhou serão consideradas investimento de *marketing*. Mariano não dá garantias, mas afirma que discutirão na reunião semanal e que a proposta contará com seu apoio.

#### O QUE ACONTECEU ATÉ AGORA

1. O consultor Máximo estabeleceu o contato inicial com o cliente, coincidentemente com um problema grave e facilitou sua identificação.
2. Introduziu uma primeira técnica de análise de causa, o diagrama de Ishikawa.
3. Utilizando a técnica chegou, junto com os clientes, à conclusão de que seus processos deixavam muito espaço para problemas como o detectado, a falta de controle das tarefas.
4. Apesar de ter um diagnóstico concreto que já apontava para a melhoria de processos, Máximo se concentrou no problema imediato para evitar um maior, começando por definir as características (ou atributos) desejáveis da solução.
5. Máximo sugeriu o método Kanban, sem imposição, e foi escutado.

### 5.3 Mostrando a Carga de Trabalho e o Estado das Tarefas

Na segunda-feira seguinte, nossa empresa de consultoria é chamada por Alfredo, que pede para Máximo ser enviado como consultor, para que facilite a reunião semanal com a presença de todos os membros da cooperativa. O objetivo é armar o painel *kanban* do que ainda não foi realizado até esse momento no projeto e carregá-lo no *software* que já baixaram no fim de semana. Alfredo afirma que os honorários devem incluir, também, as quatro horas de trabalho da sexta. Ficamos gratamente surpresos, porque percebemos que os jovens empreendedores entenderam a mensagem e estão muito satisfeitos em poder contar com nossos serviços.

Chegada a manhã de terça-feira, todos os membros da equipe estão sentados ao redor da mesa de *design*. Ana está um pouco pálida, a gravidez ainda não foi totalmente assimilada, mas nos demais há efervescência e energia para dar e vender. Apreciando o lugar, nosso consultor elogia a disposição da sala: há quatro lousas para serem usadas com marcadores, uma delas ainda mostra o diagrama de espinha de peixe, as outras mostram a arquitetura conceitual e a outra, a arquitetura modular do ISP. A quarta lousa é eletrônica e serve para registrar decisões mediante o simples apertar de um botão que transcreve o escrito na lousa a um arquivo em PDF que pode ser impresso. É a que o grupo usa para registrar a memória das reuniões semanais. O modelo da arquitetura modular está “decorado” com notas autoadesivas, representando “histórias” do cliente a serem implementadas.

Máximo se dirige até essa lousa e chama a atenção para as notas autoadesivas, solicitando uma explicação de seu uso. Marcela responde, detalhando como se movem as notas da coluna da esquerda, agora vazia, aos diferentes módulos, para mais adiante chegar à conclusão. Máximo tira uma fotografia da arquitetura com seu *laptop* e aproveitando o projetor, mostra à audiência. Máximo indica que as notas designadas aos módulos serão o ponto de partida. Esclarece que, apesar de ser possível usar o *software* para *Kanban* que já instalaram, acha que o melhor é fazer o exercício “à moda antiga” para que a percepção seja mais completa.

Pega uma pilha de notas autoadesivas, entrega-as aos participantes e diz: “Cada um precisa ficar com umas cinco ou seis notas”. Depois que todos têm seu material, por ordem, começando pela Ana e no sentido dos ponteiros do relógio ao redor da mesa, Máximo dita rapidamente os conteúdos de cada nota, pedindo que escrevam claramente e no centro do papel. Como vai projetando-os ao mesmo tempo, não se detém para esperar que sejam copiados totalmente, de modo que em poucos minutos todas as histórias do cliente, pelo menos seus nomes, estão copiadas. Ele as recolhe e coloca na lousa que havia usado na sexta. Apaga seu diagrama e desenha um painel *kanban* elementar, por enquanto com três

colunas: Solicitadas (Pedidas), à esquerda, onde estão todas as histórias. Completas, à direita, que está vazia. A grande coluna do meio está vazia e sem título, por enquanto (Figura 5.2).

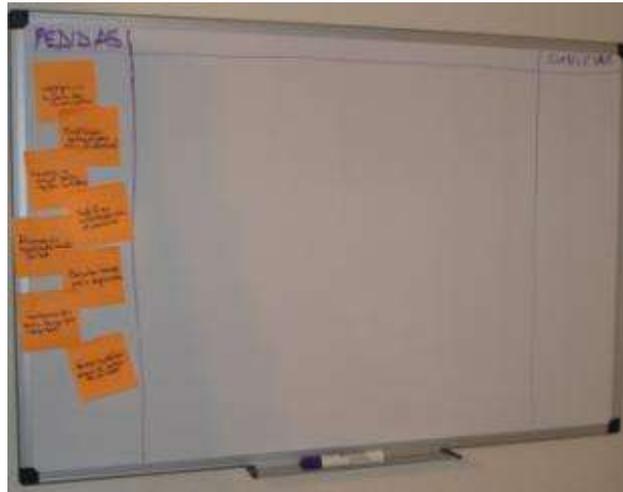


Figura 5.2: Painel *kanban* elementar

Máximo solicita então que cada um, na sua vez, diga uma tarefa derivada da implementação dessa função. Ana começa dizendo “Codificar os testes” e Máximo escreve isso em uma nota autoadesiva, que em seguida gruda debaixo da original. Marcela acrescenta, quase sem pausa: “Criar os testes”. Isto também vai para a lousa. É a vez de Alberto, que vem sacudindo a cabeça: “Investigar a história” diz e Armando acrescenta: “...com o cliente e dividi-la em funções e características”. Todos dão risada, claro. Máximo pergunta então qual é o ciclo de vida de uma história. O resultado é que ela é analisada, desmembrada em funções, os testes de cada função são construídos, ela é codificada e testada. Máximo nota que as funções podem já estar claras e que nem sempre, necessariamente, acontecem análises. Os presentes concordam. Modifica então o painel para incluir essa informação (Figura 5.3) e tira as notas que havia acrescentado, já que as tarefas são parte do ciclo.

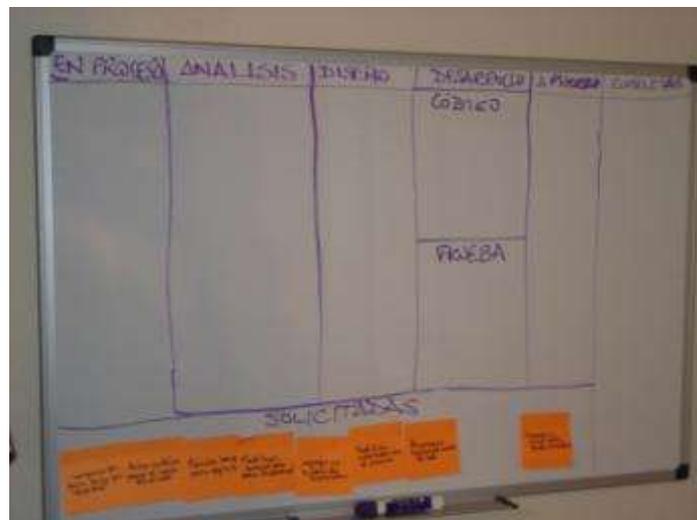


Figura 5.3: Painel *kanban* com ciclo de vida das histórias -1-

“Quem é o dono do produto?” pergunta Máximo.

“Eu”, diz Marcela.

“Não é o Doutor Molar?” volta a perguntar Máximo.

“Não”, diz Marcela, “o Doutor não conhece os detalhes das folhas de pagamentos, eu estudei a legislação vigente e sou quem sabe como se deve fazer para pagar os salários e quais opções possui o dentista individual”.

“Parabéns!” diz Máximo. “Então, o primeiro passo é que entre todos, e sob sua direção, priorizemos as histórias”.

Marcela se levanta e começa a mover as histórias que agora estão na fila mais abaixo do painel. Máximo interrompe.

“Conte-nos seu critério, Marcela”.

Marcela começa a explicar as prioridades e os gêmeos intervêm quase em coro, recordando a todos a iminência da demonstração pedida pelo grupo de clientes. Logo todos estão de pé ao redor da lousa e movendo as notas para frente e para trás. A primeira prioridade é para a história “Modificar Interfaces com o Usuário” pelo impacto que tem em uma demonstração. Quando finalmente a lista se estabiliza, Máximo aprova.

“Perfeito, temos um Backlog do produto. Agora, vamos estimar o tamanho.”

Máximo escreve na lousa eletrônica uma tabela que diferencia entre tamanhos de tarefa (Tabela 5.1). Esclarece que as definições não são do que corresponde a “tamanho” no idioma português, mas que este é um bom começo. Enfatiza, igualmente, que a tabela visa que as categorias, apesar de serem atributos ordinais, tenham relação entre si, de modo que a diferença entre “muito simples” e “simples” acabe sendo comparável com a diferença entre “normal” e “mais que o normal”. O propósito é que as categorias, quando mais adiante forem substituídas por medições objetivas, possam servir de base para elas. Com esta tabela na frente de todos, pega a primeira das histórias “Modificar interfaces com o Usuário”. Desenha um quadrado em cada um dos cantos e com um lápis escreve “tamanho” no quadrado superior da esquerda (Figura 5.4) e pede que façam uma estimativa do tamanho desta história. Há um grande silêncio, quebrado por Ana, que, sacudindo a cabeça em negação, diz:

“Esta não é uma história de usuário, é uma tarefa transversal a todas as histórias.”

Tamanho	1	2	3	4	5	6	7
Categoria	muito simples	simples	menos que o normal	normal	mais que o normal	complicado	muito complicado
funções derivadas	1	2	3	4	5 a 6	7 a 10	mais de 10
Dias	No dia				1 a 2	2 a 4	mais de 4

Tabela 5.1: Tamanhos de Tarefas

Com o consentimento de todos, Máximo a coloca na coluna Em Processo na metade de baixo e pega a próxima história da fila de Solicitadas e lê em voz alta.: “Acrescentar um módulo de folha de pagamento de Horas Extras”.



Figura 5.4: História no Painel kanban

Volta a parar na frente da tabela de tamanhos e pede que se faça uma estimativa do tamanho desta história. Há uma discussão de poucos minutos, antes que os participantes concordem de que o tamanho é “mais que o normal”. Máximo pergunta, então, se não seria melhor dividir essa história em partes (sub-histórias), pois são cinco ou seis funções derivadas que estão sendo pensadas.

Pega um pacote de notas autoadesivas de cor diferente do que já foi usado, e repete o pedido de que sejam enunciadas as funções derivadas dessa história. Desta vez escreve os requisitos derivados à medida que Marcela, com a ajuda de todos, vai expondo: “Acrescentar Opção de Carregar Horas Extras ao Módulo de Folha de Pagamento Mensal”, “Acrescentar uma Fórmula

de Cálculo ao Módulo do Funcionário”, “Modificar Sequência de Folha de Pagamento para Incluir Horas Extras”, “Modificar a Folha de Pagamento do Salário Extra para Admitir Horas Extras” e, claro, “Modificar Interfaces com o Usuário”.

“Agora temos um épico<sup>56</sup>!”, intervêm os gêmeos.

“Épicos, histórias, sub-histórias, temas, funções, características, casos de uso, casos de usuário, são todos nomes que damos às coisas. Vocês escolhem os que servirem. O importante é que tenhamos claro que há uma raiz e que dessa raiz são ‘derivados’ outros requisitos e que desses podem derivar outros mais, formando uma hierarquia. Quando os requisitos chegam a um nível onde sabemos que podem ser implementados, derivaremos desse nível as tarefas. Todo este conjunto é uma hierarquia que nos serve para identificar as histórias e os usuários que as geram. Poderíamos usar um padrão que diga – Como usuário X, neste caso um dentista, quero poder ‘fazer Y’, neste caso acertar as horas extras de trabalho do meu pessoal. Mas o importante é que tenhamos claro o que é preciso fazer e controlar isso”, diz Máximo.

Baseados nos dados que são obtidos, os gêmeos estimam que, se começarem a trabalhar nesse momento, com a ajuda de Marcela e Ana, podem ter a ‘demo’ a tempo para apresentá-la na quinta-feira depois do horário de trabalho, com as funções que não exigem novas APIs. Alfredo respira aliviado e Claudio liga para o Doutor Molar para transmitir as novidades.

#### O QUE ACONTECEU ATÉ AGORA

6. Máximo induziu o uso do painel *kanban*, sem mandar.
7. Introduziu os conceitos de sub-história e estimativa por tamanho.
8. Houve uma clara definição da abrangência do trabalho a realizar, obtido a partir da lista de histórias, com sua correspondente estimativa de tamanho.
9. Foi feito um desmembramento das histórias, onde seu tamanho garantia a sua utilidade.
10. Os conceitos de história, sub-história, desagregação e tarefa foram entendidos na prática e foram aplicados. Foi possível distinguir “tarefa” de “sub-história”.

#### 5.4 Planejamento, Monitoramento e Controle do Projeto em Doses Homeopáticas

A reunião continua a pedido de Máximo, por mais meia hora. Antes de deixar a T<sup>2</sup> dedicada ao que tinha de fazer, Máximo quer deixar clara a definição de PRONTO para cada etapa do painel. Então pergunta:

“Quando se pode afirmar que a análise da história está concluída?”

As caras dos assistentes dizem tudo: nunca tinham pensado nisso. Eles se aventuram levemente sobre algumas definições, até que Mariano acerta no ponto: a realidade é que a análise, apesar da tradição ao contrário, não é feita separadamente. É parte do *Design*. Máximo apaga a coluna de ANÁLISE do painel e redistribui as demais, cada uma tem duas subcolunas: EP, que se lê Em Processo, e PRONTO. As definições de “pronto” para *Design*, Código, Desenvolvimento de Testes e Completas ficam assim definidas:

*Design*: o *design* está PRONTO quando há um diagrama de mudança de estados que foram aprovados pelo Dono do Produto, por quem está encarregado do desenvolvimento do código e por quem está encarregado do desenvolvimento dos testes.

O Código está PRONTO quando foi feita a revisão com um colega e realizados os testes unitários do próprio codificador e não foram encontrados defeitos; se havia, foi demonstrado que estes foram corrigidos.

O Desenvolvimento de Testes está PRONTO quando foi feita a revisão de todos os testes com um colega e foram carregados os testes unitários na ferramenta de monitoramento dos testes automáticos.

<sup>56</sup> <http://www.mountaingoatsoftware.com/blog/stories-epics-and-themes>

A história em si está PRONTA quando todos os testes foram feitos, foram integrados e não se encontraram defeitos nem com os novos testes nem com os testes de regressão.

Tudo isto induz a uma última mudança do dia no painel *kanban*: a coluna A Testar passa a ser chamada de Integração e Completa passa a ser chamada PRONTA. Máximo apaga todo o painel, introduz uma nova coluna à esquerda chamada Em Espera, onde se colocarão as duas histórias de maior prioridade, e as colunas *Design*, Desenvolvimento, Integração e Prontas. O painel é visto agora como na Figura 5.5.

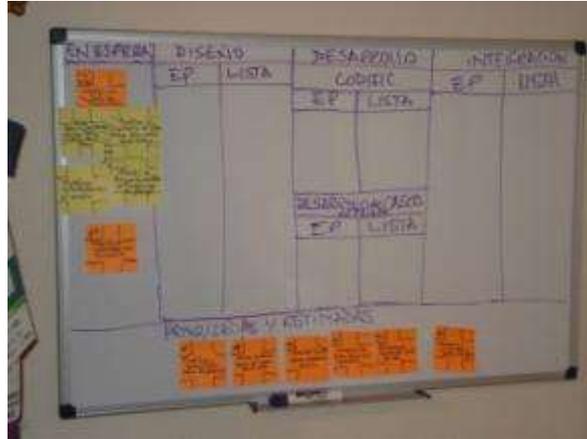


Figura 5.5: Painel *kanban* com ciclo de vida das histórias -2-

Continuando, Máximo trabalha com Marcela e os gêmeos para definir o procedimento para o uso do painel nas atividades dos próximos dias. Cada vez que alguém estiver livre, ele se aproximará de Marcela e escolherão, juntos, uma tarefa para realizar. Essa tarefa será retirada da coluna “Em Espera” e será colocada na coluna EP de “*Design*”. A pessoa que se encarregar da tarefa escreverá seu nome no quadrado superior direito da nota correspondente. No quadrado inferior esquerdo colocará a data e a hora de começo e no inferior direito a data e hora de término, quando a tarefa estiver concluída. Depois que estiver completada segundo o autor, espera-se que alguém assuma a tarefa de codificar e desenhar os casos de teste. Se isso for feito pela mesma pessoa (embora seja melhor envolver alguém mais, para ter “mais olhos” no projeto), ela revisará com Marcela e a nota passará a EP debaixo de “Codificação” e uma cópia a EP sob “Desenvolvimento de Casos de Teste”. Sempre se evitará que a pessoa que codifique desenvolva os casos de teste. Por último, define-se que a integração será realizada pela pessoa responsável pelo projeto, neste caso, Marcela. Todos satisfeitos encerra-se a sessão, e ainda resta pouco mais de uma hora para trabalhar pela manhã.

#### O QUE ACONTECEU ATÉ AGORA

11. Máximo mostrou que o uso do painel *kanban* é dinâmico e que pode ser modificado.
12. Ficou esclarecido o que significa uma tarefa estar PRONTA, para cada etapa do seu ciclo de vida.
13. Visualiza-se facilmente o ciclo de vida de uma história no painel.
14. O controle do estado geral de uma história e as tarefas associadas podem ser lidas no painel.

### 5.5 O Cliente quer Mudanças

Na quinta-feira da mesma semana, os gêmeos vão com seus *tablets* e *laptops* à sala de conferências da Sociedade de Odontólogos Profissionais em Atividade (SOPA) local. A demonstração do produto convence quase todos, e a simpatia e juventude dos profissionais fazem o resto. Um memorando de acordo é assinado entre a T<sup>2</sup> e a SOPA. Alguns dos menos entusiasmados pedem que se demonstre a capacidade de modificar o serviço, para o qual encarregaram algumas mudanças, a maioria cosméticas, mas algumas afetam os módulos de cálculo. A T<sup>2</sup> tem uma semana para responder com um plano e uma proposta de honorários. Estabelecida uma boa relação entre a T<sup>2</sup> e Máximo, ele é chamado diretamente para que facilite outra reunião para dar a resposta.

Máximo preferia seguir com seus planos de implementação das práticas do MPS, mas os negócios de seus clientes sempre são prioridade para ele, como uma política da Vitalería. De todas as maneiras, nenhuma parte do negócio de *software* é totalmente alheia ao MPS, de modo que sempre usará soluções nascidas de suas práticas. Além disso, nem tocou no tema do MPS com a T<sup>2</sup>, algo que pensa fazer mais para frente, quando chegar a ocasião para uma avaliação MPS bem-sucedida. Assim, ele organiza seu calendário e, mudando uma reunião aqui e outra ali, na tarde de sexta-feira está de novo na sala de *design* da T<sup>2</sup> com os mesmos atores da terça-feira.

Máximo repassa com os participantes os passos que levaram ao pequeníssimo plano de implementação de terça-feira, mas utiliza desta vez a ferramenta de *software* (*sprint.er*) que instalaram na T<sup>2</sup>. A sequência de passos é a mesma: identificar o *Backlog*, fazer a distribuição dos componentes, organizar as histórias por prioridades, estimar o volume de trabalho, desagregar as histórias muito volumosas em pequenas histórias e tarefas derivadas, revisar e fechar. Mas esta é a primeira proposta com desenvolvimento sob medida que a T<sup>2</sup> vai assinar; até aqui só venderam serviços de um produto relativamente estável com modificações que não alteravam o esquema de trabalho, baseados sobretudo em correções no código para eliminar defeitos ou adequações a mudanças na legislação. É por isso que a equipe não possui um padrão de apresentação de planos para propostas.

Máximo introduz agora a planilha para a definição de histórias que a Vitalería desenvolveu. Os assistentes sugerem pequenas mudanças e a planilha é aprovada, aceita e adotada (Figura 5.6). Utilizando as facilidades de *design* na ferramenta *sprint.er* incluem os campos necessários. Sobre a base da planilha, é rearmado o *Backlog*, agora muito mais completo e é realizada a estimativa com *sprint.er*. Os resultados projetam a necessidade de utilizar três *sprints* de duas semanas. O primeiro implementará as mudanças mais profundas, o segundo introduzirá as mudanças de interface sugeridas, e o terceiro servirá de garantia de estabilidade. Sobre a base do planejado são acrescentados custos à estimativa.

PLANILHA DE DEFINIÇÃO DE HISTÓRIAS						
ID	NOME	IMPORTÂNCIA	TAMANHO	VALIDAÇÃO	RISCOS ASSOCIADOS	CAPACIDADES NECESSÁRIAS
1						
2						
3						
4						
5						

Figura 5.6: Planilha para Definição de Histórias

Para poder medir o possível impacto de algum risco que se materialize, Máximo introduz uma planilha de definição e análise de riscos (Figura 5.7). Toda esta documentação é simples e não toma tempo extra para ser preenchida, mas joga muita luz sobre o projeto, obrigando os membros da equipe a pensar no que estão fazendo, usando outras faculdades além de sua capacidade de construir<sup>57</sup>.

<sup>57</sup> [DE BONO, 1985]. O livro faz referência a cinco maneiras de pensar sobre um problema. Informação (Branco), Emoções (Vermelho), Discernimento (Preto), Otimista (Amarelo), Criativo (Verde). Também há um chapéu Azul para discutir as regras do jogo em si.

PLANILHA DE DEFINIÇÃO E CONTROLE DE RISCOS							Nome do Projeto			
							Data			
							Preparado por			
ID - Identificador do risco Descrição do Risco - problema potencial (prioridade e consequência) Prob - probabilidade de que o risco se transforme em um problema Prioridade neste análise - tempo por exposição Ação - ações a tomar para lidar com o risco							Adicione as colunas que sejam necessárias para monitorar a evolução dos riscos. Perd - Perda - potencial relativo à perda (monetária) ou um número entre 1 e 100 Prioridade a última vez - reflete a mudança iminente Quem - pessoa responsável pelas ações			
							Exp - Exposição - produto entre prob e perd			
							#Vezes na lista - resistência às ações			
							Quando - data na que se realizou as ações			
ID	Descrição do Risco	Prob	Perd	Exp	Prioridade neste análise	Prioridade na última vez	# Vezes na lista	Ação	Quem	Quando
1	Condição									
	Consequência									
2										
3										
4										
5										
6										
7										
8										
9										
10										

Figura 5.7: Planilha para Definição e Análise de Riscos

Chegado este ponto, Máximo introduz a planilha que a Vitaleria usa para suas propostas de consultoria, que segue uma estrutura que contém o *Backlog* do produto, o plano de *sprints* e o painel original do primeiro *sprint*. Também são explicitados nela tempos e custos, assim como um resumo extraído da planilha de análise de riscos e uma lista das responsabilidades mútuas (ver Figura 5.8).

<ul style="list-style-type: none"> <li>• Introdução</li> <li>• Resumo Executivo da Proposta</li> <li>• Escopo do Projeto               <ul style="list-style-type: none"> <li>– Método de Desenvolvimento das atividades                   <ul style="list-style-type: none"> <li>• Descrição geral</li> <li>• Ciclo de vida de cada atividade</li> <li>• Preparação do pessoal envolvido</li> </ul> </li> <li>– Histórias &lt;<a href="#">link com documento Histórias</a>&gt;                   <ul style="list-style-type: none"> <li>• Identificador</li> <li>• Nome</li> <li>• Importância</li> <li>• Tamanho</li> <li>• Validação</li> <li>• Riscos &lt;<a href="#">link com a planilha de riscos</a>&gt;</li> <li>• Capacidades Requeridas</li> </ul> </li> <li>– Calendário de Trabalho                   <ul style="list-style-type: none"> <li>• Sprint 1</li> <li>• Sprint 2</li> <li>• Sprint ...</li> <li>• Sprint de estabilização</li> </ul> </li> <li>– Orçamento econômico e financeiro</li> </ul> </li> <li>• Riscos previstos</li> <li>• Obrigações mútuas               <ul style="list-style-type: none"> <li>– Comunicações</li> <li>– Entregas</li> <li>– Aprovações</li> <li>– Pagamentos</li> </ul> </li> </ul>
--

Figura 5.8: Planilha de Proposta de Projeto

A nova estrutura, com as correspondentes conexões aos produtos vinculados (*Backlog* na ferramenta de *software* e planilha de riscos), constitui uma prova útil das resoluções aprovadas em equipe. Em menos de uma hora de trabalho sobre esta planilha chegam a um acordo e se dispõem a mandá-la ao cliente. Máximo pede duas coisas: uma é que os presentes se comprometam a assinar sua concordância com a proposta, a segunda é que o cliente faça o mesmo. Os gêmeos acham um pouco estranho, mas Máximo explica que os clientes que não querem assinar não querem se comprometer, sendo que o resultado é que não haverá acordo no momento da entrega.

**O QUE ACONTECEU ATÉ AGORA**

15. Máximo incorporou melhorias no documento do *Backlog*, tornando-o mais sólido e mais útil.
16. Utilizando uma planilha de riscos foram analisados os possíveis efeitos indesejados associados com as histórias do projeto.
17. Foi incorporada uma planilha de propostas que permite reunir em um documento dinâmico o *Backlog* e o orçamento, acrescentando as responsabilidades das partes contratuais.
18. Os compromissos mútuos de cliente e fornecedor ficam documentados, em especial o escopo do projeto, representado pelo *Backlog*.

Sem mais nada a tratar, a reunião termina. Máximo é chamado a parte por Claudio. Na sala das conversas privadas, explica que a empresa tem poucos recursos e que ele gostaria muito de contar com seu apoio como consultor, mas que não possuem orçamento para isso. Talvez, apesar de tudo, esta intervenção tenha sido a última.

Máximo sorri, porque esperava que este problema surgisse.

*“Justamente sobre isso queria que conversássemos”, diz. “Há um programa<sup>58</sup> que permite investir em qualidade e receber recursos do estado para isso”.*

Máximo explica o programa e os olhos de Claudio se abrem cada vez mais.

*“De nossa experiência, apesar de ser relativamente novo no ambiente, o modelo mais adequado para empresas como a T<sup>2</sup> é o MPS.”*

*“O que é o MPS?”* pergunta Claudio.

Máximo dá uma explicação precisa sobre as vantagens do modelo MPS, matizada com conceitos como avaliações, evidência direta e outros termos que hoje são novos na T<sup>2</sup>, mas que vão se converter, com a passagem do tempo, em moeda corrente. Claudio fica com a tarefa de contar o resumo do que foi conversado a todos os sócios, enquanto que Máximo precisa falar na Vitalería para que a T<sup>2</sup> seja incluída na lista de empresas que aspiram ao apoio para implementação do MPS.

### 5.6 Avanços na implementação do MPS-SW

Algumas semanas depois, a T<sup>2</sup> está preparando sua avaliação de Nível G do MPS-SW. A primeira atividade no plano que a Vitalería preparou com eles é a realização de uma análise de lacunas (*gap analysis*) que avalia como os processos estão implementados e os resultados esperados de implementação de processos do MR-MPS-SW. Para isso, é útil contar com a lista dos resultados esperados no Nível G (Tabela 5.2) e compará-los com o que está implementado na T<sup>2</sup>:

#### Gerência de Projetos (GPR) no Nível G

- |             |   |
|-------------|---|
| <b>GPR1</b> | <b>O escopo do trabalho para o projeto é definido;</b>  |
| <b>GPR2</b> | <b>As tarefas e os produtos de trabalho do projeto são dimensionados utilizando métodos apropriados;</b>  |
| <b>GPR3</b> | <b>O modelo e as fases do ciclo de vida do projeto são definidos;</b>   |
| <b>GPR4</b> | <b>O esforço e o custo para a execução das tarefas e dos produtos de trabalho são estimados com base em dados históricos ou referências técnicas;</b> |
| <b>GPR5</b> | <b>O orçamento e o cronograma do projeto, incluindo a definição de marcos e pontos de controle, são estabelecidos e mantidos;</b>                     |

<sup>58</sup> Em vários países da América Latina isto é uma realidade (*external funding*). A forma que o programa é adotado varia de país para país, por isso evitamos neste texto fazer uma referência mais explícita. Os programas costumam pagar às empresas uma parte substancial das despesas em consultoria quando ela é avaliada ou certificada sob um modelo internacional, como o MPS, CMMI ou ISO.

<b>GPR6</b>	<b>Os riscos do projeto são identificados e o seu impacto, probabilidade de ocorrência e prioridade de tratamento são determinados e documentados;</b>
<b>GPR7</b>	<b>Os recursos humanos para o projeto são planejados considerando o perfil e o conhecimento necessários para executá-lo;</b>
<b>GPR8</b>	<b>Os recursos e o ambiente de trabalho necessários para executar o projeto são planejados;</b>
<b>GPR9</b>	<b>Os dados relevantes do projeto são identificados e planejados quanto à forma de coleta, armazenamento e distribuição. Um mecanismo é estabelecido para acessá-los, incluindo, se pertinente, questões de privacidade e segurança;</b>
<b>GPR10</b>	<b>Um plano geral para a execução do projeto é estabelecido com a integração de planos específicos;</b>
<b>GPR11</b>	<b>A viabilidade de atingir as metas do projeto é explicitamente avaliada considerando restrições e recursos disponíveis. Se necessário, ajustes são realizados;</b>
<b>GPR12</b>	<b>O Plano do Projeto é revisado com todos os interessados e o compromisso com ele é obtido e mantido;</b>
<b>GPR13</b>	<b>O escopo, as tarefas, as estimativas, o orçamento e o cronograma do projeto são monitorados em relação ao planejado;</b>
<b>GPR14</b>	<b>Os recursos materiais e humanos bem como os dados relevantes do projeto são monitorados em relação ao planejado;</b>
<b>GPR15</b>	<b>Os riscos são monitorados em relação ao planejado;</b>
<b>GPR16</b>	<b>O envolvimento das partes interessadas no projeto é planejado, monitorado e mantido;</b>
<b>GPR17</b>	<b>Revisões são realizadas em marcos do projeto e conforme estabelecido no planejamento;</b>
<b>GPR18</b>	<b>Registros de problemas identificados e o resultado da análise de questões pertinentes, incluindo dependências críticas, são estabelecidos e tratados com as partes interessadas;</b>
<b>GPR19</b>	<b>Ações para corrigir desvios em relação ao planejado e para prevenir a repetição dos problemas identificados são estabelecidas, implementadas e acompanhadas até a sua conclusão.</b>

**Tabela 5.2 Processo GERÊNCIA DE PROJETOS no Nível G [SOFTEX, 2012a]**

O *Backlog* permite conhecer o escopo do trabalho (GPR1) e junto com o painel define as tarefas e os produtos associados, com seus correspondentes esforços que foram estimados pelo consenso entre os participantes (GPR2). O plano dos *sprints* define o modelo a seguir no projeto e as fases correspondem aos *sprints*, da mesma forma que o painel mostra nas tarefas, o ciclo da vida de cada história (GPR3). Na proposta está incluída uma análise de custos e o cronograma previsto (GPR2, GPR4 e GPR5). O *Backlog* identifica os riscos que são analisados na planilha de riscos (GPR6). No *Backlog* também são identificadas as habilidades e conhecimentos necessários para realizar as tarefas relacionadas com cada história e quando uma tarefa é designada, considera-se isto (GPR7). A nova planilha da proposta funciona como o plano integrado, quando são incluídas as conexões aos documentos periféricos como os que estão armazenados na ferramenta *sprint.er* e a planilha de riscos (GPR10). Em cada *sprint*, é realizada a análise do esforço necessário e são revisadas estimativas para garantir que as histórias sejam terminadas (GPR11). As assinaturas dos participantes no projeto, externos e internos, evidencia que os envolvidos assumem o compromisso com ele (GPR12).

Das tarefas de planejamento só faltam GPR8 e GPR9. Máximo e a equipe da T<sup>2</sup> terão que trabalhar para gerar as mudanças necessárias para que os processos estejam implementados.

Enquanto isso, é mais produtivo voltar-se para a tarefa de garantir que as atividades de monitoramento e controle estejam sendo implementadas.

A primeira ação já realizada na primeira semana do contrato foi incorporar ao repertório de atividades o *Scrum* diário. Seguindo os alinhamentos gerais já vistos no “*Scrum: Organizando o sistema para um estado de equilíbrio orgânico*” do Capítulo 3, Marcela passa a cumprir funções de *Scrum Master* e, junto com os gêmeos e Ana, como equipe do *sprint*, se reúnem diariamente na sala de *design* para avaliar os avanços do dia anterior e o estado geral do *sprint*, incluindo os riscos associados. Esta reunião não é exatamente um *Scrum* de acordo com as regras de [SCHWABER e BEEDLE, 2002], mas servem ao propósito de controlar o projeto. Continuando com a análise de lacunas (*gap analysis*) entre o implementado e o que é exigido pelo MPS, os componentes do modelo que o *Scrum* diário contribui para evidenciar são então GPR 13, pois nesta reunião o escopo, as tarefas, as estimativas, o orçamento e o cronograma do projeto são supervisionados com relação ao planejado; GPR14, porque os recursos materiais e humanos, assim como os dados relevantes do projeto são supervisionados com relação ao planejado para o *sprint*; GPR15, pois são supervisionadas as tarefas relacionadas com os riscos e GPR 16 porque a participação das partes interessadas no projeto é planejada, supervisionada e mantida.

Depois são agregados ao processo demonstrações de produto que são realizadas com o cliente ao final de cada *sprint* e, desse modo, a participação de todos os interessados está completamente evidenciada. Com todos estes elementos, Máximo constrói um processo que vira um diagrama de raias (um exemplo deste tipo de diagrama é mostrado na Figura 5.9).

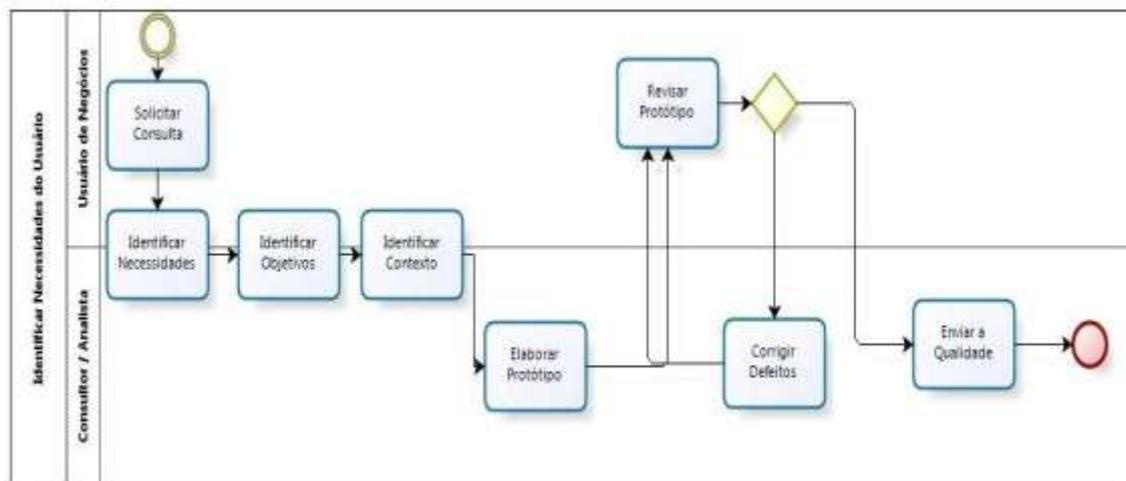


Figura 5.9: Diagrama de Raias

Ainda falta implementar vários resultados esperados do MR-MPS-SW para o Nível G, inclusive não se discutiu todos os processos correspondentes, já que a Gerência de Requisitos (GRE) é parte do que é necessário fazer para alcançar o Nível, mas Máximo tem a segurança de que o que foi feito até agora evoluiu para atender, também, esse processo. Ele analisa então os resultados esperados do GRE seguindo a tabela correspondente (Tabela 5.3) e comprova que a proposta, enquanto documento integrador do projeto, contém as histórias e, como está sendo assinada para mostrar sua aprovação pelos clientes, cobre desse modo o resultado esperado GRE 1; que a revisão que faz a equipe das histórias para estimá-las e revisar os riscos e habilidades requeridas cobre GRE 2; que o painel *kanban*, como está sendo utilizado, permite rastrear as histórias por meio dos produtos do projeto, cobrindo assim GRE 3, que nas reuniões diárias são introduzidas tarefas derivadas que afetam as estimativas das histórias e disparam novas análises que determinam que GRE 4 está coberto, e que entre estas mudanças que são geradas a partir da equipe e as mudanças que possam surgir por pedido do cliente de um *sprint* ao seguinte, provocados ou não pela demonstração do que foi até então realizado, surge a prova necessária para GRE 5. O próximo passo para Máximo é completar os resultados esperados com os dois que faltam em GPR e analisar e implementar os resultados esperados para os atributos de processo correspondentes, AP 1.1 e AP 2.1.

<b>Gerência de Requisitos (GRE)</b>	
<b>GRE1</b>	<b>O entendimento dos requisitos é obtido em conjunto com os fornecedores de requisitos;</b>
<b>GRE2</b>	<b>Os requisitos são avaliados com base em critérios objetivos e é obtido o compromisso da equipe técnicas com estes requisitos;</b>
<b>GRE3</b>	<b>A rastreabilidade bidirecional entre os requisitos e os produtos de trabalho é estabelecida e mantida;</b>
<b>GRE4</b>	<b>Revisões em planos e produtos de trabalho do projeto são realizadas com o objetivo de identificar e corrigir inconsistências relacionadas aos requisitos;</b>
<b>GRE5</b>	<b>Mudanças nos requisitos são geridas durante o projeto.</b>

**Tabela 5.3 Processo GERÊNCIA DE REQUISITOS [SOFTEX, 2012a]**

Em conformidade com os resultados até o momento, Máximo introduz um curto procedimento para definir a estrutura de pastas de um projeto no *site* de cada projeto e senta com Marcela e um dos gêmeos para escrever o *script* que vai automatizar sua execução. Alinhada às necessidades da organização, a estrutura contém uma pasta para cada um dos passos do processo que a organização segue, mais algumas pastas de comunicações e ações a serem tomadas. Dentro de cada pasta, a equipe pode encontrar as planilhas correspondentes, a serem utilizadas no processo. Desse modo, Máximo está seguro de que os dois resultados que faltam do GPR ficaram cobertos.

<b>O QUE ACONTECEU ATÉ AGORA</b>
19. Máximo introduziu o <i>Scrum</i> diário para avaliar os avanços dia a dia e o estado geral do <i>sprint</i> ; foram incluídos os riscos associados, para com isso melhorar o controle e fornecer evidências de vários resultados esperados do MR-MPS-SW.
20. A organização incorporou uma demonstração do produto ao final de cada <i>sprint</i> como critério de aceitação pelo usuário que serve para evidenciar sua participação.
21. A análise de lacunas ( <i>gap analysis</i> ) exercida como atividade contínua mostra que os resultados esperados para GRE surgem dos processos já implementados.
22. Máximo introduziu um curto procedimento para definir a estrutura de pastas de um projeto em seus próprios <i>sites</i> , que evidencia a implementação de GPR 8 e GPR 9.

## **5.7 Preparando a Avaliação**

Para alcançar o nível G do MR-MPS-SW ficaram ainda por definir as atividades que planejem e disponibilizem aos interessados os recursos e o ambiente de trabalho necessários para executar o projeto, bem como seus dados relevantes. Para estes últimos é preciso definir como são reunidos, armazenados e distribuídos. Também é preciso definir o mecanismo para acessá-los incluindo, quando for pertinente, restrições de acesso. Além disso, é necessário revisar os atributos dos processos GPR e GRE, esses que falam da capacidade da organização para levá-los adiante. Estes são abreviados como AP e tem, por sua vez, resultados esperados que são abreviados como RAP.

Em primeiro lugar, como AP 1.1. é simplesmente “O processo é executado”, com um único resultado esperado, RAP 1, “O processo consegue seus resultados definidos”, o que Máximo leva em conta é que os resultados definidos, tirando os já identificados, estão implementados, depois não há ações relacionadas.

A coisa muda quando se trata do segundo atributo, AP 2.1 “O processo é gerenciado”, porque os resultados esperados são nove:

RAP2. Existe uma política organizacional estabelecida e mantida para o processo;

RAP3. A execução do processo é planejada;

RAP4. A execução do processo é supervisionada e são realizados os ajustes necessários;

- RAP5. As informações e os recursos necessários para a execução do processo são identificados e colocados à disposição dos interessados;
- RAP6. As responsabilidades e a autoridade para executar o processo são definidos, atribuídos e comunicados;
- RAP7. As pessoas que executam o processo são competentes em termos de formação, treinamento e experiência relacionados;
- RAP8. A comunicação entre as partes interessadas no processo é planejada e executada de modo que se assegure sua participação;
- RAP9. Os resultados do processo são revisados com a alta gerência para proporcionar visibilidade sobre sua situação na organização;
- RAP10. O processo planejado para o projeto é executado.

Máximo propõe uma política sobre as atividades de um projeto de Nível G que diz o seguinte: “Os projetos da Tahini-Tahini se originam de requisitos do cliente e são aprovados por este, são analisados para assegurar sua viabilidade, são estimados e orçados. Os custos baseiam-se em estimativas de tamanho e esforço, e as etapas nas quais serão realizados são planejadas e coordenadas com os clientes. O pessoal escolhido para realizar as tarefas é competente, baseado em sua capacidade objetiva. Os planos que são aprovados são utilizados como base para controlar e monitorar o projeto. Toda mudança que impacte nos compromissos internos ou externos deve contar com as mesmas aprovações que o projeto original. Qualquer violação a esta regra será considerada motivo de sanções e, eventualmente, suspensão dos que a violem reiteradamente”. Em discussões com os gêmeos, que se opunham a uma menção a sanções, Máximo conseguiu convencê-los de que uma regra sem sanção é mais um desejo do que uma regra, e que, além disso, a sanção era facultativa, ou seja, se a razão por trás da violação fosse embasada e justificada, não havia por que chegar à sanção. Este enunciado aparece hoje encabeçando a página de processos da T<sup>2</sup>. Com isto, temos evidências da implementação do RAP 2.

Depois, Máximo atacou os seguintes RAPs, um a um: planejar, supervisionar a execução e ajustá-la, se for necessário; colocar à disposição daquelas pessoas e papéis responsáveis os recursos, dados e informações necessários para levar o trabalho adiante e atribuir autoridade para conduzi-lo, proporcionando-lhes capacitação quando não tiverem a competência adequada, ou nomear quem tiver apto a executar o papel; garantir a participação das partes interessadas; revisar os processos com a alta gerência para proporcionar visibilidade sobre sua situação na organização e, por último, executar o que foi definido..

Incorporou ainda um processo de iniciação de projetos. que descreve como deve ser completada uma proposta, obviamente baseando-se no processo que descreveu para levar adiante um projeto usando o painel virtual, mas estendendo-o para que seja coberta todas as etapas de planejamento, controle e gestão de requisitos. No mais alto nível, descreve a sequência do processo de maneira gráfica. Para cada atividade usa a planilha de procedimento (Figura 5.10) e um diagrama de raias que desenha usando um programa gráfico.

Por enquanto, deixa de lado as medições de cada atividade, mas inclui o diagrama. O processo inclui todas as atividades para gerar o plano, definindo os papéis e funções de cada ator e interessado. Quando a T<sup>2</sup> começar seu próximo projeto, o processo se executará e gerará a proposta com todas as planilhas correspondentes, assim como as atividades de Gerência de Requisitos necessárias e as atividades de controle do projeto. O processo assim definido cobre o planejamento do planejamento, pois identifica quem são os responsáveis por gerar a proposta, como se estabelece a responsabilidade e quais recursos lhe dão suporte. Deste modo, a T<sup>2</sup> atende aos resultados esperados dos Atributos de Processo.

PLANILHA DE DETALHES DO PROCEDIMENTO
<b>Nome da Atividade:</b>
<b>Propósito (O que esperamos conseguir com esta atividade?)</b>
<b>Envolvidos (pelo menos o responsável)</b>
<b>Entradas exigidas</b>
<b>Produtos de trabalho criados (saídas)</b>
<b>Critério de Entrada (Como sabemos quando começar esta atividade?)</b>
<b>Critério de Saída (Como sabemos que a atividade foi concluída satisfatoriamente?)</b>
<b>Subatividades (3 a 6 subatividades [se existirem] para realizar esta atividade, que serão incorporadas no diagrama de raias)</b>
<b>Medições (Como podemos medir o desempenho desta atividade?)</b>
<b>Sequenciamento (Quais atividades serão realizadas antes e depois desta?)</b>

Figura 5.10: Planilha de Detalhamento de um Procedimento

Um dos principais impulsores da melhoria contínua em uma organização é a realização de reuniões periódicas que possibilitem aos envolvidos analisar o que foi feito e sugerir mudanças aos processos para melhorá-los. Essas reuniões recebem o nome de retrospectivas, e a T<sup>2</sup> as adota sob a preparação e supervisão<sup>59</sup> de Máximo.

O mais importante das retrospectivas é que elas aconteçam. Sem elas, as equipes repetem seus erros ao invés de aprender com eles. A responsabilidade do *Scrum Master*, portanto, é que estas reuniões sejam planejadas entre *sprints* e realizadas. É possível que uma situação crítica no meio de um *Sprint* motive convocar uma retrospectiva “de emergência”, mas seu lugar natural é entre *Sprints*. Deve-se reservar entre uma e três horas da equipe com este propósito, sendo que a duração dependerá da participação e da quantidade de temas a serem discutidos na reunião. É importante que participem todos os interessados, incluindo o Dono do Produto. O lugar da reunião não pode ser a própria sala de trabalho, porque é sumamente importante não sofrer interrupções. Máximo escolheu a sala de *design* porque tem as lousas que permitem trabalhar de forma cômoda e não tem telefones nem telas de computadores. Não são permitidos *laptops* na reunião de análise retrospectiva. Divide-se a lousa em três colunas: “O que Andou Bem”, “O que Precisa Melhorar” e “Sugestões”. Primeiro são completadas as duas primeiras colunas e quando já não existirem mais propostas para nenhuma delas, são iniciadas as rodadas de sugestões.

As técnicas para isto são múltiplas e é valioso conhecer tantas quanto for possível. Já mencionamos o diagrama de *Ishikawa*, ou de espinha de peixe, assim como os cinco “porquês”, e as oito disciplinas. Existem mecanismos de pensamento, como já mencionamos o dos chapéus de cores de [DE BONO, 1985], que permitem exercitar o pensamento crítico. Estas e muitas outras técnicas têm mérito e sua aplicação dá excelentes resultados.

O informe de Máximo para seu chefe inclui a seguinte tabela (Figura 5.13), que mostra as evidências que temos da obtenção dos resultados esperados pela implementação do processo GPR do modelo. As colunas definem a documentação existente, da esquerda para a direita: Painel *kanban*, Planilha de Histórias, Proposta, *Script* de Definição de Pastas e Recursos, Demonstração Final de Cada *Sprint*, Planilha de Riscos, Informe da Retrospectiva e o *Scrum* Diário. Cada uma dessas evidências contribui para que a totalidade dos resultados esperados sejam cobertos na implementação escolhida. Da mesma forma, há outras tabelas de cobertura para GRE e RAP.

<sup>59</sup> Escolhemos “preparação e supervisão” para englobar o significado de “*coaching*”, que lamentavelmente não tem uma boa tradução em um único vocábulo em português.

Gerencia de Projetos no Nível G		Kanban	Backlog	Proposta	Scripts	Demos	Planilhas	Repositório	Privacidade	Scrum
GPR 1	O escopo do trabalho para o projeto é definido;		*							
GPR 2	As tarefas e os produtos de trabalho do projeto são dimensionados utilizando métodos apropriados	*	*							
GPR 3	O modelo e as fases do ciclo de vida do projeto são definidos	*								
GPR 4	O esforço e o custo para a execução das tarefas e dos produtos de trabalho são estimados com base em dados históricos ou referências técnicas		*	*						
GPR 5	O orçamento e o cronograma do projeto, incluindo a definição de marcos e pontos de controle, são estabelecidos e mantidos	*		*						
GPR 6	Os riscos do projeto são identificados e o seu impacto, probabilidade de ocorrência e prioridade de tratamento são determinados e documentados						*			*
GPR 7	Os recursos humanos para o projeto são planejados considerando o perfil e o conhecimento necessários para executá-lo	*	*							
GPR 8	Os recursos e o ambiente de trabalho necessários para executar o projeto são planejados			*	*					
GPR 9	Os dados relevantes do projeto são identificados e planejados quanto à forma de coleta, armazenamento e distribuição. Um mecanismo é estabelecido para acessá-los, incluindo, se pertinente, questões de privacidade e segurança				*					
GPR 10	Um plano geral para a execução do projeto é estabelecido com a integração de planos específicos			*						
GPR 11	A viabilidade de atingir as metas do projeto é explicitamente avaliada considerando restrições e recursos disponíveis. Se necessário, ajustes são realizados	*	*							
GPR 12	O Plano do Projeto é revisado com todos os interessados e o compromisso com ele é obtido e mantido			*						
GPR 13	O escopo, as tarefas, as estimativas, o orçamento e o cronograma do projeto são monitorados em relação ao planejado	*	*							*
GPR 14	Os recursos materiais e humanos bem como os dados relevantes do projeto são monitorados em relação ao planejado	*								*
GPR 15	Os riscos são monitorados em relação ao planejado;					*				*
GPR 16	O envolvimento das partes interessadas no projeto é planejado, monitorado e mantido					*				*
GPR 17	Revisões são realizadas em marcos do projeto e conforme estabelecido no planejamento					*		*		*
GPR 18	Registros de problemas identificados e o resultado da análise de questões pertinentes, incluindo dependências críticas, são estabelecidos e tratados com as partes interessadas					*		*	*	*
GPR 19	Ações para corrigir desvios em relação ao planejado e para prevenir a repetição dos problemas identificados são estabelecidas, implementadas e acompanhadas até a sua conclusão							*		*

Figura 5.11: Evidências para GPR no Nível G

O QUE ACONTECEU ATÉ AGORA

- 23. Máximo definiu e codirigiu várias implementações do procedimento de análise retrospectiva, que fornece evidência de GPR 17, GPR 18 e GPR 19.
- 24. Máximo redigiu uma política que a organização aprovou para estabelecer os alinhamentos do que é esperado do pessoal da T<sup>2</sup>.
- 25. Máximo desenvolveu um processo que permite evidenciar as capacidades de processo de Nível G do MR-MPS-SW para as atividades de planejamento, por meio da proposta e gestão de requisitos, do *Backlog* do produto e da evolução do painel *kanban*.

Deixamos Máximo e a T<sup>2</sup> organizando a avaliação formal do Nível G do modelo MPS.

## CAPÍTULO 6 - UMA ORGANIZAÇÃO EM CRESCIMENTO

### 6.1 Nível F do MPS-SW

Este capítulo tem como objetivo discutir a implementação de práticas que são exigidas no nível F do modelo de referência MR-MPS-SW. No nível F, Gerenciado, os processos são Aquisição (AQU), Gerência de Configuração (GCO), Garantia da Qualidade (GQA), Gerência do Portfólio de Projetos (GPP) e Medição (MED). Notemos o que diz o modelo:

*“A implementação dos processos para o nível F pode ser feita em qualquer sequência, visto que os processos desse nível são de apoio à gestão do projeto, fornecendo maior qualidade e controle aos produtos de trabalho. Uma vez que necessitam de processos de apoio, as organizações podem ter a necessidade de incorporar à sua equipe alguns novos perfis para realizar atividades de garantia da qualidade, gerência de configuração, medição, gerência do portfólio de projetos e aquisição de produtos. Note, no entanto, que a existência de um novo perfil não obriga necessariamente a contratação de novas pessoas, mas a definição de novas competências necessárias e delimitação de novas responsabilidades”<sup>60</sup>.*

#### O QUE ACONTECEU ATÉ AGORA

26. T<sup>2</sup> alcançou o nível G do MPS-SW em uma avaliação oficial

### 6.2 Crescem os pedidos

Tahini-Tahini passou a fazer parte do considerável número de empresas que foram avaliadas no Nível G do MR-MPS-SW. Mesmo com alguns detalhes aqui e ali identificados pela equipe de avaliação MPS, estes foram resolvidos prontamente pelo pessoal da T<sup>2</sup> e a organização recebeu sua placa de aço em um evento do Simpósio Brasileiro de Qualidade de *Software* (SBQS) realizado anualmente. A notícia circulou rapidamente nos meios locais e a T<sup>2</sup> teve seus quinze minutos de fama. Por mais breve que tenha sido essa notoriedade, os amigos da T<sup>2</sup>, como o Doutor Molar, se sentiram obrigados a compartilhar o orgulho que a notícia trouxe. A consequência imediata foi que outros clientes se aproximaram da T<sup>2</sup> para conhecer seus serviços.

Uma oferta muito tentadora chegou do Conselho Profissional de Contadores Públicos (CoProCoP) para que os tradicionais serviços de folha de pagamento de salários fossem ofertados aos membros do CoProCoP e, ao mesmo tempo, que a T<sup>2</sup> trabalhasse para e com eles (os membros do Conselho), estendendo esses serviços a eles e a outros clientes potenciais com o conhecimento coletivo do conselho em certas aplicações contábeis e de administração. A oferta é sumamente interessante, porque possibilitaria o aumento imediato do faturamento com clientes existentes ao oferecer as novas funcionalidades e também ampliaria o mercado potencial.

Depois de algumas breves conversas iniciais com o CoProCoP, uma reunião plenária da T<sup>2</sup> foi feita para que decisões sobre o caminho a ser seguido fossem tomadas. Esta reunião é crucial porque a oferta coloca uma encruzilhada real: crescer ou não crescer. Sem dúvida, crescer é algo que toda organização acredita aspirar. Mais clientes, mais faturamento, melhores salários, férias pagas, um futuro afortunado. Mas também implica em deixar de lado a cotidianidade dos amigos, profissionalizar a administração, descolar-se das tarefas divertidas ou ser obrigado a fazê-las de outro modo. As emoções entre os participantes da reunião oscilam entre os que estão assustados e os que estão eufóricos. As vias de crescimento são pressentidas como estranhas e complicadas. Depois de duas horas de reunião, não se chega a nenhuma conclusão. Então, um dos gêmeos propõe que Máximo seja chamado.

### 6.3 Procurando Ajuda Fora da Tahini-Tahini

Dois dias mais tarde, Máximo facilita a reunião plenária da T<sup>2</sup> na sala de *design*. Desta vez, trouxe consigo seis chapéus de diferentes cores: [DE BONO, 1985]. As cores são: azul, branco, vermelho, preto, amarelo e verde. Cada cor representa uma modalidade de pensamento da qual nosso cérebro é capaz. Uma pessoa coloca o chapéu azul quando quer expressar pensamentos sobre o método que segue, quer dizer, pensamentos sobre o método de trabalho. O chapéu branco é usado para trabalhar com os dados concretos, com a

<sup>60</sup> SOFTEX, 2012c, página 7

informação disponível. Com o chapéu vermelho são expressos os sentimentos. Não é preciso justificá-los de forma alguma. A lógica da cautela é expressa com o chapéu preto, que tende a apontar os problemas, daí a cor. O amarelo expressa a lógica do otimismo, do que pode acontecer e do que gostaríamos que acontecesse. O verde é para provocar. Não há uma ordem pré-determinada de seu uso, em geral começamos a usá-los para estimular a discussão e evitar que o pensamento fique estancado na lógica do que pode andar mal. Pode-se especificar uma ordem inicial e todos usam essa cor por uma rodada, por exemplo, o chapéu azul. Também é possível que uma pessoa pegue o chapéu que corresponde ao que quer expressar antes de falar, de modo que todos saibam que essa comunicação é feita a partir do ponto associado à cor.

Máximo coloca o chapéu azul e explica os papéis das cores e propõe usá-los em rodadas. Primeiro, todos usarão o amarelo, depois o vermelho e depois o preto. Quando chega o momento de expressar a negatividade, todos já analisaram as possibilidades e seus sentimentos a respeito das mudanças, e os compartilharam. Máximo foi registrando-os no quadro eletrônico e muda de quadro quando chegam ao preto e, ao invés de anotar os pensamentos em uma lista, faz duas colunas (Tabela 6.1):

Não estamos prontos para crescer	
Crescimento muito rápido	
Muitas linhas de produto	
Perda de controle	

**Tabela 6.1: Pensamentos Negativos sobre a Mudança**

Máximo copiou os pensamentos negativos de todos. Alguns foram repetidos ou estavam suficientemente relacionados e puderam ser expressos em uma única frase. Assim, todos os pensamentos foram sintetizados nas frases que escreveu. Como todos já conhecem o diagrama de *Ishikawa* de espinha de peixe, ele o utiliza para começar a sessão de análise do enunciado e chegar a uma estratégia de crescimento. Desenha um esqueleto de sardinha e escreve a primeira frase na cabeça do peixe (Figura 6.1):

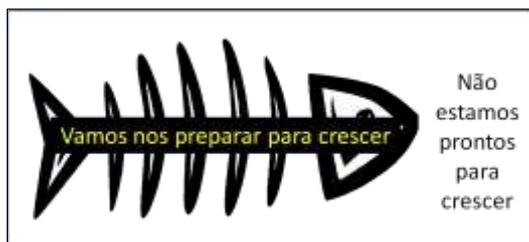


**Figura 6.1: Diagrama *Ishikawa* sobre Crescimento 1**

As risadas dão lugar a caras preocupadas. Há perguntas e discussões entre os sócios. Máximo deixa que continuem por uns minutos e depois interrompe:

“Para que serve o diagrama de *Ishikawa*?”

A pergunta é retórica, mas tem seu resultado. Imediatamente, todos percebem que voltaram a colocar o chapéu preto, alguns também o vermelho. Querendo que voltem ao exercício de busca de soluções, ele mesmo escreve a solução (Figura 6.2).



**Figura 6.2: Diagrama *Ishikawa* sobre Crescimento 2**

Há um silêncio de concordância. Máximo deixa que a ideia vá crescendo nos sócios e encabeça sua lista com este título: “Preparando-nos para Crescer” e apaga a primeira linha. A

reunião vai se animando aos poucos e onde se viam problemas agora há soluções. Algumas são descartadas, por exemplo, o rápido crescimento pode ser controlado mediante a contratação de pessoal, mas o custo em esforço do pessoal atual e o investimento em facilidades são grandes. Por outro lado, é possível considerar a contratação de serviços de desenvolvimento de um fornecedor estabelecido. A tabela completa ficou assim constituída (Tabela 6.2):

PREPARANDO-NOS PARA CRESCER	
Crescimento muito rápido	Implica na contratação de serviços externos
Muitas linhas de produto	implica contar com gestão de configuração
	implica selecionar entre opções de investimento
Perda de controle	implica controle objetivo mediante decisões baseadas em dados
	implica controlar objetivamente a aplicação de normas e políticas

**Tabela 6.2: Preparando-nos para Crescer**

Agora Máximo sugere que eles ataquem os riscos, um por um, começando pela contratação de uma empresa fornecedora. Máximo produz uma lista de riscos e a projeta na parede:

Contratação de Fornecedores, Riscos:

1. Serem contratados os serviços equivocados;
2. Serem contratados os fornecedores equivocados;
3. Serem escolhidos a partir de um subconjunto inadequado de fornecedores;
4. Serem escolhidos subjetivamente;
5. Ser feito um contrato que não atenda aos interesses das duas partes;
6. O contrato ser encerrado sem que todos os requisitos tenham sido cumpridos.

Os sócios revisam a lista com Máximo, ponto por ponto. Vão concordando, com algumas perguntas. O item 5 é discutido um pouco mais que os outros. Claudio quer saber por que é importante fazer um contrato que sirva às duas partes, além das considerações éticas. Máximo explica que se o contrato não serve ao cliente e ao fornecedor é possível que o fornecedor não possa entregar seu produto, arrastando, como consequência, também o cliente em sua queda.

Máximo pergunta se alguém quer agregar mais algum risco e espera a resposta. Depois de um silêncio suficientemente longo, continua com as mitigações:

*“Bem, agora podemos ver o que precisamos fazer para que estes riscos não se materializem ou, se ocorrerem, que tenham baixo impacto”.*

Máximo não se preocupa em definir os termos de maneira formal, pois no momento apropriado fará isso. Por enquanto, só precisa que colaborem com ele. Usando os diagramas de espinha de peixe de maneira muito rudimentar, já que as soluções são óbvias, chega-se à seguinte conclusão para cada um dos riscos:

Para não contratar os serviços equivocados, é preciso planejar a aquisição, determinando o que será adquirido e por que isso será produzido.

Para não contratar os fornecedores equivocados, é preciso estabelecer claramente, antes de buscá-los, quais são os atributos que deverão possuir para participar da análise de preços.

Para não escolher um subconjunto inadequado de fornecedores, é preciso pesquisar o mercado, com base no tipo de aquisição e nos atributos exigidos dos fornecedores.

Para não escolher subjetivamente entre os candidatos, é preciso construir um modelo de decisão que permita realizar a escolha baseando-se ao máximo em dados objetivos.

Para não fazer um contrato que não sirva às duas partes, é preciso negociar com boa disposição, procurando o “ganha-ganha” com coragem, maturidade, imaginação e integridade. O contrato deve permitir dirigir processos conjuntos.

Para que o contrato não seja encerrado sem que todos os requisitos tenham sido cumpridos, é necessário que haja um acordo claro sobre o que deve ser entregue, quando e em quais condições.

“Esta é uma boa lista”, diz Máximo. “O mais interessante é que estas ações podem ser implementadas cumprindo com a implementação de todos os resultados do Processo de Aquisição do MR-MPS-SW” e sorri ao falar isso. Projeta então esses resultados:

<b>Aquisição (AQU)</b>	
<b>AQU1</b>	<b>As necessidades de aquisição, as metas, os critérios de aceitação do produto, os tipos e a estratégia de aquisição são definidos;</b>
<b>AQU2</b>	<b>Os critérios de seleção do fornecedor são estabelecidos e usados para avaliar os potenciais fornecedores;</b>
<b>AQU3</b>	<b>O fornecedor é selecionado com base na avaliação das propostas e dos critérios estabelecidos;</b>
<b>AQU4</b>	<b>Um acordo que expresse claramente as expectativas, responsabilidades e obrigações de ambas as partes (cliente e fornecedor) é estabelecido e negociado entre elas;</b>
<b>AQU5</b>	<b>Um produto que satisfaça a necessidade expressa pelo cliente é adquirido baseado na análise dos potenciais candidatos;</b>
<b>AQU6</b>	<b>A aquisição é monitorada de forma que as condições especificadas sejam atendidas, tais como: custo, cronograma e qualidade, gerando ações corretivas quando necessário;</b>
<b>AQU7</b>	<b>O produto é entregue e avaliado com relação ao acordado e os resultados são documentados;</b>
<b>AQU8</b>	<b>O produto adquirido é incorporado ao projeto, caso pertinente.</b>

**Tabela 6.3: Processo AQUISIÇÃO [SOFTEX, 2012a]**

Os sócios se dedicam à criação de guias, critérios e processos para realizar aquisições. Rapidamente alguns critérios surgem claramente: a aquisição deve ficar claramente delimitada na arquitetura, com interfaces bem definidas; os fornecedores devem ser suficientemente experientes para poder entregar um produto de qualidade, mas não tão grandes ao ponto de o acordo de negócios ser totalmente unilateral; a visibilidade dentro dos processos do fornecedor e a colaboração que estejam dispostos a prestar para isso serão objeto de análise e será dada importância àqueles que demonstrem essa vontade. Esses e outros aspectos vão dando forma ao processo de aquisição. As primeiras dúvidas e os medos se dissipam na atividade criativa.

Fica a dúvida, no entanto, sobre a decisão de “fabricar ou comprar” já que é preciso considerar fatores como as características do produto que os fornecedores oferecem e como cada um se encaixa no projeto, a disponibilidade de recursos e perfis de projetos, porque é preciso equilibrar o que será feito internamente com os recursos que não serão necessários por serem do fornecedor, mais a administração do contrato, o custo de adquirir versus o custo do desenvolvimento interno; as datas críticas de entrega e integração; a possibilidade de implementar uma estratégia de alianças, incluindo os requisitos de negócio de alto nível. Também se propõem a investigar os produtos em desenvolvimento e os que já estão no mercado, procurando entender a funcionalidade e a qualidade desses produtos, o impacto sobre a concorrência, licenças, permissões, responsabilidades e limitações associadas aos produtos que serão comprados, assim como a disponibilidade do produto, as questões relacionadas com a propriedade e se a decisão aumenta ou diminui os riscos. Máximo propõe criar uma matriz de Pugh<sup>61</sup> que contenha as alternativas e usá-la quando for necessário. Com essa tarefa pendente termina a reunião.

<sup>61</sup> Ver Capítulo 2.

**O QUE ACONTECEU ATÉ AGORA**

27. Máximo utilizou a técnica dos chapéus de cores para semear o caminho de ideias e, com o diagrama de Ishikawa, ajudou os sócios a elucidar qual caminho seguir na empresa (crescer ou não crescer).
28. Máximo ajudou a revisar os riscos de contratação de fornecedores e estudar as alternativas de comprar, reutilizar ou fabricar, introduzindo a técnica da matriz de Pugh.

**6.4 O que deveríamos fabricar?**

No dia seguinte, Máximo recebe uma ligação de Claudio. Aconteceu algo inesperado: à proposta de trabalho conjunto do CoProCoP se somou outra de uma empresa que mantém e vende um sistema de gerenciamento de estoque para farmacêuticos, que poderia se aproveitar bem de um sistema de folha pagamento de salários. A pergunta é, quando Máximo pode ajudá-los a resolver essa questão, facilitando a reunião dos sócios. Quando Máximo chega ao escritório da T<sup>2</sup> já há mais duas propostas de trabalho conjunto: aos contadores e fabricantes de *software* para farmácias, se somaram uma ligação da associação de serviços de saúde, que está interessada em desenvolver um sistema novo, na nuvem, pelo qual a experiência de T<sup>2</sup> é bastante importante, e um fornecedor de serviços a agentes de seguros que também quer “se mudar” para a nuvem de modo a integrar mais facilmente novas tecnologias (*laptop*, *tablets*, *netbooks* e também *smart phones*) à oferta. As quatro propostas estão sendo avaliadas como fornecedores usando a matriz de Pugh que Máximo havia deixado como tarefa.

“Um momento”, pede Máximo. “Não se trata de fornecedores alternativos do mesmo produto. Não concorrem por um único espaço. Por que não é possível pensar em dois ou três projetos paralelos?”

O silêncio é quase audível. A seleção de projetos que uma organização encara deve ser o resultado de um profundo conhecimento do mercado e da visão do rumo que levará o crescimento da organização. A governança dos recursos da organização, em um nível superior ao dos projetos, é essencial para o máximo aproveitamento dos patrimônios coletivos. Qualquer outra forma de definir a designação de recursos aos projetos, pode dar lugar a vinculações inadequadas. A falta de visão é um primeiro obstáculo. Máximo se dirige à lousa e escreve: “Visão para T<sup>2</sup>: nos próximos dois anos nos comprometemos a”, e deixa o cenário em aberto.

Claudio fala primeiro: “Gostaria que fôssemos uma força decisiva no mercado de SaaS”. Máximo passa a caneta para que escreva, textualmente, sua frase no quadro. Máximo se levanta, pega a caneta e corrige a frase. Agora pode-se ler: “Nos próximos dois anos nos comprometemos a ser uma força decisiva no mercado de SaaS para a América Latina.” Agora se levanta Marcela, pega a caneta das mãos de Máximo e risca “força decisiva” e em cima dela escreve “uma referência entre as cinco melhores empresas”. Um dos gêmeos se levanta e com o dedo apaga a palavra “uma” e escreve “a” em seu lugar. A visão diz então: “Nos próximos dois anos nos comprometemos a ser a referência entre as cinco melhores empresas no mercado de SaaS para a América Latina”. O silêncio agora é solene e Máximo o interrompe:

“Se for assim, vender sistemas contábeis com folha de pagamento de salários não é suficiente. Precisamos ter uma carteira de produtos e gerenciar os projetos como um portfólio de investimentos”.

Pode-se entender um portfólio, ou “carteira”, segundo o [PMI, 2008], como “(...) um conjunto de projetos, programas e outros trabalhos que se agrupam para facilitar a gestão efetiva do conjunto do trabalho para cumprir com os objetivos estratégicos específicos. Os componentes da carteira não necessariamente precisam ter alguma relação de dependência ou estar diretamente relacionados”. Além do mais, pode-se entender que “gestão da carteira” refere-se à “gestão centralizada de uma ou mais carteiras, o que inclui a identificação, priorização, autorização, administração e controle de projetos, programas e outros trabalhos relacionados, para alcançar os objetivos estratégicos específicos” [PMI, 2008].

“Então, quais das propostas estão alinhadas com a visão?” pergunta Máximo.

“Todas, menos a dos agentes de seguros”, responde Marcela com segurança.

Máximo risca da matriz de Pugh essa oferta e apaga todos os atributos desejáveis da solução da matriz, deixando em branco a coluna da esquerda.

“NÃO!!!”, gritam todos. O trabalho não tinha sido, ainda, copiado em nenhum lugar. Por sorte, Mariano tem o costume de tirar, periodicamente, fotografias do que é escrito nos quadros e foi possível recriar a informação apagada sem muito esforço. Máximo, então, escreve os atributos da nova matriz, a de seleção de aliados. Os atributos que coloca são: esforço (exigido pelo projeto), domínio (da aplicação), sinergia (com os outros projetos), investimento (montante), alinhamento (com a visão estratégica) e mercado pot. (mercado potencial de venda do produto resultante). Preenchem a matriz e o resultado parcial é o seguinte:

atributo	contadores	software para drogarias	Associação serviços de saúde	serviços a agentes de seguros
esforço	1 sistema completo	1 sistema migração a SaaS	1 sistema integrado para hospitais	
domínio	contabilidade	manejo de stocks	ERP completo	
sinergia	boa	mediana	total	
investimento	media	media	enorme	
alinhamento	bom	mediano	enorme	
mercado pot.	muito amplo	drogarias	hospitais, drogarias, médicos	????????

**Tabela 6.4: Matriz de Pugh sobre Propostas**

Os pontos de interrogação são escritos pelos gêmeos Galarraga, que, de repente, se levantaram ao mesmo tempo e disputaram a caneta.

“O que aconteceria se”, diz Alberto,

“...a tecnologia que querem usar os agentes de seguros seja usada...” continuou Armando,

“...e aos pacientes, as salas de primeiros socorros, os históricos clínicos” terminou Armando.

Os olhos de todos brilhavam de animação. Máximo foi o encarregado de jogar um balde de água fria no conjunto:

“Esperem, esperem. Nada de soluções ainda. Vamos tomar um café e pensarmos juntos na estratégia.”

Máximo usou aqui uma tática de facilitação que consiste em romper a reunião em pequenos grupos para que as ideias sejam cruzadas. Esta tática é uma das variantes do que se chama “polinização cruzada” que adota várias formas, mas que em sua essência consiste em ativar a participação de todos para cruzar ideias. Algumas outras formas são mais divertidas, como “O Coquetel”, onde são entregues ordens e os assistentes circulam trocando ideias entre eles aos pares como em um evento social. Os resultados alcançados são apresentados a todos. Máximo tirou todo o formalismo da situação para permitir uma troca menos rígida, mas espera que a falta de ordens não derive em outros temas por causa do interesse manifestado de todos os presentes.

De volta ao redor da mesa de *design*, cada um com seu cafezinho, Máximo pergunta:

“O que vamos fazer?”

Marcela e Claudio se entreolham. Ambos estiveram conversando nos cinco minutos do intervalo. Claudio é o mais velho de todos no grupo e pede a palavra formalmente:

“Marcela e eu pensamos que é preciso crescer, e é preciso crescer rápido. A dor do crescimento vai se compensar com a magnitude da recompensa, e se perdemos agora não perdemos muito, podemos voltar a começar com o que aprendemos nesta aventura. Para crescer assim rapidamente é preciso correr riscos. Em particular, é preciso conseguir capital”.

Outra vez o silêncio é incômodo. Claudio olha um por um para enfatizar as ideias e continua:

“Para conseguir capital é preciso ter um plano de negócios. Devemos considerar a expansão do nosso próprio capital humano, por exemplo, contratando mais profissionais que trabalhem sob nossas ordens. Isto vai nos levar a uma divisão técnica do trabalho e à especialização.

*Podemos ir mudando as posições se alguém não se sentir confortável com a posição para a qual foi designado, mas não vemos outra solução. Se pararmos, desapareceremos”.*

Alfredo e Ana se olham. *“Estamos de acordo, diz Alfredo. Levante a mão quem está de acordo”.*

Todas as mãos se levantam. Máximo volta a intervir: *“Façamos o plano”.* E logo em seguida, monta uma nova planilha, que ao final de uns minutos fica assim completada (Tabela 6.5)

<b>risco:</b>	<b>mitigação:</b>
escolhermos as linhas de produto sem uma visão estratégica como base	seguirmos um processo ordenado para fixar a visão, a estratégia e as linhas de negócio
não há orçamento suficiente para todas as linhas de produto	as linhas de produto escolhidas serão financiadas proporcionalmente à sua importância estratégica
ninguém ficar a cargo de cumprir a visão estratégica	vincularmos, claramente, a responsabilidade por cada linha de produto com ênfase nos resultados de negócio
com o tempo a visão estratégica ser perdida	controlarmos que a linha escolhida seja a linha seguida
	quando há desvios da estratégia, implementarmos ações corretivas
a escassez de recursos implicar desequilíbrios entre linhas	utilizarmos a visão estratégica para resolver conflitos de recursos e dependências críticas
certos projetos perderem valor para a estratégia	revisar, periodicamente, a visão estratégica e redefinir as linhas de produto, cancelando ou continuando projetos
a organização em seu conjunto não visualizar as decisões estratégicas	comunicar a visão e o estado dos projetos

**Tabela 6.5: Riscos do Crescimento**

*“Bom, diz Máximo, já estamos preparados para implementar o processo gerência de portfólio”.*

Projeta então a seguinte tabela (Tabela 6.6)

<b>Gerência de Portfólio de Projetos (GPP)</b>	
<b>GPP1</b>	<b>As oportunidades de negócio, as necessidades e os investimentos são identificados, qualificados, priorizados e selecionados em relação aos objetivos estratégicos da organização por meio de critérios objetivos;</b>
<b>GPP2</b>	<b>Os recursos e orçamentos para cada projeto são identificados e alocados;</b>
<b>GPP3</b>	<b>A responsabilidade e autoridade pelo gerenciamento dos projetos são estabelecidas;</b>
<b>GPP4</b>	<b>O portfólio é monitorado em relação aos critérios que foram utilizados para a priorização;</b>
<b>GPP5</b>	<b>Ações para corrigir desvios no portfólio e para prevenir a repetição dos problemas identificados são estabelecidas, implementadas e acompanhadas até a sua conclusão;</b>
<b>GPP6</b>	<b>Os conflitos sobre recursos entre projetos são tratados e resolvidos, de acordo com os critérios utilizados para a priorização;</b>
<b>GPP7</b>	<b>Projetos que atendem aos acordos e requisitos que levaram à sua aprovação são mantidos, e os que não atendem são redirecionados ou cancelados;</b>

**GPP8 A situação do portfólio de projetos é comunicada para as partes interessadas, com periodicidade definida ou quando o portfólio for alterado.**

Tabela 6.6: Processo GERÊNCIA DE PORTFÓLIO DE PROJETOS [SOFTEX, 2012a]

#### O QUE ACONTECEU ATÉ AGORA

29. Para poder desenvolver mais opiniões em paralelo, Máximo introduziu a técnica de “polinização cruzada” com uma oportuna interrupção na reunião.
30. Utilizando a análise de risco como ferramenta de exploração, Máximo introduz as práticas que implementam os resultados esperados do MR-MPS-SW correspondentes à gerência de portfólio de projetos (ou carteira de projetos).

### 6.5 Medindo resultados

A primeira consequência da decisão de crescer é um novo organograma (Figura 6.3). Alfredo e Ana irão dividir a direção da empresa perante o mercado e coordenarão as atividades dos demais integrantes da empresa.



Figura 6.3: Organograma da Tahini-Tahini

Ana será a Arquiteta-Chefe, por sua posição como docente de Arquitetura de *Software*, enquanto que Marcela será o apoio administrativo e contábil, com inclusão da gestão de pessoal e, além disso, será gerente de qualidade. Claudio ocupará o cargo de Gerente de Finanças, os gêmeos dirigirão as equipes de desenvolvimento (um deles) e suporte e manutenção (o outro). Os sócios desistiram de nomear quem ocupará cada cargo porque não conseguem distinguir entre os dois. Mariano, finalmente, se ocupará da atenção ao cliente. Esta função cuida do *marketing*, vendas, desenvolvimento de novos produtos e da verificação e validação das versões antes de seu lançamento ao público. Uma reorganização completa. Para facilitar a definição, Máximo compartilha com o grupo sua planilha de Missão e Funções (Tabela 6.7).

Estabelecidas as funções e os canais orgânicos de comunicação, é necessário contar com um sistema de decisão. As decisões devem ser as mais objetivas possíveis. A objetividade surge da existência de dados que apoiem a decisão. Sem o apoio de dados, a informação não existe, é simplesmente uma opinião e não é realmente informação. Um sistema de decisão deve se basear em um sistema de informação, e um sistema de informação deve estar sustentado por dados coletados da execução dos processos. Há dois locais onde já vimos a necessidade de coletar dados: na planilha de processos, para poder medir como estão se comportando os processos quando são executados, e nos indicadores de gestão, que ocupam uma coluna na planilha da Tabela 6.7.

Falamos de dados, de informação e de decisão. Vamos explicar, agora, o que queremos dizer. Um dado é simplesmente uma sequência de símbolos, gramaticalmente bem formada, mas cuja interpretação é pouco clara. O exemplo mais extremo disto é uma cadeia de uns (1) e

zeros (0) que, codificados na memória de um computador, pode ser uma instrução, um dado numérico ou um caractere que pode ser impresso. Menos extremo que isso é, por exemplo, uma célula de uma planilha de cálculo que nos indica \$118. Se no contexto da planilha o símbolo \$ indica dólares americanos, é possível entender o conteúdo da célula: é um dado que significa cento e dezoito dólares americanos. Mas esse é um nível sintático, muito pouco útil, não nos serve para entender muito. Precisamos de mais contexto para subir desse nível sintático a um nível semântico. Em um contexto, esse dado se converte em uma mensagem. Digamos que esse dado está em uma célula, em uma coluna sob o título “Preço Futuro do Barril de Óleo Cru em Tirkít”. Agora, a combinação do valor da célula com o valor do título nos dá uma interpretação válida e, possivelmente, única da mensagem. De qualquer forma, sem um propósito para a decisão, ficamos na mensagem. Para que a mensagem tenha informação é necessário que sirva para a nossa decisão. Assim, se a decisão é vender ou não vender meus valores futuros em petróleo, esta mensagem contém informação.

Nome do cargo	Nome do cargo ao qual reporta	Experiência	Formação profissional desejável ou requerida	Contatos dentro da organização	Contatos fora da organização
<b>Missão do cargo</b>					
Funções do cargo	O que faz	Como faz	Com quem faz	Indicadores de gestão	Habilidades requeridas

Tabela 6.7: Missão e Funções

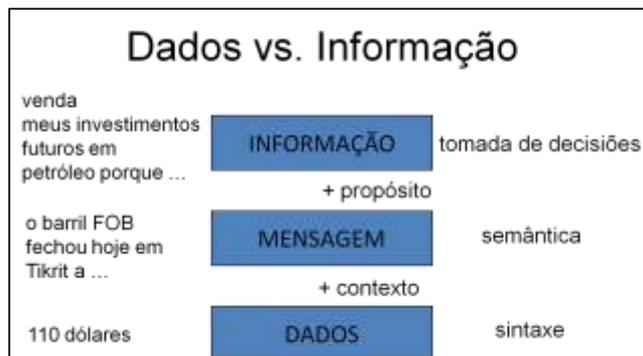


Figura 6.4: Dados vs. Informação

Um bom sistema de informação se apoia tanto em dados confiáveis como na construção de indicadores que permitam visualizar facilmente a decisão. Por exemplo, para a decisão que usamos como exemplo nesta seção, apresentada na Figura 6.5, uma linha paralela ao eixo X do tempo pelo ponto 105 do eixo dos Preços Futuros constitui um indicador do ponto de venda

segundo um critério fixado de antemão. Quando o gráfico dos preços ultrapassa o limite, a decisão é automática ou, pelo menos, pode ser automatizada.

Partindo das decisões que devem ser tomadas, é definido quais indicadores permitem tomar a decisão com confiança. Por exemplo, uma decisão típica de um líder de projeto é se ele deve redefinir o escopo dos requisitos com os quais se comprometeu. Para poder decidir isso, necessita um indicador que aponte se, na data de entrega, o produto estará finalizado com a qualidade prevista. Um possível indicador surge de um método conhecido como valor agregado, EVM (*Earned Value Management*). O interessante do EVM é que é mais simples do que a forma como é apresentado na literatura. Imaginemos que (a) são seis horas da tarde e você está sozinho em seu escritório, pronto para ir para casa, (b) recebe uma ligação de seu primo e melhor amigo: ele teve um acidente com seu automóvel e está detido em uma cidade distante 375 quilômetros de onde você se encontra ao receber a ligação. Se você não chegar na delegacia em menos de quatro horas, seu primo passará a noite na prisão. Se chegar antes e pagar a fiança, seu primo dormirá em casa.

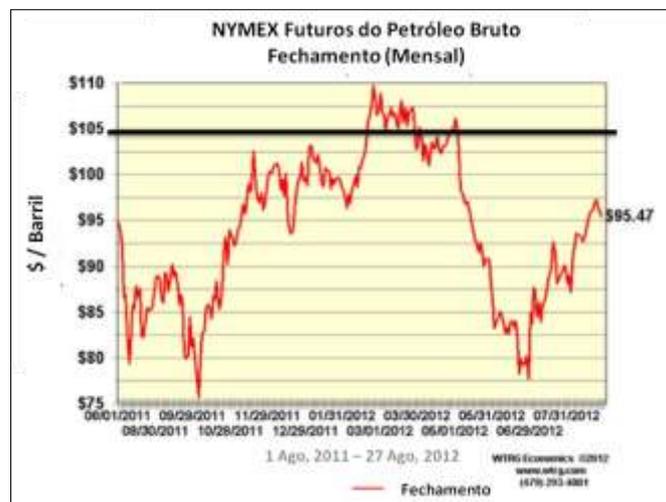


Figura 6.5: Gráfico sobre Preços Futuros do Petróleo

Seu carro está com o tanque parcialmente cheio e você acha que ele possui autonomia e velocidade necessárias para chegar a tempo. Você vai ao caixa eletrônico para retirar dinheiro e em poucos minutos está a caminho. O que pode acontecer? Na verdade, pode ser que o trânsito do horário de pico faça com que você chegue depois da delegacia fechar e seu primo passará a noite entre baratas e outros animais. Como não quer perder tempo, pode ser que no caminho fique sem combustível, longe de um posto e seja obrigado a caminhar perdendo um tempo valioso e possivelmente, para piorar, o carro seja roubado no acostamento. Como saber se estas coisas podem acontecer?

Tem-se dois indicadores: quantos quilômetros o seu automóvel percorre por hora e quanto combustível consome, seja por quilômetro ou por hora. Claro, o primeiro indicador é conhecido como "velocidade" e voltaremos a encontrá-lo neste capítulo quando falarmos com mais detalhe da estimativa. O segundo é o consumo, mas é diferente considerá-lo por quilômetro ou por hora. Por exemplo, é possível diminuir o consumo a zero se alguém detém o automóvel, mas então perde-se o objetivo de resgatar o primo da prisão. Com EVM estes dois indicadores são calculados. O primeiro é o desempenho no tempo e é equivalente à velocidade. O segundo é o desempenho de custos e é equivalente ao consumo.

Em um automóvel medimos os quilômetros percorridos, enquanto que em um projeto de *software* devemos medir o avanço por meio do tamanho do produto ou o número de tarefas que são completadas. Como as tarefas não são todas idênticas e às vezes é difícil compará-las, é melhor tomar uma medida de tamanho (veremos uma mais adiante). Um bom indicador nos mostra, em um golpe de vista, se estamos chegando a tempo ou não. Em EVM são calculados índices, comparando o realizado e o planejado: um valor exatamente igual a 1 é interpretado como estar justo e alinhado ao planejado, enquanto que um valor maior que 1 significa que estamos adiantados em relação ao planejado e um valor menor que 1, que estamos atrasados.

Idealmente, para cada tarefa existe um produto que é resultado de sua execução. Definimos aqui cinco medidas básicas [PUTNAM e MYERS, 2003] que devem estar presentes desde o primeiro momento, na definição inicial dos processos. O que vamos medir, e para quê, deve ser parte integral da definição das tarefas. Desse modo, são estabelecidas de imediato as condições para controlar os processos (uma tarefa não é mais que a definição, instanciação ou execução de um passo de um processo). Essas medições são: tamanho e número de defeitos relacionados com o produto, e duração, esforço e custo da tarefa. Com isto em mente, a T<sup>2</sup> começou a revisão de seus processos e completou a definição de medições que tinha deixado pendente nesse nível.

O seguinte nível é o dos indicadores de gestão que exigem os novos papéis. Outra vez, a sala de *design* vê nossos velhos conhecidos reunidos. Máximo facilita a criação de uma tabela de riscos associados (Tabela 6.8) a uma possível má definição do sistema de decisão<sup>62</sup> que os leva a estabelecer medidas para evitá-los ou minimizá-los.

<b>risco:</b>	<b>mitigação:</b>
o sistema de decisão ser definido arbitrariamente sem considerar a visão estratégica	estabelecer objetivos de medição que correspondam à visão estratégica, considerando as decisões derivadas
as medidas e indicadores utilizados nos projetos serem decididos sem considerar as necessidades organizacionais	decidir as medidas a serem usadas nos projetos em conjunto pela gerência do projeto e o PMO
cada projeto ter uma versão diferente do sistema de decisão	especificar claramente o procedimento de medição com responsabilidades e controles
ter-se um sistema de decisão mas não se ter dados	controlar para que os dados requeridos sejam coletados e analisados
não ser possível reproduzir uma decisão baseada em dados porque os dados não são guardados	guardar os dados e os resultados das análises de maneira que possam ser recuperados
nem todos os interessados entenderem o porquê das decisões	difundir os resultados e as análises das medições obtidas

**Tabela 6.8: Riscos Associados**

Como acontece sempre que Máximo facilita a reunião, os resultados coincidem com os esperados pelo MR-MPS-SW:

<b>Medição (MED)</b>	
<b>MED1</b>	<b>Objetivos de medição são estabelecidos e mantidos a partir dos objetivos de negócio da organização e das necessidades de informação de processos técnicos e gerenciais;</b>
<b>MED2</b>	<b>Um conjunto adequado de medidas, orientado pelos objetivos de medição, é identificado e definido, priorizado, documentado, revisado e, quando pertinente, atualizado;</b>
<b>MED3</b>	<b>Os procedimentos para a coleta e o armazenamento de medidas são especificados;</b>
<b>MED4</b>	<b>Os procedimentos para a análise das medidas são especificados;</b>
<b>MED5</b>	<b>Os dados requeridos são coletados e analisados;</b>
<b>MED6</b>	<b>Os dados e os resultados das análises são armazenados;</b>

<sup>62</sup> A T<sup>2</sup> decidiu que seu sistema de apoio à decisão baseado em dados, que outras empresas chamam de seu sistema de medição, seja denominado "sistema de decisão". É, em parte, um nome que não representa completamente a realidade; mas, por enquanto, vejamos para onde nos leva.

**MED7 Os dados e os resultados das análises são comunicados aos interessados e são utilizados para apoiar decisões.**

**Tabela 6.9: Processo MEDIÇÃO [SOFTEX, 2012a]**

Máximo compartilha com o grupo mais dois ativos, a planilha de definição de medidas (Tabela 6.10) e a planilha de definição de indicadores (Tabela 6.11).

A situação a respeito das expansões fica então assim definida: trabalharão com os contadores internamente. Os gêmeos conduzirão suas respectivas equipes para desenvolver o novo sistema e integrá-lo ao existente, ao qual por sua vez darão manutenção. Não buscarão novos clientes para o sistema atual, pois o financiamento deverá vir da liquidez para manter a T<sup>2</sup> funcionando. Procurarão uma aliança com a empresa de SaaS para serviços de saúde. Em troca de colocar o *know-how* de SaaS e da arquitetura, eles esperam participação no negócio e nas patentes que possam surgir. Com a empresa que quer fundir seu *software* para farmácias com o existente da T<sup>2</sup>, realizarão um convênio pelo qual as duas partes contribuirão para a geração do código necessário.

Nome da Medida		
Estado da Codificação		
Contar o número de programas com codificação e teste unitário finalizados. É utilizada durante a etapa final para compreender o trabalho que falta fazer, comparando com o estimado inicialmente. Permite planejar novas datas de entrega com alguma antecipação.		
Medidas Básicas	Fonte de Dados	Critério de Validade
Data Planejada de finalização do Programa	Planilha de construção de programas do líder técnico do projeto	O Plano está aprovado e sob controle de configuração
Data Real de finalização do programa	Subversion	A data de entrada é gerada automaticamente pela ferramenta
Atributos		
Identificador Único de Programa (UPI)		
Nome do programador responsável		
Data estimada de começo da programação		
Data real de começo da programação		
Data estimada de finalização da programação		
Data real de finalização da programação		
Medidas Derivadas	Cálculos utilizados	
Programas Acumulados Planejados para a Semana	Contagem do número de programas cuja data de finalização segundo o plano cai dentro da semana em questão.	
Programas Acumulados Realizados na Semana	Contagem do número de programas cuja data de finalização segundo o Subversion ocorreu dentro da semana em questão.	

**Tabela 6.10: Definição de Medidas**

#### O QUE ACONTECEU ATÉ AGORA

31. Máximo contribuiu com a definição de papéis trazendo a planilha com a missão, função e comunicação entre eles.
32. Máximo enfatizou a definição dos indicadores de gestão exigidos pelos novos papéis com base nos riscos.
33. Máximo compartilhou com o grupo dois ativos: a planilha de definição de medidas e a planilha de definição de indicadores.

O trabalho é intenso, mas proveitoso. A nova estrutura começa a dar frutos em forma de ações paralelas. O ambiente se energiza e a T<sup>2</sup> começa a preparar sua decolagem. Claudio inicia

ações que incorporarão apoio financeiro à empresa. Marcela prepara os processos com a ajuda de uma estagiária da Escola de Engenharia. Os gêmeos fazem um curso de *Scrum Master* na Califórnia. Alfredo assina acordos e estreita laços. Mariano mantém a continuidade com os clientes. Praticamente sozinho, ele sintetiza o que a T<sup>2</sup> era até a semana anterior. Ana ocupa suas últimas semanas, antes de ser mãe, deixando todas as decisões sobre a arquitetura bem documentadas.

<b>Data de criação</b>	<data>																								
<b>Especificação e Uso do Indicador</b>	<nome do indicador, explicar seu uso>																								
<b>Amostra do Indicador</b>	<div style="text-align: center;"> <p><b>Título, com Data do Relatório</b></p> <table border="1"> <caption>Dados do Gráfico de Barras</caption> <thead> <tr> <th>Data</th> <th>Medida Derivada 1</th> <th>Medida Derivada 2</th> </tr> </thead> <tbody> <tr> <td>data 1</td> <td>5</td> <td>0</td> </tr> <tr> <td>data 2</td> <td>10</td> <td>10</td> </tr> <tr> <td>data 3</td> <td>15</td> <td>60</td> </tr> <tr> <td>data 4</td> <td>20</td> <td>90</td> </tr> <tr> <td>data 5</td> <td>25</td> <td>115</td> </tr> <tr> <td>data 6</td> <td>30</td> <td>120</td> </tr> <tr> <td>data 7</td> <td>135</td> <td>135</td> </tr> </tbody> </table> </div>	Data	Medida Derivada 1	Medida Derivada 2	data 1	5	0	data 2	10	10	data 3	15	60	data 4	20	90	data 5	25	115	data 6	30	120	data 7	135	135
Data	Medida Derivada 1	Medida Derivada 2																							
data 1	5	0																							
data 2	10	10																							
data 3	15	60																							
data 4	20	90																							
data 5	25	115																							
data 6	30	120																							
data 7	135	135																							
<b>Modelo de Análise</b>	<p>Descreve as ações necessárias quando o indicador mostra certos padrões de comportamento. Por exemplo, "se dois valores sucessivos do indicador são menores do que o valor anterior na série, é necessário realizar uma atividade de análise de causa-raiz e notificar a Alta Gerência, o Comitê de Controle do Produto e a área de Gestão de Qualidade sobre a situação".</p>																								
	Xxx																								
<b>Critério de Decisão</b>	<p>Identifica os pontos que disparam novas ações ou servem para determinar futuras investigações</p>																								
	Xxx																								
<b>Medidas Utilizadas</b>	<p>Identifica as medidas básicas ou derivadas que são usadas na construção do indicador</p>																								
	<p>Medida Derivada: xxx  Medida Derivada: yyy  Medida Básica: zzz  Medida Básica: aaa</p>																								

<b>Construção do Indicador</b>	Descreve os passos a seguir na construção do indicador, por exemplo: "Calcule, cada mês, o índice xxx a partir dos valores médios de zzz e aaa. Faça o mesmo para a medida derivada yyy, que é a média ponderada do valor hora da equipe de projetos. Crie um diagrama de barras como exemplificado acima. Pode-se utilizar a planilha "amostra" para gerar o gráfico."
	Medida Derivada: xxx Medida Derivada: yyy Medida Básica: zzz Medida Básica: aaa

Tabela 6.11: Definição de Indicadores

## 6.6 Protegendo os Ativos

Se os produtos que temos serão misturados com componentes ou sistemas externos, eles devem estar protegidos. Mesmo se não for um programa tão extenso, os ramos de um só produto vão se expandindo à medida que surgem variantes no código e há diferentes versões instaladas. Os riscos de não controlar os ativos da empresa são vários:

<b>risco:</b>	<b>mitigação:</b>
Se não sabemos onde estão armazenados os produtos e os resultados, perderemos tempo e possivelmente trabalho	adotar um sistema de gestão de documentação
Se "não se pode encontrar uma folha no bosque", vai ser perdido muito tempo identificando cada parte	adotar um sistema de classificação dos itens armazenados
Se os documentos são atualizados sem que os interessados saibam da atualização, pode haver "conflitos" entre mudanças	adotar um critério de <i>baseline</i> onde só se pode modificar um documento da <i>baseline</i> sob regras rigorosas de controle
Se não sabemos quem mudou o que, nem quando mudou, pode ser difícil reproduzir os motivos e decisões em torno de uma mudança	registrar a história das mudanças de <i>baseline</i> seguir um processo formal para mudar um produto
Se as pessoas não respeitarem as regras e os ativos armazenados em qualquer local e qualquer um tiver acesso e liberdade para modificar, pode-se introduzir muito retrabalho	controlar para que os produtos de trabalho sejam armazenados, possam ser lidos e liberados para uso de acordo com um processo

Tabela 6.12: Riscos Derivados da Falta de Controle de Ativos

Máximo introduz ao grupo a noção de *baseline* de produtos<sup>63</sup>. Uma *baseline* é um conjunto de especificações ou produtos de trabalho que foram formalmente revistos e acordados, que servem como base para um desenvolvimento posterior e que só podem ser modificados por meio do procedimento estabelecido para controle de mudanças. O propósito de definir uma *baseline* é manter o acordo que a produz, sem fixar uma armadilha ao redor dos produtos para que nunca mudem, mas protegendo-os de mudanças não comunicadas. Para que seja possível gerar a *baseline* é necessário definir claramente as responsabilidades a respeito do produto em questão. Para a T<sup>2</sup> existem três tipos de arquivos, segundo o seu conteúdo. Existem os de código, de documentação de *baseline* e de comunicação. Cada um tem seu tipo e suas propriedades, que fazem com que sejam protegidos de diferentes maneiras. Certos

<sup>63</sup> Existem outras *baselines*, por exemplo, *baselines* de cronograma e de comportamento de um processo que veremos mais adiante.

documentos, como os casos de teste, o documento de arquitetura e a especificação funcional, têm uma versão única que é mantida simplesmente com restrições de acesso: um único dono, ou pelo menos um único papel que pode modificá-los. Em outros, um único dono é indesejável, em particular quando se trata de código, onde é desejável que coexistam várias versões. A Tabela 6.13 mostra as propriedades de cada tipo.

Para armazenar cada um dos diferentes tipos, o grupo precisa tomar decisões sobre o sistema que utilizará. Em princípio adotam para o código o modelo de [MOREIRA, 2010] de integração contínua. O mesmo produto *freeware* que estavam utilizando serve para armazenar o código com as características definidas na Tabela 6.13. Há várias possibilidades para os outros dois tipos: *baseline* e comunicações. Podem ser usados produtos denominados DMS (*Document Management Systems*) ou soluções mais simples como um disco virtual com uma estrutura de pastas que sejam definidas a partir da estrutura e tipo do projeto. Por exemplo, para um projeto que adote a estrutura de *Scrum* é possível que existam pastas para o *Backlog* do Produto, o *Backlog* do *Sprint*, as comunicações do Dono do Produto, as medidas, o código e os casos de teste. Pode ser que as pastas, por sua vez, estejam subdivididas. Cada pasta terá seu controle de leitura e escrita de acordo com os papéis e as necessidades detectadas. Por exemplo, o engenheiro de testes pode escrever e apagar na pasta de casos de teste, mas o *Scrum Master*, não. No entanto, o *Scrum Master* poderá ter acesso aos casos de teste para lê-los, mas outras pessoas fora da equipe não podem fazer isso.

TIPO	Dono	acesso	Atualização	versões
código	Múltiplos	check out	check in e merge	múltiplos
<i>baseline</i>	único (por papel)	<i>read only</i>	<i>write</i> do dono	única em vigência
comunicação	Ninguém	<i>read only</i>	não se aplica	não há versões

**Tabela 6.13: Propriedades de Cada Tipo de Item de Configuração**

Outra questão a ser decidida são as auditorias. Há auditorias físicas que controlam que em um repositório não haja nada a menos nem a mais do que o que deve estar ali. Deste modo, protege-se a integridade da organização: não queremos distribuir *Trojans* com nosso *software*. Também existem as auditorias lógicas. Neste caso, o que se coloca em jogo é a sequência com que se chegou ao produto. Por exemplo, se para subir o código fonte à área de testes, é necessário, antes, ter passado por três etapas (digamos que primeiro se criou o caso de teste que devia mostrar um defeito, por falta de código a acrescentar; depois, este código foi criado e vimos que não acusava mais erros, e na terceira etapa ele foi integrado e verificado em relação ao conjunto de casos de teste de regressão, a auditoria lógica procuraria evidências de que todos os passos foram cumpridos antes de aceitar que o código em questão seja subido.

Mais uma vez a responsabilidade de escrever os processos e catalogá-los cai nas mãos de Marcela, mas sua experiência faz com que seja fácil o que a princípio exigia muito esforço. Marcela continua usando os materiais apresentados por Máximo, definindo os produtos de cada procedimento e as medidas e indicadores pertinentes. A nomenclatura dos produtos surge das práticas habituais, o resto foi decidido entre todos. Uma estagiária ajuda a escrever e corrigir os processos.

O conjunto de práticas definidas, quando for implementado, cobre os resultados esperados de Gerência de Configuração:

#### **Gerência de Configuração (GCO)**

- GCO1 Um Sistema de Gerência de Configuração é estabelecido e mantido;**
- GCO2 Os itens de configuração são identificados com base em critérios estabelecidos;**
- GCO3 Os itens de configuração sujeitos a um controle formal são colocados sob *baseline*;**
- GCO4 A situação dos itens de configuração e das *baselines* é registrada ao longo do tempo e disponibilizada;**
- GCO5 Modificações em itens de configuração são controladas;**

<b>GCO6</b>	<b>O armazenamento, o manuseio e a liberação de itens de configuração e baselines são controlados;</b>
<b>GCO7</b>	<b>Auditorias de configuração são realizadas objetivamente para assegurar que as baselines e os itens de configuração estejam íntegros, completos e consistentes.</b>

Tabela 6.14: Processo GERÊNCIA DE CONFIGURAÇÃO [SOFTEX, 2012a]

## O QUE ACONTECEU ATÉ AGORA

34. Como a T<sup>2</sup> irá criar produtos a serem integrados com sistemas externos e usar outros produtos, em vários subsistemas, Máximo introduziu o uso de práticas de gestão de configuração como medida para proteger os diferentes ativos.

## 6.7 Garantindo a Qualidade dos Processos e Produtos

Passaram-se três meses e os novos processos estão em fase de implementação. Há doze pessoas novas trabalhando na T<sup>2</sup> e a sala de *design* passou a ser a sala de uma das duas equipes de *Scrum*. A um quarteirão dali, mais perto do rio, abriram uma filial da T<sup>2</sup>, com a direção geral e financeira, e o lugar que antes era ocupado pelos escritórios de Alfredo, Ana e Claudio agora é um pequeno labirinto de postos de trabalho. As rodas da produção giram aceleradas. Marcela tem uma preocupação: poder julgar se o crescimento altera a qualidade dos produtos, para o qual precisa revisar “o que” e “como” é produzido.

Poderíamos pensar que, em uma cultura que respeita a decisão de escolher os próprios processos, não é necessário que alguém verifique o que vem depois, mas é o conjunto da organização que exige isso. Marcela conhece a técnica de auditorias, mas luta contra o problema de sua falta de valor agregado. Para que, se pergunta, serve o resultado da auditoria à organização? E ao projeto auditado? Marcela vê que coletar dados sobre as não conformidades com as normas e processos é um bom modo de começar a entender como os processos são usados e quais normas são aplicadas, mas vacila ante o retorno que possa ter para o projeto individual.

Quando Máximo explicou a escrita de processos à T<sup>2</sup> partiu da planilha da Figura 5.10, mas depois utilizou uma planilha de folha de cálculo, uma linha para cada passo do processo. As colunas agora representam os itens da planilha: Propósito, Envolvidos, Entradas Exigidas e assim sucessivamente, inclusive com a sequência prevista de que alguns passos sejam condicionais ou possam ser executados em paralelo com outros e não sejam todos simplesmente sequenciais. Com essa folha de cálculo na mão, Marcela teve uma inspiração. Procurando entre suas pastas no computador, encontrou um *webinar* que baixou do *SlideShare*<sup>64</sup>. Voltou a lê-lo e decidiu elaborar auditorias proativas, uma contradição em termos, mas que parece ter sido sugerida pelo *webinar*. Ao invés de estabelecer um programa de auditorias frequentes, Marcela se propõe a participar em eventos de lançamento de etapas, como o começo de cada *sprint* e as apresentações ao dono do produto, e fazer a abertura lembrando, a todos, os passos que foram escolhidos para o processo. Nesse momento, decisões podem ser modificadas, dispensas outorgadas e a atuação pode ser revista. Claro que Marcela participará em todas as retrospectivas.

Como o Dono do Produto é interno à T<sup>2</sup>, Marcela decide utilizá-lo na função de auditor de encerramento. Além disso, os membros da equipe receberão aleatoriamente mensagens de correio eletrônico pedindo que confirmem a execução do processo em todos os seus passos. Desse modo, as auditorias aconteceriam sem entorpecer o desenvolvimento e só seriam sobre aqueles pontos que Marcela, como gerente de qualidade, sintia necessidade de enfatizar.

Para introduzir seus processos de garantia de qualidade, Marcela “rouba” uma técnica de Máximo e apresenta os riscos de não seguir seus processos:

<sup>64</sup> <http://www.slideshare.net/jorgeboria/ga-3-best-practices>

<b>risco:</b>	<b>mitigação:</b>
Se não é possível compartilhar os produtos porque possuem diferentes formatos é possível que tenhamos muitos erros involuntários	verificar se os produtos estão aderentes às regras que são política da organização
Se as pessoas não seguirem os processos que estabelecemos, não é possível assegurar que o que acontece é o que se espera; além de não aprender com os erros, é provável que façamos muitos	verificar se as pessoas seguem as regras e processos que são política da organização
Se detectarmos problemas mas não os comunicamos, eles não serão resolvidos por si mesmos, e é possível que todos percam o respeito aos processos	comunicar os problemas e as não conformidades que possam ser resolvidas
Se não conseguirmos acordo sobre as ações a serem seguidas para corrigir não conformidades, ou se as ações decididas não forem controladas, é possível que as boas práticas sejam abandonadas	quando existirem não conformidades ou problemas, elaborar um plano de ação para que se mantenham as boas práticas ou se corrija o processo

**Tabela 6.15: Riscos de não Auditar**

Considerando esses riscos, os participantes da reunião aprovam o processo de auditoria apresentado por Marcela, o que inclui, para cada projeto:

1. revisar os riscos
2. revisar o plano
3. selecionar processos críticos
4. selecionar produtos críticos
5. planejar auditorias
6. realizar auditorias
7. analisar resultados

Marcela também definiu uma escala de severidade das não conformidades entre o realizado e as normas e padrões (Tabela 6.16).

<b>SEVERIDADE</b>	
SEV 1	INSALVÁVEL: Corrigir imediatamente, escalar de imediato;
SEV 2	GRAVE: Consertar antes do próximo marco, escalar se não for corrigido;
SEV 3	NEGOCIÁVEL : (a) Consertar antes do próximo marco ou (b) auditar na próxima oportunidade, escalar se (a) não for consertado;
SEV 4	SALVÁVEL: Fornecer dispensa, registrar não conformidade, fechar;
SEV 5	REPREENSÍVEL: Advertir, fechar.

**Tabela 6.16: Severidade de Não conformidades em Auditorias**

O procedimento “realizar auditorias” contém um método de escalonamento que também é submetido ao julgamento dos sócios. Dependendo da severidade, o procedimento permite alcançar níveis da direção da T<sup>2</sup> cada vez mais altos quando uma não conformidade não é tratada. O procedimento de escalonamento é completado quando a não conformidade é resolvida ou a pessoa que é escalada decide fechá-lo sem que seja necessário tratá-la. A Figura 6.6 mostra o processo desenhado por Marcela.

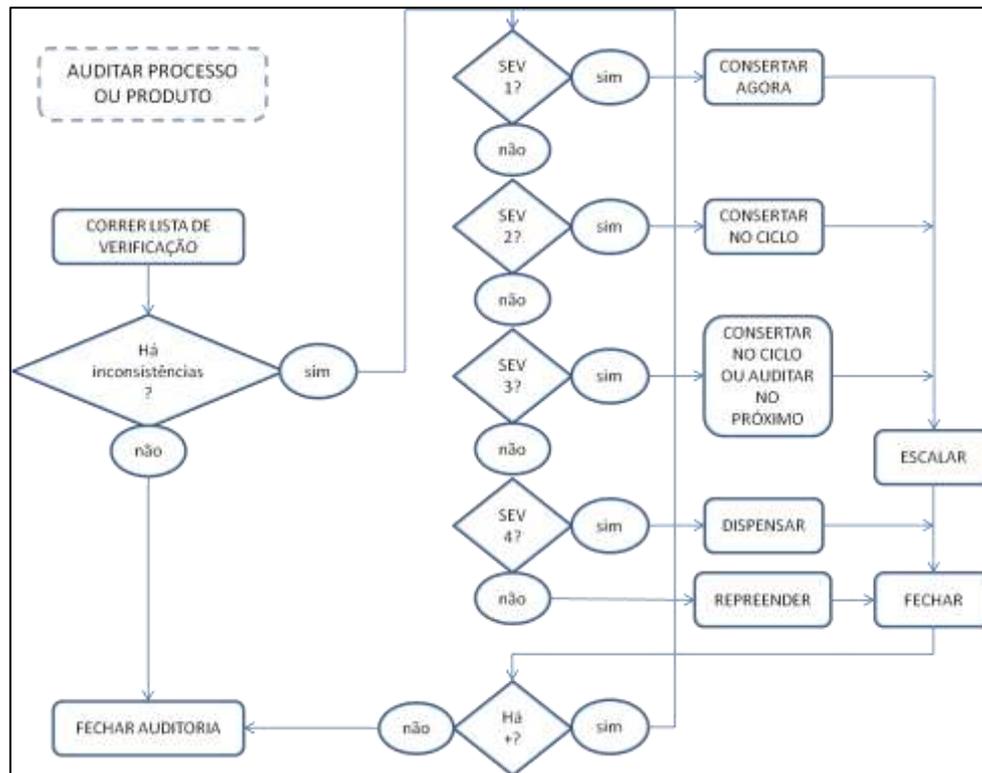


Figura 6.6: Processo de Auditoria da Qualidade

Para completar a reunião, a equipe de direção revê os resultados esperados correspondentes ao processo Garantia da Qualidade.

#### Garantia da Qualidade (GQA)

- GQA1** A aderência dos produtos de trabalho aos padrões, procedimentos e requisitos aplicáveis é avaliada objetivamente, antes de os produtos serem entregues e em marcos predefinidos ao longo do ciclo de vida do projeto;
- GQA2** A aderência dos processos executados às descrições de processo, padrões e procedimentos é avaliada objetivamente;
- GQA3** Os problemas e as não conformidades são identificados, registrados e comunicados;
- GQA4** Ações corretivas para as não conformidades são estabelecidas e acompanhadas até as suas efetivas conclusões. Quando necessário, o escalamento das ações corretivas para níveis superiores é realizado, de forma a garantir sua solução.

Tabela 6.17: Processo GARANTIA DA QUALIDADE [SOFTEX, 2012a]

#### O QUE ACONTECEU ATÉ AGORA

- 35. Marcela utilizou a técnica de identificar riscos para induzir à introdução de práticas de garantia da qualidade.

### 6.8 A pequena fábrica de software com Scrum

Os gêmeos fizeram progressos enormes com o desenvolvimento de *software* utilizando as práticas de *Scrum*, colocando em bom uso o investimento da T<sup>2</sup> em sua formação como *Scrum Master*. Eles se reuniram com os programadores, engenheiros de teste e documentadores contatados por Marcela e depois de duas semanas de esforço, selecionaram duas equipes de trabalho. As equipes vão utilizar as técnicas de *Scrum*, mas com o acréscimo de um *sprint* curto de análise que exigirá a intervenção de Ana, para que a arquitetura esteja bem clara para todos. A equipe de manutenção se dedicará exclusivamente a corrigir os defeitos encontrados e a fornecer à equipe de melhorias a tecnologia de desenvolvimento criando, por exemplo, um *framework* para *scripts* de testes e diversas extensões à ferramenta de integração contínua. A

equipe de desenvolvimento “puro” se encarregará de introduzir melhorias sistemáticas, como novas funcionalidades e refatoração do código, para o contrato com os contadores. Por causa da aceitação do painel *kanban*, os gêmeos o mantêm dentro do grupo de técnicas que vão aplicar nos *sprints*. Para aumentar a visibilidade e a transparência, as equipes utilizarão também um gráfico de “*Burndown*” que mostre os avanços e retrocessos em um *sprint*.

Da mesma forma, os gêmeos compilaram uma lista dos componentes mais importantes do *Scrum* e a transformaram em um quadro que foi pendurado em cada sala. No estilo próprio deles, é uma imagem muito forte que mostra claramente a mensagem. Eles a chamam: “A Morte do *Scrum*” (Figura 6.7).



Figura 6.7: A Morte do *Scrum*

Levando em conta as técnicas que incorporaram, os gêmeos realizam uma auto avaliação do grau de cobertura dos resultados esperados do MR-MPS-SW na T<sup>2</sup>. Revisam os resultados com Máximo e, com satisfação, reafirmam que o caminho escolhido está correto (Figura 6.8). T<sup>2</sup> começa a preparar sua avaliação nível F do modelo MPS de *Software* (MPS-SW).

#### O QUE ACONTECEU ATÉ AGORA

36. Os gêmeos implementaram o *Scrum* em duas equipes conservando os painéis *kanban* para entender o estado do trabalho, acrescentando o gráfico de *Burndown* de tarefas e o *Backlog* do *sprint* como ferramentas de controle do projeto.
37. Tahini-Tahini se prepara para a avaliação de Nível F.

Gerência de Projetos nos Níveis G e F		Kanban	Backlog	Jogo de Planif.	Velocidade Histórica	Taxa de "Queimado"	Retrospectiva	Demo	Scrum Diário
GPR 1	O escopo do trabalho para o projeto é definido;	✓							
GPR 2	As tarefas e os produtos de trabalho do projeto são dimensionados utilizando métodos apropriados	✓		✓					
GPR 3	O modelo e as fases do ciclo de vida do projeto são definidos	✓			✓				
GPR 4	O esforço e o custo para a execução das tarefas e dos produtos de trabalho são estimados com base em dados históricos ou referências técnicas	✓			✓				
GPR 5	O orçamento e o cronograma do projeto, incluindo a definição de marcos e pontos de controle, são estabelecidos e mantidos	✓							
GPR 6	Os riscos do projeto são identificados e o seu impacto, probabilidade de ocorrência e prioridade de tratamento são determinados e documentados		✓						✓
GPR 7	Os recursos humanos para o projeto são planejados considerando o perfil e o conhecimento necessários para executá-lo								
GPR 8	Os recursos e o ambiente de trabalho necessários para executar o projeto são planejados								
GPR 9	Os dados relevantes do projeto são identificados e planejados quanto à forma de coleta, armazenamento e distribuição. Um mecanismo é estabelecido para acessá-los, incluindo, se pertinente, questões de privacidade e segurança								
GPR 10	Um plano geral para a execução do projeto é estabelecido com a integração de planos específicos								
GPR 11	A viabilidade de atingir as metas do projeto é explicitamente avaliada considerando restrições e recursos disponíveis. Se necessário, ajustes são realizados			✓	✓				
GPR 12	O Plano do Projeto é revisado com todos os interessados e o compromisso com ele é obtido e mantido								
GPR 13	O escopo, as tarefas, as estimativas, o orçamento e o cronograma do projeto são monitorados em relação ao planejado	✓	✓			✓			✓
GPR 14	Os recursos materiais e humanos bem como os dados relevantes do projeto são monitorados em relação ao planejado	✓		✓		✓			✓
GPR 15	Os riscos são monitorados em relação ao planejado;						✓	✓	✓
GPR 16	O envolvimento das partes interessadas no projeto é planejado, monitorado e mantido							✓	✓
GPR 17	Revisões são realizadas em marcos do projeto e conforme estabelecido no planejamento							✓	✓
GPR 18	Registros de problemas identificados e o resultado da análise de questões pertinentes, incluindo dependências críticas, são estabelecidos e tratados com as partes interessadas						✓		
GPR 19	Ações para corrigir desvios em relação ao planejado e para prevenir a repetição dos problemas identificados são estabelecidas, implementadas e acompanhadas até a sua conclusão						✓		

**Figura 6.8: Cobertura dos Resultados Esperados com Scrum e Kanban**

#### O QUE ACONTECEU ATÉ AGORA DESDE O PRIMEIRO DIA

1. O consultor Máximo estabeleceu o contato inicial com o cliente, coincidentemente com um problema grave e facilitou sua identificação.
2. Introduziu uma primeira técnica de análise de causas, o diagrama de Ishikawa.
3. Utilizando a técnica chegou, junto aos clientes, à conclusão de que seus processos deixavam muito espaço a problemas como o detectado, a falta de controle das tarefas.
4. Apesar de ter um diagnóstico concreto que já aponta para a melhoria de processos, Máximo se concentrou no problema imediato para evitar um problema ainda maior, começando por definir as características (ou atributos) desejáveis da solução.
5. Máximo sugeriu o método Kanban, sem impô-lo, e foi escutado.
6. Máximo induziu o uso do painel kanban, sem mandar.
7. Introduziu os conceitos de sub-história e estimativa por tamanho.
8. Há uma clara definição do escopo do trabalho a realizar, encarnado na lista de histórias, com sua correspondente estimativa de tamanho.
9. Foi feita uma desagregação das histórias, onde o tamanho destas fazia com que fossem úteis.
10. Os conceitos de história, sub-história, desagregação e tarefa foram entendidos na prática e aplicados. Distingue-se entre "tarefa" e "sub-história".
11. Máximo mostrou que o uso do painel kanban é dinâmico e pode ser modificado.
12. Ficou esclarecido o que significa uma tarefa estar PRONTA, para cada etapa do seu ciclo de vida.
13. Visualiza-se, facilmente, o ciclo de vida de uma história no painel.
14. O controle do estado geral de uma história e as tarefas associadas podem ser lidos no painel.

15. Máximo incorporou melhorias ao documento de Backlog, tornando-o mais sólido e útil.
16. Utilizando uma planilha de riscos, foram analisados os possíveis efeitos indesejados associados com as histórias do projeto.
17. Foi incorporada uma planilha de propostas que permite reunir em um documento dinâmico, o Backlog e o orçamento, além de acrescentar as responsabilidades das partes contratuais.
18. Os compromissos mútuos entre cliente e fornecedor ficam documentados, em especial, o escopo do projeto, representado no Backlog.
19. Máximo introduziu o Scrum diário para avaliar os avanços dia a dia e o estado geral do sprint, incluídos os riscos associados para, com isso, melhorar o controle e proporcionar evidências de vários resultados esperados do MR-MPS-SW.
20. A organização incorpora uma demonstração do produto ao final de cada sprint como critério de aceitação pelo usuário, que serve para evidenciar a sua participação.
21. A análise de lacunas exercida como atividade contínua mostra que os resultados esperados para GRE surgem dos processos já implementados.
22. Máximo introduziu um curto procedimento para definir a estrutura de pastas de um projeto em seu site, o que deixa evidente a implementação de GPR 8 e GPR 9.
23. Máximo definiu e codirigiu várias implementações do procedimento de análise retrospectiva, o que traz evidências de GPR 17, GPR 18 e GPR 19.
24. Máximo redigiu uma política que a organização aprovou para estabelecer os alinhamentos do que se espera dos colaboradores da T<sup>2</sup>.
25. Máximo desenvolveu um processo que permite evidenciar as capacidades de processo de Nível G do MR-MPS-SW para as atividades de planejamento, por meio da proposta e gestão de requisitos, do Backlog do produto e da evolução do painel kanban.
26. T<sup>2</sup> alcançou o nível G do MR-MPS-SW em uma avaliação oficial.
27. Máximo utilizou a técnica dos chapéus de cores para semear o caminho de ideias e, com o diagrama de Ishikawa, ajudar os sócios a decidir que caminho seguir na empresa (crescer ou não crescer).
28. Máximo ajudou a rever os riscos de contratação de fornecedores e a estudar as alternativas de comprar, reutilizar ou construir, introduzindo a técnica da matriz de Pugh.
29. Para poder desenvolver mais opiniões em paralelo, Máximo introduziu a técnica de “polinização cruzada” mediante um oportuno corte na reunião.
30. Utilizando a análise de riscos como ferramenta de exploração, Máximo introduziu as práticas que implementam os resultados esperados do MR-MPS-SW correspondentes à gestão de portfólio (ou carteira) de projetos.
31. Máximo contribuiu para a definição de papéis apresentando a planilha com a missão, função e comunicação entre eles.
32. Máximo enfatizou a definição dos indicadores de gestão que requerem os novos papéis baseando-se nos riscos.
33. Máximo compartilhou com o grupo dois ativos: a planilha de definição de medidas e a planilha de definição de indicadores.
34. Como a T<sup>2</sup> irá criar produtos a serem integrados com sistemas externos e usar outros produtos, em vários subsistemas, Máximo introduziu o uso de práticas de gestão de configuração como medida para proteger os diferentes ativos.
35. Marcela utilizou a técnica de identificar riscos para induzir à introdução de práticas de garantia da qualidade.

36. Os gêmeos implementaram o Scrum em duas equipes conservando os painéis kanban para entender o estado do trabalho, acrescentando o gráfico de Burndown de tarefas e o Backlog do sprint como ferramentas de controle do projeto.
37. Tahini-Tahini se prepara para a avaliação de Nível F.
- 
-

# Parte III – Evolução

## CAPÍTULO 7 - ORGANIZANDO A ORGANIZAÇÃO

### 7.1 Nível E do MR-MPS-SW

Este capítulo tem como objetivo discutir a implementação de processos que são exigidos no nível E do modelo de referência MPS-SW. Para alcançar o Nível E de maturidade é necessário conservar ou alcançar os níveis de maturidade anteriores (G e F), e implementar os processos Avaliação e Melhoria do Processo Organizacional (AMP), Definição do Processo Organizacional (DFP), Gerência de Recursos Humanos (GRH) e Gerência de Reutilização (GRU). Da mesma forma, o processo de Gerência de Projetos (GPR) sofre sua primeira evolução, passando a ter como propósito a utilização de um processo definido para o projeto como base da gestão, por meio de planos bem integrados. Também há uma evolução quanto aos atributos de processos, já que aos atributos de processo AP 1.1, AP 2.1 e AP 2.2 são agregados AP 3.1 e AP 3.2. (Ver Capítulo 4 para maiores detalhes destes atributos de processo).

### 7.2 Uma Empresa em Franco Crescimento

Nem bem terminam as mensagens de felicitação por terem alcançado o Nível F do MR-MPS-SW, os avanços realizados são colocados à prova. As tarefas de desenvolvimento se multiplicam e os ajustes ao sistema para novos e velhos clientes também. Alberto Galarraga exige duplicar o número de equipes para desenvolvimento e manutenção, para o qual utiliza a análise de portfólio que já tinha sido utilizada para decidir o crescimento. Trabalhando com Marcela, que começou há meses um mestrado em administração, tentam duplicar a seleção de pessoal que, embora funcionasse relativamente bem na criação das duas primeiras equipes de *Scrum*, ao ter sido esgotada a fonte para contratação de jovens no último ano de Engenharia da Computação, gerou problemas para conseguir pessoal capacitado. Mesmo quando alguma pessoa foi selecionada e seu perfil é promissor, a incorporação é longa e dolorosa.

Já acostumados às técnicas de Máximo, juntam-se com Marcela para analisar os problemas e identificar as suas causas-raiz. O tema da reunião é o esforço que demanda incorporar novos colaboradores:

1. O perfil do colaborador solicitado não está bem definido
2. Não existe um repositório de conhecimentos que ajude a minimizar o aprendizado individual de técnicas e métodos
3. Não há uma definição do ambiente de trabalho que permita solicitar os recursos necessários com antecipação para que o recurso humano “entre em ação de imediato”
4. Não há uma integração aos processos da T<sup>2</sup> que acelere a formação de equipes
5. Não se leva em conta para a seleção de pessoal o tempo requerido para análise dos candidatos

Como já é tradição, são listadas as medidas que serão tomadas. Claramente, neste caso, é suficiente a negação dos problemas: definir bem os perfis, montar um repositório de conhecimentos, definir o ambiente de trabalho, etc.

Marcela é a encarregada de realizar a busca, seleção e contratação na nova estrutura e, assim, cabe a ela a maioria das atividades que surgem da reunião. Trabalha com os gêmeos na elaboração de um modelo de competências<sup>65</sup> para membros da equipe, para eliminar

---

<sup>65</sup> *The Art and Science of Competency Models. Pinpointing Critical Success Factors in Organizations* LUCIA, A.D., LEPSINGER, R., 2007

rapidamente a primeira parte do problema. Com base no que foi realizado, pensa em estender o modelo a todos os papéis que existem dentro da organização. Começa utilizando o formulário de missão e funções da T<sup>2</sup> que é uma variação da planilha que apresentamos no Capítulo anterior. Como exemplo, colocamos na Figura 7.1 o formulário completo para o cargo de Engenheiro de Testes.

<b>Missão e Funções</b>
<b>Nome do Cargo</b>
Engenheiro de Teste
<b>Nome do Cargo ao Qual se Reporta</b>
Gerente de Teste (matricialmente), <i>Scrum Master</i> (tecnicamente), Equipe de <i>Scrum</i> (funcionalmente)
<b>Missão (Por que existe o cargo)</b>
A missão dos Engenheiros de Teste é detectar e informar os defeitos e problemas nos produtos da empresa, para que esta possa tomar corretamente as decisões de negócio sobre a qualidade dos produtos em relação a estarem prontos para serem liberados para uso. É nossa política exceder a satisfação do cliente em nossas entregas de produto.
<b>Fator(es) Crítico(s) de Sucesso (Os aspectos mais visíveis do cargo)</b>
<ol style="list-style-type: none"><li>1. Falhas detectadas pelo usuário (prévio a e nos primeiros seis meses depois do lançamento)</li><li>2. Satisfação dos engenheiros de <i>software</i> com os relatórios recebidos</li><li>3. Eficiência dos casos de teste criados (rendimento em defeitos por caso sobre total de defeitos)</li><li>4. Eficiência na execução de casos de teste e relatório dos resultados</li></ol>
<b>Medidas de Desempenho (Como se pode medir o êxito do cargo)</b>
<ol style="list-style-type: none"><li>1. Número de defeitos, não previamente detectados, que foram encontrados no teste de aceitação</li><li>2. Número de defeitos detectados pelos usuários nos primeiros seis meses de uso que não foram previamente detectados</li><li>3. Número de defeitos informados não reproduzíveis a partir do relatório</li><li>4. Número de relatórios rejeitados pelos engenheiros de <i>software</i></li><li>5. Número de defeitos que passam de “aberto” a “fechado” sem passos intermediários depois do relatório</li><li>6. Produtividade média dos casos de teste criados</li><li>7. Número total de relatórios aceitos pelos engenheiros de <i>software</i></li><li>8. Número de casos executados por unidade de tempo</li><li>9. Número de defeitos informados por unidade de tempo</li></ol>
<b>Funções (O que faz o ocupante do cargo para cumprir a missão)</b>
<ol style="list-style-type: none"><li>1. Interpreta o Plano de Testes de <i>Software</i></li><li>2. Revê e testa documentos de requisitos (com técnicas como Gráficos de Causa e Efeito)</li></ol>

<ol style="list-style-type: none"><li>3. Cria Procedimentos de Teste</li><li>4. Cria Casos de Teste</li><li>5. Executa os procedimentos de teste usando os casos de teste</li><li>6. Informa os resultados de cada teste</li><li>7. Cria casos de teste críticos para assegurar a identificação do defeito</li><li>8. Executa os casos de teste sob a proteção de monitoramento de cobertura e analisa as medições resultantes</li><li>9. Escreve <i>scripts</i> de casos de teste para automatizar sua execução</li><li>10. Entende as prioridades entre casos e otimiza o tempo para aumentar a produtividade</li></ol>
<b>Comunicação dentro da organização</b>
<ul style="list-style-type: none"><li>• Com os engenheiros de <i>software</i>, para expandir o realizado em testes executados e os respectivos relatórios</li><li>• Com o gerente de testes, para receber instruções acerca das tarefas a realizar e informar as atividades realizadas e situações excepcionais</li><li>• Com os analistas de negócios, para desenvolver requisitos verificáveis por testes e receber sugestões para o desenvolvimento de casos de teste</li><li>• Com os analistas de negócios, para detalhamento do que foi informado nos documentos de requisitos</li><li>• Com o <i>Scrum master</i> e a equipe do <i>Scrum</i>, para priorizar e classificar os defeitos encontrados (leve, sério, fatal, etc.)</li></ul>
<b>Comunicação fora da organização</b>
<ul style="list-style-type: none"><li>• Em casos particulares, com os usuários finais ou um representante deles, para planejar e realizar o teste de aceitação do usuário</li></ul>
<b>Qualificação para o cargo</b>
<ul style="list-style-type: none"><li>• Três ou mais anos em posições técnicas relacionadas (redator técnico, programador) ou dois anos de experiência de teste em projetos similares</li><li>• Experiência com integração contínua</li><li>• Experiência com automatização de testes de <i>software</i></li><li>• Clareza de pensamento e capacidade de comunicação</li><li>• Boa memória visual</li><li>• Atenção aos detalhes</li><li>• Experiência em TDD desejável</li><li>• Experiência em métodos ágeis desejável</li></ul>
<b>Educação formal</b>
<ul style="list-style-type: none"><li>• Diploma superior ou técnico relacionado com a profissão</li><li>• Certificações profissionais MPS, MS ou semelhantes, desejáveis.</li></ul>

**Figura 7.1: Formulário Completo Missão e Função para Engenheiro de Teste**

Esta é a definição do cargo. Para construir o modelo de competências, Marcela precisa considerar três dimensões de cada papel: as competências básicas, as competências técnicas e as competências específicas. As mesmas competências básicas são exigidas para todas as posições. Uma competência básica, também chamada de nuclear ou elementar, é definida como um conhecimento, habilidade ou comportamento essencial para que alguém possa atuar

corretamente como pessoal efetivo da T<sup>2</sup>. As competências básicas são aplicadas geralmente a todos os cargos da mesma maneira. Uma competência técnica é uma habilidade particular relacionada, especificamente, com o papel em questão, por exemplo, o engenheiro de testes precisa ser capaz de realizar o *design* de um caso de testes. As competências específicas de um cargo são aquelas que são próprias dos processos da T<sup>2</sup>, por exemplo, um *Scrum Master* deve possuir as competências que lhe permitam fazer funcionar sua equipe, mas, além disso, deve fazer da maneira esperada na T<sup>2</sup>, quer dizer seguindo suas políticas, processos e procedimentos. Tipicamente, as competências básicas e as técnicas são utilizadas no processo de seleção, embora às vezes os colaboradores possam ter acesso à capacitação para adquirir ou aumentar suas competências técnicas, e existem programas de sensibilização a respeito das competências básicas.

Em seguida é apresentada uma lista das competências básicas que Marcela e os gêmeos entendem serem indispensáveis para trabalhar na T<sup>2</sup>, com a descrição do comportamento esperado em cada uma:

- Ética e Integridade: Comporta-se sempre com honradez e admite os erros quando são assinalados, não assume ideias alheias como próprias;
- Serviço ao Cliente: Cumpre com os objetivos do projeto a respeito dos níveis de serviço ao cliente, esforçando-se constantemente para alcançá-los e até superá-los;
- Comunicação: Comunica-se com eficiência de forma verbal e escrita;
- Resolução de problemas: Desenvolve estratégias eficazes ajustadas às necessidades do negócio e resolve problemas;
- Flexibilidade: Demonstra flexibilidade nos papéis em que atua e gerencia suas mudanças de maneira que tenham como resultado um desempenho produtivo;
- Tecnologia: Utiliza equipamentos e tecnologia de maneira segura, eficiente e eficaz;
- Segurança: Cumpre com as normas de segurança, observa as práticas de trabalho seguras e proporciona informação sobre questões de segurança;
- Autogestão: Maximiza o próprio tempo e aproveita seu talento para alcançar os objetivos da organização;
- Crescimento: Procura oportunidades para inovar e melhorar continuamente;
- Adaptação: Desenvolve enfoques eficazes para a gestão de si mesmo na mudança organizacional;
- Trabalho em equipe: Trabalha de forma eficiente com a equipe de trabalho e também com aqueles que estão fora da linha formal de autoridade para conseguir os objetivos organizacionais;
- Prudência: Utiliza os recursos baseando-se sensatamente nas prioridades da organização.

Além disso, o modelo que desenvolvem contém as competências técnicas para cada um dos cargos. Ainda não desenvolveram um modelo completo, que deverá conter um mecanismo de avaliação do grau de adequação do pessoal ao modelo, assim como as competências específicas, cujos detalhes surgirão dos processos que devem seguir os papéis. Por enquanto, Marcela se conforma com uma descrição do que procura em cada atividade que faz parte da cadeia de valor que é atendida pelo cargo, o que constitui conteúdo suficiente para suas buscas técnicas.

Por exemplo, para o cargo de Engenheiro de Testes, no processo de “*Design* da Estratégia de Testes”, o propósito é encontrar a melhor estratégia possível para a produtividade dos testes. Um Engenheiro de Testes deve ser capaz de interpretar o Plano de Testes de *Software* e escolher uma estratégia para os testes, baseada nos objetivos de negócios e nos riscos. Quando participa da “Análise de Requisitos”, espera-se que seja capaz de encontrar uma alta porcentagem de não conformidades e incompletudes, o que consegue ao revisar e testar

documentos de requisitos, com técnicas como Gráficos de Causa e Efeito. Utiliza estas técnicas de geração sistemática de testes para identificar requisitos omissos ou divergentes. Estes são os elementos técnicos que servirão para a seleção, as entrevistas e os exames de aptidão dos candidatos.

Armada com estas definições, Marcela começa a procurar pessoal para ocupar oito cargos de engenheiro de *software* e cinco de engenheiros de testes, destes últimos pelo menos um deles deverá ter experiência em documentação de *software*. Buscando no mercado, as etapas de divulgação da oportunidade, os filtros iniciais das respostas, das comunicações, das entrevistas, da preparação de ofertas e da contratação tiveram como resultado uma demora de várias semanas. Marcela volta a reunir os sócios, para propor uma atitude pró-ativa na busca. Usando os números de suas buscas e dados do estudo de busca de pessoal que a auxilia, adverte sobre as demoras que se apresentam na contratação e do risco de não contarem com o pessoal necessário no começo de um projeto, o que impedirá atingirem os objetivos e cumprirem com os compromissos com os clientes. Propõe uma atividade de mitigação que consiste em gerar uma base de dados dos candidatos. Seu mecanismo será o seguinte: uma vez por mês os sócios se reunirão para rever o portfólio de projetos e fazer ajustes no plano estratégico. Nessa reunião serão estabelecidos os potenciais cargos a cobrir. Com esse fundamento, Marcela estabelecerá um plano tático de contratações.

Marcela apresenta a tabela (Tabela 7.1) com as atividades do processo de recrutamento e incorporação de pessoal. Além disso, como dados de ajuste, Marcela tem duas tabelas, uma para desenvolvedores e outra para pessoal de testes, que medem a porcentagem de respostas positivas a um evento (por exemplo, quantas buscas de currículo em uma base de dados de pessoal dão resultados positivos para cada tipo de cargo) e o tempo de demora de cada atividade. Como exemplo, Marcela apresenta uma tabela com as respostas positivas percentuais em cada evento significativo, para cada uma de muitas posições relacionadas com o desenvolvimento, que conseguiu na agência que a ajuda em suas buscas (Tabela 7.2). Desse modo, pode calcular quantas instâncias de cada atividade precisa realizar em cada caso e com que antecipação. Vendo os dados da tabela para o pessoal de testes pode-se deduzir que é preciso realizar seis ofertas (5,6 para ser preciso) para obter cinco aceitações, o que exige que sejam feitas 5,7 entrevistas técnicas, 6,4 entrevistas funcionais, que sejam enviados a 12,9 candidatos selecionados entre 143,2 currículos da base de dados. Se as oportunidades não se concretizam, a busca pode ser igualmente útil se surgirem outras novas necessidades ou se simplesmente “se esfria” até que apareça algo. O preço que se paga é menor do que o de adiamentos.

	Atividade
1	Filtrar os candidatos da BD
2	Escolher entre os filtrados
3	Agendar entrevistas funcionais
4	Realizar entrevistas funcionais
5	Agendar entrevistas técnicas
6	Realizar entrevistas técnicas
7	Realizar oferta inicial
8	Obter aprovação
9	Realizar oferta final
10	Contratar
11	Introduzir o conhecimento da empresa
12	Capacitar no papel

**Tabela 7.1: Atividades de Recrutamento e Incorporação de Pessoal**

Deste modo, Marcela tem um processo de incorporação de pessoal e um sistema de medição que permitem antecipar o tempo e o esforço do recrutamento. Isto permite atender as previsões para a seleção, agora falta eliminar os riscos no processo de incorporação, que se sintetizam no atraso dos projetos porque demora muito para levar o pessoal incorporado a um

nível de rendimento aceitável dado o excesso de treinamento exigido para incorporar os novos colaboradores às equipes.

Evento	Oracle DBM	Designer Java	Programador Sr Java	Engenheiro de Testes Sr.	Scrum Master	Especialista em Estatística
P1: Filtro inicial	2%	17%	25%	9%	12%	1%
P3: Propor entrevistas	40%	28%	55%	50%	25%	9%
P4: Entrevista funcional	80%	55%	33%	88%	92%	40%
P5: Entrevista técnica	95%	92%	88%	98%	95%	99.80%
P6: Oferta de emprego	67%	94%	97%	90%	98%	99.90%

Porcentagem que se acha na base  
 Porcentagem dos que tem na base aos que se escolhe para entrevistar  
 Porcentagem dos entrevistados em primeira instancia que são convidados para uma outra entrevista  
 Porcentagem dos entrevistados em segunda instancia que merecem ser oferecidos o cargo  
 Porcentagem dos que são oferecidos o cargo que aceitam

**Tabela 7.2: Porcentagem de Êxito em Atividades Seleccionadas por Tipo de Cargo**

Em primeiro lugar, Marcela propõe reforçar a estrutura e o papel de sua Gerência de Recursos Humanos. Aqui estão seus argumentos, novamente analisados como riscos para a T<sup>2</sup>:

Desenvolvimento de Recursos Humanos	
risco:	mitigação:
se crescermos sem previsão de recursos humanos vamos ter muito atraso para incorporar pessoal e nem sempre poderemos responder à demanda	estudar as necessidades estratégicas da organização e prever as necessidades de recursos, antecipando-nos a elas identificar os candidatos a ocupar postos e desenvolver relações para recrutá-los
dados nossos processos específicos, é provável que o período de aprendizado e ajuste às necessidades da T <sup>2</sup> seja muito longo para nossos negócios.	identificar as necessidades estratégicas de capacitação de pessoal gerar um plano de longo prazo para atender as necessidades de capacitação implementar as partes principais do plano no curto e médio prazos possuir registros claros dos treinamentos realizados
é provável que haja treinamentos que não tenham o valor esperado e que o investimento seja perdido	avaliar se os treinamentos cumprem com seu propósito
se quisermos ser proativos na melhoria de desempenho, devemos entender como avaliá-lo e melhorá-lo constantemente ou corremos o risco de investir no que não tem retorno	definir métodos objetivos de medir desempenho e utilizá-los para melhorar o rendimento do pessoal e identificar oportunidades de melhorias
se desperdiçarmos o conhecimento deveremos recriá-lo periodicamente, a um custo muito alto	ter uma estratégia para gerar, captar, difundir e aproveitar o conhecimento formar grupos de interesse em cada disciplina e apoiá-los estabelecer meios de difundir e compartilhar conhecimento e habilidades entre pares

**Tabela 7.3: Riscos da T<sup>2</sup> Derivados de Políticas Pobres em Recursos Humanos**

Felizmente várias atividades já estão sendo realizadas. Por exemplo, o método adotado para rever o portfólio de projetos, agora, inclui estudar as necessidades estratégicas da organização e prever as necessidades de recursos, o que Marcela propõe aprofundar ainda mais utilizando as ferramentas de um modelo de competências. Da mesma forma, o modelo de competências,

junto com o processo de recrutamento, vai ajudar na identificação dos candidatos a ocupar postos e desenvolver relações para recrutá-los. Marcela se propõe a contratar um profissional da área de recursos humanos que tenha conhecimentos de modelos de competências e *design* instrucional, uma combinação não frequente, mas tampouco extravagante. Juntando isto com a análise mensal do portfólio, é possível identificar as necessidades estratégicas de capacitação de pessoal e, desse modo, gerar um plano de longo prazo para atender às necessidades de capacitação e implementar as partes principais do plano no curto e médio prazo, como plano tático. Certamente, será preciso estabelecer registros explícitos dos treinamentos realizados e avaliar se eles atendem a seus objetivos.

Nesse sentido, o modelo de competências adotado permite, ou deverá permitir, a definição de métodos objetivos para medir o desempenho e utilizá-los para melhorar o rendimento do pessoal e identificar oportunidades de melhorias. Medir esse desempenho, antes e depois do treinamento realizado, permite relacionar a melhoria de desempenho à capacitação. Como Marcela tem em mente um modelo de pessoal que chega aos custos individuais (o custo derivado do salário incluindo encargos sociais e os custos de manter o cargo em funcionamento, como eletricidade, licenças de *software* e *hardware*, amortização dos metros quadrados ocupados, etc.) e desempenho individual (a receita que pode ser relacionada ao desempenho da pessoa no cargo), é possível atribuir um valor monetário à diferença de produtividade, que pode ser medido com relação ao custo da capacitação.

É possível, da mesma forma, fazer análises do período de reembolso, o valor da capacitação e outras análises financeiras. A ponto de receber o título de Mestre em Administração, Marcela confia que as análises financeiras aplicadas a todo tipo de investimentos renderão benefícios na hora de tomar decisões. Apesar de que isto possa parecer uma forma de “matar mosquitos com canhões”, é uma pauta cultural necessária para tomar decisões baseadas em dados e não em sensações. Nas palavras de D. Edwards Deming, “*In God we trust, all the others bring data*” (Em Deus confiamos, os demais trazem dados)<sup>66</sup>.

Para mostrar que um modelo de competências é possível, Marcela oferece um exemplo parcialmente completo para o cargo de gerente de contas, que foi enviado por Máximo. Na versão utilizada, cada passo crítico do processo de vendas de um projeto novo é listado na primeira coluna. Depois há cinco colunas representando o grau de competência do gerente de contas, do “leigo” ao “especialista”. Debaxo de cada grau há uma descrição do que se entende por esse grau (Tabela 7.4). Da mesma forma, a Tabela 7.5 mostra a lista de avaliação correspondente à primeira tarefa. Desse modo, é possível relacionar o rendimento com os indivíduos e medir o resultado da capacitação como a diferença de produtividade antes e depois do evento.

COMPETÊNCIA POR NÍVEIS DE UM GERENTE DE CONTAS					
	COMPETÊNCIA				
Fase	1	2	3	4	5
Tarefa	Leigo	Principiante	Júnior	Sênior	Especialista
Apoio à Venda					
Participa da reunião inicial	Não tem ideia sobre o que se espera dele ou dela	Faz anotações e perguntas relacionadas	Faz anotações, perguntas relacionadas e esclarecimentos	Faz anotações, perguntas relacionadas e esclarecimentos bem como participa nas discussões	Conduz e facilita a atividade

<sup>66</sup> Na verdade, não há confirmação de que tenha sido Deming que disse isto, mas é parte do mito que o acompanha. Ver [http://en.wikipedia.org/wiki/W.\\_Edwards\\_Deming](http://en.wikipedia.org/wiki/W._Edwards_Deming)

<b>Realiza a apresentação do método de desenvolvimento da T<sup>2</sup></b>	Não conhece o material e não é capaz de apresentá-lo	Ajuda na explicação sobre o uso dos materiais no exercício	Apresenta o material com mínimos problemas e conduz os exercícios	Prepara a apresentação, apresenta o material sem problemas e conduz os exercícios	Prepara a apresentação, apresenta muito bem o material e conduz os exercícios
<b>Realiza ajustes à proposta frente ao cliente ante pedidos expressos de mudança</b>	Não sabe como se comportar frente às mudanças no projeto	Tenta seguir o processo de controle de mudanças, mas se perde e não consegue resultados	Avalia mudanças medindo seu impacto em custo e tempo de entrega e gerencia as próprias mudanças	Mantém a integridade da proposta sem se confrontar com o cliente e assegura que as mudanças de impacto se refletem no escopo da proposta	Coordena as mudanças com todos os interessados e equilibra os interesses coletivos para chegar ao ganha-ganha

Tabela 7.4: Primeira Parte de um Modelo de Competências

LISTA DE AVALIAÇÃO					
<b>Fase de Apoio à Venda</b>					
<b>Participa da reunião inicial</b>	<b>Tarefa</b>	Anota	Faz perguntas adequadas para que todos entendam melhor a discussão	Participa e facilita a reunião	É capaz de sintetizar os resultados alcançados
	<b>Quase Nunca (0% - 10%)</b>				
	<b>Muito Pouco (1% - 33%)</b>				
	<b>Com frequência (34% - 66%)</b>				
	<b>Quase Sempre (67% - 95%)</b>				
	<b>Sistematicamente (96% - 100%)</b>				

Tabela 7.5: Lista de Avaliação Correspondente à Primeira Tarefa

Marcela implementa as ações e liga para Máximo para revisar sua pertinência. Máximo mostra-se satisfeito com os progressos conseguidos e sugere que as práticas sejam avaliadas em comparação com os resultados do modelo MR-MPS-SW, desta vez com o processo Gerência de Recursos Humanos:

#### Gerência de Recursos Humanos (GRH)

- GRH1** As necessidades estratégicas da organização e dos projetos são revistas para identificar recursos, conhecimentos e habilidades requeridos e, de acordo com a necessidade, planejar como desenvolvê-los ou contratá-los;
- GRH2** indivíduos com as habilidades e competências requeridas são identificados e recrutados;
- GRH3** As necessidades de treinamento que são responsabilidade da organização são identificadas;

GRH4	Uma estratégia de treinamento é definida, com o objetivo de atender às necessidades de treinamento dos projetos e da organização;
GRH5	Um plano tático de treinamento é definido, com o objetivo de implementar a estratégia de treinamento;
GRH6	Os treinamentos identificados como responsabilidade da organização são conduzidos e registrados;
GRH7	A efetividade do treinamento é avaliada;
GRH8	Critérios objetivos para avaliação do desempenho de grupos e indivíduos são definidos e monitorados para prover informações sobre este desempenho e melhorá-lo;
GRH9	Uma estratégia apropriada de gerência de conhecimento é planejada, estabelecida e mantida para compartilhar informações na organização;
GRH10	Uma rede de especialistas na organização é estabelecida e um mecanismo de apoio à troca de informações entre os especialistas e os projetos é implementada;
GRH11	O conhecimento é disponibilizado e compartilhado na organização.

Tabela 7.6: Processo GERÊNCIA DE RECURSOS HUMANOS [SOFTEX, 2012a]

Máximo está contente, seus afilhados estão ficando melhores do que o professor.

O QUE ACONTECEU ATÉ AGORA
38. Tahini-Tahini chega ao nível F do modelo MPS-SW.
39. Marcela e os gêmeos analisam as causas-raiz das dificuldades para contratar pessoal idôneo e fazê-los render no curto prazo.
40. Marcela e os gêmeos começam a construção de um modelo de competências para Tahini-Tahini.
41. Marcela propõe um mecanismo de seleção proativo para a contratação de pessoal.
42. Máximo revisa e aprova a implementação escolhida para o processo Gerência de Recursos Humanos.

### 7.3 A Necessidade de Ativos de Processo

Ao fazer a análise das necessidades de capacitação, é frequente descobrir que estão faltando alguns dos recursos necessários para ajudar o pessoal na realização de suas tarefas: planos, ferramentas de *software*, diretrizes de apoio, formulários tipo, padrões de trabalho, evidência histórica e numérica do desempenho dos processos, materiais de treinamento e de apoio aos usuários dos próprios produtos, entre outros. Todos estes auxiliam nas decisões e facilitam a realização das tarefas cotidianas. Sua ausência representa um mal investimento, pois o conhecimento não capturado deve ser reproduzido, com os consequentes riscos, ou compartilhado pelos que já possuem, com as consequentes demoras.

A capacitação centralizada é só uma parte da solução: a organização que aprende, faz isso constantemente, por designio de sua estrutura. É preciso criar uma estratégia para gerar, captar, difundir e aproveitar o conhecimento em cada um dos papéis e funções da organização, uma corrente de grupos de interesse em cada disciplina e apoiá-la com múltiplos meios de difusão e compartilhar conhecimento e habilidades entre pares, como *blogs* internos, fóruns internos, *webs* para armazenar conhecimento, reforço das atitudes de compartilhar com os que sabem menos, etc. Marcela escolhe uma ferramenta de gestão de conhecimento entre várias opções<sup>67</sup> e, com a ajuda de Armando, a implementam na organização. De imediato, é produzida uma corrente de interesse que começa a gerar artefatos que, pouco a pouco, vão sendo agregados àqueles ativos que já descrevemos, armazenados na rede interna da T<sup>2</sup>.

Com o crescimento organizacional, uma empresa que alcançou isto por estar relativamente bem organizada, pode descobrir que, na realidade, essa evolução a fez retroceder, empurrada pelo influxo de pessoal novo e a multiplicação do número de projetos, até parar em um mundo de processos caóticos. A solução é simples, mas demanda organização para que possa ser aplicada e ter paciência para que, com o tempo, dê seus frutos: trata-se de definir processos organizacionais adaptáveis, adotáveis e com ativos fáceis de usar, mesmo quando os

<sup>67</sup> <http://bsix12.com/knowledge-sharing-tools/> é um bom ponto para começar a busca.

processos devem ser detalhados para que sua utilização (ou a falta dela) possa ser facilmente avaliada em cada projeto individual, segundo suas características. Um processo muito flexível não é um processo avaliável. Já vimos no Capítulo 5 quais atributos descrevem um processo, com a planilha que introduziu Máximo. Sem trocar essa estrutura, o que se pede é que os passos para a sua execução sejam muito concretos e que seja possível avaliar o seu seguimento ou não.

Marcela começa a construir uma arquitetura para organizar os ativos que vão sendo gerados, assim como a construir aqueles que não só lhe deem estrutura aos que surgem espontaneamente das equipes de projetos, mas sobretudo incorporando outros que inspirem a continuar crescendo. Por exemplo, ela está segura de que *Kanban* e *Scrum* não serão as únicas fontes de inspiração para a gestão de projetos, por isso decide que em “sua” biblioteca de processos sempre haverá lugar para definir ciclos de vida em uso na T<sup>2</sup> e torná-los adaptáveis aos processos. Em especial, está pensando em algumas ideias que discute com Ana, que continua trabalhando de sua casa, desde as últimas semanas de sua gravidez, enquanto cuida do recém-nascido, e que têm a ver com *Feature Driven Development*<sup>68</sup>.

Por causa da sua experiência com o recrutamento de novos colaboradores e os objetivos fixados para controlar o desempenho, sobretudo como medida da efetividade dos treinamentos, Marcela está segura da utilidade de manter um repositório de dados de desempenho dos processos. A planilha de definição de processos já incorpora medidas, mas Marcela vai acrescentando indicadores de desempenho, individuais e por equipes, que servem para tomar decisões no momento de fazer ofertas de trabalho com conhecimento da capacidade para realizá-lo. Para evitar que as medições não tenham nenhum sentido, Marcela propõe que somente sob exceções justificadas e aprovadas pela direção (da qual é parte) podem ser utilizados procedimentos que não tenham sido sancionados previamente. Desse modo, e como os processos são avaliáveis em relação a seu seguimento<sup>69</sup>, os resultados podem ser comparados. Se os caminhos para executar os processos fossem individuais, a comparação de resultados não seria possível.

A “aprovação” de um processo, segundo pensa Marcela, é efetivada pela publicação na Biblioteca de Ativos de Processos, BiPro como é chamada, na *intranet*. Como parte das medidas para acelerar o rendimento do pessoal recentemente incorporado, Marcela detectou que é necessário definir padrões dos ambientes de trabalho, pois a compra de novos equipamentos e licenças não é um assunto tão imediato quanto gostaria. Para poder conseguir bons preços no mercado é necessário contar com certa antecipação para a compra. Por outro lado, a experiência mostra que a incorporação de pessoal com experiência em outros tipos de projetos ágeis é difícil, por isso, receber um padrão de capacitação é bastante útil. Marcela opta por um mecanismo que faz a incorporação menos traumática: cada novo colaborador, de agora em diante, contará com o apoio de outro, já experiente, que o ajudará em seus primeiros passos: cada um dos cinco primeiros dias na T<sup>2</sup> estão pautados na Diretriz de Incorporações, a apresentação à direção, o percurso das instalações, o uso da BiPro, os cursos *online* e o ambiente de trabalho.

A BiPro contém muitos elementos. Marcela foi acumulando experiências próprias e alheias, filtrando os melhores ativos da ferramenta de gestão de conhecimento e a lista do conteúdo é longa: em princípio contém o Plano Organizacional de Melhoria de Processos, que discutiremos em detalhe na seção a seguir. Este plano permite a qualquer um que esteja interessado em conhecer as necessidades de melhorias de processos que foram reconhecidas pela organização e como esta se propõe a atendê-las, identificando as estratégias, enfoques e ações para realizar melhorias de processos.

No nível mais alto, há uma descrição da arquitetura dos processos organizacionais, que se baseia no Processo Padrão da T<sup>2</sup>, com sua Diretriz de Adaptação para seu uso nos projetos. Marcela já incorporou algumas Políticas Organizacionais, para comunicar os princípios que guiam a organização, e Descrições Organizacionais de Ciclos de Vida, que permitem guiar a determinação das principais fases de um projeto ou produto sobre as quais é possível determinar o escopo do planejamento.

---

<sup>68</sup> Ver Capítulo 3

<sup>69</sup> Ver no Capítulo 6 Garantia da Qualidade

Também há um Exemplo do Plano de Projeto, que ilustra o nível adequado das estimativas, definição de papéis e outros atributos críticos relacionados ao planejamento de um projeto, e Definições Operacionais de Medidas para fornecer descrições precisas e inequívocas dos atributos das entidades que podem ser úteis medir em um processo. Para auxiliar na padronização dos processos, há planilhas e diretrizes de uso, como as Planilhas para Procedimentos e Critérios de Testes de Validação e Verificação, e Normas de Desenvolvimento de Produto. Em geral, há claras definições das expectativas da organização para processos, ferramentas, técnicas e métodos a serem usados no ambiente de desenvolvimento. Há também outras diretrizes particulares, como as Diretrizes de Análise de Decisão para guiar a seleção de temas que devem ser submetidos a avaliações formais. Ilustrar vários atributos que deveriam ser considerados na avaliação do *design* de componentes de produto.

Existem, da mesma forma, materiais para ajudar na seleção e uso de técnicas e ferramentas, como os Exemplos de Técnicas de Levantamento de Requisitos para fornecer apoio com exemplos de técnicas para a coleta de necessidades, expectativas, restrições e interfaces dos interessados, o mesmo com a planilha para especificações de requisitos que ajuda a comunicar efetivamente os compromissos e produtos do projeto a todos os interessados mais relevantes. Há outros, mas, como exemplo, estes são suficientes. O critério é que se é útil para alguém, pode ser guardado para ser acessado e o conhecimento, reusado. Cada elemento tem seu guia de uso e os critérios associados para sua escolha diante de várias alternativas.

Como algo tão diverso e amplo é difícil de utilizar, uma parte corresponde aos ativos recomendados necessários para o uso cotidiano, mas a BiPro tem capacidades de *wiki* auto-organizada baseada em um algoritmo de redes neurais para reuso do conhecimento desenvolvido por um dos novos programadores em seu trabalho de graduação. Desenvolvidos, inicialmente, para o reuso de *software*<sup>70</sup>, os *wikis* semânticos, como são chamados, simplificam o armazenamento da informação ao utilizar algoritmos que relacionam os conteúdos de maneira automática, incrementando a sua usabilidade. Um problema conhecido dos *wikis* é que, no início, quando há pouco conteúdo, a usabilidade é muito alta. Ao aumentar o conteúdo, a usabilidade diminui rapidamente. Por isso, o uso de auto-organização diminui o esforço e permite que uma empresa pequena como a T<sup>2</sup> possa armazenar, com facilidade, o aprendizado gerado a partir da ferramenta de gestão do conhecimento.

Marcela mantém um indicador de uso dos elementos da BiPro. Pelas ferramentas de apoio escolhidas, é sempre mais fácil acessar o documento ou conhecimento na biblioteca do que guardar cópias em pastas locais e acessá-las quando forem usadas. Isto faz com que as estatísticas de uso sejam representativas da realidade. Utilizando estes dados, Marcela vai entendendo que há perfis de projetos, embora às vezes haja perfis de *sprints*. Decidida a entender, armam com Ana um grupo de trabalho que inclui Mariano e os gêmeos, e começam um trabalho de detetive, juntando dados dos atributos dos *sprints* e correlacionando estes dados com o uso de certos elementos na prática. Analisando estes dados, chegam às seguintes conclusões, que publicam na BiPro para sua difusão e uso.

<b>Orientação Sugerida por Perfil de <i>Sprint</i></b>	
<b>condição</b>	<b>escolha sugerida</b>
o tamanho da funcionalidade excede a capacidade do <i>sprint</i>	<i>Kanban</i> com uma mini equipe dedicada
a funcionalidade é nova e tem muitas arestas inexploradas	mini espiral de <i>Boehm</i> dentro do <i>sprint</i> ou dividir o <i>sprint</i> a partir de uma análise para fazer demos mais frequentes.
a funcionalidade é totalmente central ao funcionamento do produto (controle)	usar TDD e inspeções, além de programação por pares.
o dono do produto tem dúvidas sobre a funcionalidade que sugere	ampliar o jogo do planejamento até incluir um esboço de modelo dinâmico usando técnicas de UML e FDD
	diminuir a duração do <i>sprint</i> para fazer demos mais

<sup>70</sup> DECKER, B., et al, 2005.

	frequentes
a equipe não tem experiência	usar inspeções e programação por pares com o <i>scrum master</i> em um dos papéis
a arquitetura é crucial para o desempenho futuro do produto	usar um <i>sprint</i> para definir a arquitetura, à maneira do FDD, e decidir depois sobre <i>Kanban</i> , <i>Scrum</i> ou <i>Scrumban</i>

Tabela 7.7: Orientação Sugerida por Perfil de *Sprint*

Quatro semanas depois de contar com a presença do consultor no escritório da T<sup>2</sup>, volta-se a ligar para Máximo para a revisão do processo Definição do Processo Organizacional, com resultados perfeitos.

<b>Definição do Processo Organizacional (DFP)</b>	
<b>DFP1</b>	<b>Um conjunto definido de processos padrão é estabelecido e mantido, em conjunto com a indicação da aplicabilidade de cada processo;</b>
<b>DFP2</b>	<b>Uma biblioteca de ativos de processo organizacional é estabelecida e mantida;</b>
<b>DFP3</b>	<b>Tarefas, atividades, papéis e produtos de trabalho associados aos processos padrão são identificados e detalhados, juntamente com o desempenho esperado do processo;</b>
<b>DFP4</b>	<b>As descrições dos modelos de ciclo de vida a serem utilizados nos projetos da organização são estabelecidas e mantidas;</b>
<b>DFP5</b>	<b>Uma estratégia para adaptação do processo padrão é desenvolvida considerando as necessidades dos projetos;</b>
<b>DFP6</b>	<b>O repositório de medidas da organização é estabelecido e mantido;</b>
<b>DFP7</b>	<b>Os ambientes padrão de trabalho da organização são estabelecidos e mantidos;</b>
<b>DFP8</b>	<b>Regras e diretrizes para a estruturação, formação e atuação de equipes são estabelecidas e mantidas.</b>

Tabela 7.8: Processo DEFINIÇÃO DO PROCESSO ORGANIZACIONAL [SOFTEX, 2012a]

O QUE ACONTECEU ATÉ AGORA
43. Marcela incorpora uma ferramenta para compartilhar conhecimento entre os membros da Tahini-Tahini.
44. Marcela define uma arquitetura para organizar os ativos de processo.
45. Marcela define indicadores a partir de uma base de dados de desempenho dos processos que permitem melhorar o conhecimento da capacidade individual e das equipes.
46. Marcela define padrões dos ambientes de trabalho.
47. Marcela define um processo e ativos para a incorporação de pessoal novo às equipes.
48. Marcela conecta a ferramenta de gestão do conhecimento à biblioteca de ativos de processo, utilizando um algoritmo que permite auto-organizar os conteúdos.
49. Um <i>workshop</i> interno identifica correlações entre usos de ativos da biblioteca e atributos dos projetos e produtos a construir e gera uma diretriz de uso.
50. Máximo revisa os processos e ativos de Definição do Processo Organizacional e os aprova.

#### 7.4 Retrospectivas e processos

Assim como o crescimento do pessoal e a multiplicação dos projetos pode, como já foi dito, acabar em um mundo de processos caóticos, a aceleração da mudança pode levar a confundir mudanças de processos com melhoria de processos, perdendo tempo e dinheiro. Marcela tem

um espírito prático com formação contábil e de sistemas, uma combinação que a faz ser muito esperta. Para evitar que os esforços na área de processos sejam dispersos adotou uma estratégia que lhe permite entender e documentar os objetivos estratégicos de processo da T<sup>2</sup>, o que acontece durante as reuniões mensais de revisão do portfólio. A revisão permanente desses objetivos é feita a partir do sistema de medição da capacidade dos processos que Marcela coloca em prática.

Com estes dados, Marcela mantém seu plano de melhoria baseado na medição da capacidade e dos objetivos estratégicos de negócio. O plano parte de atividades que realizam os projetos, em particular as autoavaliações que são realizadas ao final de cada *sprint*, chamadas retrospectivas<sup>71</sup>, onde são discutidas tanto novas necessidades quanto as melhorias já realizadas, para selecionar e adotar melhorias aprovadas e que serão difundidas e implantadas nos outros projetos. Este é mais um dos planos de projeto e seu avanço é discutido nas reuniões mensais, pois os planos de projeto têm os mesmos atributos e a mesma necessidade de monitoramento e controle. A diferença com os projetos de desenvolvimento é que, para cada iniciativa, é escolhido o grupo de especialistas que irá participar na confecção dos ativos, sejam estes processos, procedimentos ou diretrizes, ou todos os anteriores, e é definida a sua dedicação, geralmente de tempo parcial. Marcela conduz estes grupos, no máximo dois por mês, seguindo seu próprio painel *kanban* e com reuniões semanais. Uma boa prática que foi adotada é a de fazer com que todos os especialistas trabalhem ao mesmo tempo no mesmo dia da semana, para obter resultados mais rápidos e assegurar sua participação. A designação é por múltiplos de 10% de dedicação semanal, de modo que se considerarmos a semana de quarenta horas, cada múltiplo é de meio dia (um turno). Desse modo, é possível controlar os compromissos com o projeto.

A inclinação de Marcela pelos números a leva a ser sistemática ao medir os resultados obtidos comparando com os resultados esperados do plano de melhorias, que documenta em cada caso e apresenta nas reuniões mensais de revisão do portfólio. O planejamento estrito não impede que incorpore melhorias oportunas quando existam, como as ideias que Ana traz de suas aulas. Marcela revisa por sua conta os resultados do processo Avaliação e Melhoria do Processo Organizacional e, satisfeita com os resultados, envia um relatório à Gerência Geral recomendando começar os preparativos para uma avaliação de nível E do MR-MPS-SW.

#### **Avaliação e Melhoria do Processo Organizacional (AMP)**

- AMP1** A descrição das necessidades e os objetivos dos processos da organização são estabelecidos e mantidos;
- AMP2** As informações e os dados relacionados ao uso dos processos padrão para projetos específicos existem e são mantidos;
- AMP3** Avaliações dos processos padrão de organização são realizadas para identificar seus pontos fortes, pontos fracos e oportunidades de melhoria;
- AMP4** Registros das avaliações realizadas são mantidos acessíveis;
- AMP5** Os objetivos de melhoria dos processos são identificados e priorizados;
- AMP6** Um plano de implementação de melhorias nos processos é definido e executado, e os efeitos desta implementação são monitorados e confirmados com base nos objetivos de melhoria;
- AMP7** Ativos de processo organizacional são implantados na organização;
- AMP8** Os processos padrão da organização são utilizados em projetos a serem iniciados e, se pertinente, em projetos em andamento;
- AMP9** A implementação dos processos padrão da organização e o uso dos ativos de processo organizacional nos projetos são monitorados;
- AMP10** Experiências relacionadas aos processos são incorporados aos ativos de processo organizacional.

Tabela 7.9: Processo AVALIAÇÃO E MELHORIA DO PROCESSO ORGANIZACIONAL [SOFTEX, 2012a]

<sup>71</sup> Ver Capítulo 3

**O QUE ACONTECEU ATÉ AGORA**

51. Marcela define um plano de melhoria contínua de processos baseado nos objetivos organizacionais de negócios e na capacidade real dos processos.
52. Marcela revisa por sua conta os resultados do processo Avaliação e Melhoria de Processos Organizacionais

**7.5 Diminuição de custos por reuso de componentes**

Mas antes de alcançar o nível E, a T<sup>2</sup> deve atender alguns outros processos. Apesar dos esforços de contratação de pessoal, organização e difusão de conhecimento e processos que aceleram as atividades, as equipes ficam aquém da produtividade exigida. Ana, já reincorporada parcialmente (participa de todas as reuniões de direção e quando é convidada a revisar artefatos em conjunto com a equipe) mostra que, como já está sendo usado o algoritmo que um de seus alunos aplicou no trabalho de graduação para a auto-organização do conhecimento nos *wikis*, este mesmo algoritmo poderia ser usado para encontrar os componentes úteis na biblioteca dos projetos. O que ela sugere é implementar uma estratégia de reuso que contemple critérios claros para a seleção dos componentes, com mecanismos de armazenamento e uso dos ativos, além de procedimentos para que sejam atualizados permanentemente com usuários claramente identificados do sistema, para comunicá-los sobre mudanças e melhorias. Os procedimentos precisam atender tanto os aspectos técnicos quanto os administrativos. Entende-se como artefato ativo reutilizável qualquer *software* que seja preparado, quer dizer, que seja empacotado expressamente para ser reutilizado pelos processos da organização. Isto sugere que existam alguns critérios especiais para sua construção e teste. A primeira consideração é o ajuste arquitetônico. Os componentes se agrupam por tecnologia e interfaces, e sua versão é atualizada constantemente. As APIs são publicadas em um relatório, que é indexado para ser utilizado mais adiante. Isto evita que seja necessário fazer um processo manual de busca, porque os filtros são automáticos, o que coloca à disposição da equipe um conjunto de oportunidades para explorar justo quando estão buscando alternativas para integrar seu plano do *sprint*.

Outros critérios fundamentais são: o custo versus o benefício de reutilizar; a adequação ao plano e ao *design* do produto; se é ou não um produto herdado; e outros critérios que podem ser dependentes do projeto em si. Como os projetos são ágeis, é a equipe que decide os critérios.

Decide-se que os projetos incorporem a seu procedimento de planejamento, no momento de estabelecer o *Backlog* do *sprint*, uma análise de opções que siga os passos do processo de reuso. Brevemente, os passos são os seguintes:

1. Identificação de objetivos, isto é, os motivos pelos quais se busca reutilizar componentes. Alguns exemplos são diminuir prazos sem perda de qualidade, permitir manter a duração do *sprint* desenvolvendo mais pontos de usuário, diminuir custos, etc.
2. Escolha de atributos desejáveis, onde a equipe discute se vale a pena, ou não, seguir o processo de reuso, a partir dos requisitos, do *design* a adotar ou pré-existente, da história com reuso da equipe, do volume de trabalho no *sprint*, e/ou qualquer outro adequado ao projeto.
3. Identificação de candidatos, que se realiza automaticamente usando o algoritmo neural baseado nos atributos escolhidos.
4. Avaliação de candidatos, realizada por testes *ad-hoc*, que são projetados de acordo com os atributos escolhidos e a história do componente.
5. Análise de opções, realizada seguindo um método pré-estabelecido que utiliza uma matriz de Pugh como a que se viu no Capítulo 4. Uma das opções sempre é não reutilizar um componente.
6. Seleção de alternativas, também seguindo o método anterior.
7. Adoção de componentes, quando aplicado, é realizado o ajuste do componente, de acordo com as necessidades da equipe. Pode ser que, chegando neste ponto, o resultado da avaliação dê alternativas com resultado negativo e se decida não usar nenhum componente.

8. Avaliação do resultado, na qual compara-se o resultado com os objetivos identificados no primeiro passo para continuar montando a história do componente e acrescentando os conhecimentos adquiridos.

**Figura 7.2: Análise de Opções sobre Reuso**

Da mesma forma, para que um componente possa integrar a biblioteca de ativos de reuso, é preciso que, primeiro, seja proposto como tal pelos seus autores, antes mesmo de construí-lo, porque o desenvolvimento de um componente para reuso é diferente do desenvolvimento normal de código. Os critérios exigem que se acrescente uma documentação especial para guiar o reuso para que, no futuro, seja possível selecionar o componente a partir de características predefinidas, assegurando: que os testes sejam exaustivos e o critério de aceitação seja zero-defeitos; que o *design* e a arquitetura, ou arquiteturas com as quais o componente interage normalmente também formem parte dessa documentação; e que esta seja inspecionada formalmente. Além disso, há um critério especial para a análise estática do código, que requer seguir normas organizacionais<sup>72</sup>, como a indentação<sup>73</sup>, os comentários, os nomes de variáveis e a documentação de mudanças no código. A biblioteca de componentes para reuso tem seu próprio mecanismo de gerência de configuração e sua própria rastreabilidade entre ativos, que é simplesmente um subconjunto com restrições maiores do que as usuais do sistema de gerência de configuração organizacional para *baselines*. Os usuários deste subsistema recebem os relatórios sobre os ativos gerenciados desta forma.

#### O QUE ACONTECEU ATÉ AGORA

53. Ana sugere introduzir reuso de componentes para aumentar a capacidade das equipes.
54. É definido e realizado um processo de criação, adoção e manutenção de componentes de reuso com um procedimento para decidir quando deve ser reutilizado um componente e se estende a diretriz de ajuste do processo para que contenha este critério.

#### Gerência de Reutilização (GRU)

- GRU1** Uma estratégia de gerenciamento de ativos é documentada, contemplando a definição de ativo reutilizável, além dos critérios para aceitação, certificação, classificação, descontinuidade e avaliação de ativos reutilizáveis;
- GRU2** Um mecanismo de armazenamento e recuperação de ativos reutilizáveis é implantado;
- GRU3** Os dados de utilização dos ativos reutilizáveis são registrados;
- GRU4** Os ativos reutilizáveis são periodicamente mantidos, segundo os critérios definidos, e suas modificações são controladas ao longo do seu ciclo de vida;
- GRU5** Os usuários de ativos reutilizáveis são notificados sobre problemas detectados, modificações realizadas, novas versões disponibilizadas e descontinuidade de ativos.

**Tabela 7.10: Processo GERÊNCIA DE REUTILIZAÇÃO [SOFTEX, 2012a]**

### 7.6 Utilizando o conhecimento histórico nos projetos

A partir do nível E, alguns resultados esperados dos processos evoluem e novos resultados são incorporados. A gerência de projetos deve, agora, ser realizada a partir de um processo especialmente definido para o projeto e selecionado com base em ativos da organização. Este processo deve contemplar todos os elementos constitutivos do projeto, desde a construção de planos integrados até o seu encerramento. Marcela tem consciência das vantagens que trazem

<sup>72</sup> Nos métodos ágeis, as equipes escolhem as normas que utilizarão, mas ao incluir um componente na biblioteca de componentes para reuso, é necessário seguir normas da organização.

<sup>73</sup> Algo conseguido facilmente com ferramentas de “*Pretty Printing*”.

estas mudanças, mas seu espírito prático a obriga colocar um marco firme nas necessidades da T<sup>2</sup>.

Algumas diferenças entre a maneira de interpretar e, portanto, de executar certos procedimentos fazem com que Marcela precise reforçar firmemente sua mensagem de uniformidade na execução. Seu argumento é que sem uniformidade não há clareza na interpretação dos resultados e a capacidade real não pode ser conhecida. Da mesma forma, sustenta que a qualidade vem da previsibilidade. Sua tese baseia-se em que qualidade não é nem subjetiva, nem objetiva, é um atributo da relação entre um objeto e o sujeito que avalia a qualidade<sup>74</sup>. Não existe qualidade “absoluta”, sempre é qualidade “para” e em um contexto. Da mesma forma, a qualidade está relacionada às expectativas dos clientes. No modelo de Kano<sup>75</sup> é possível ver que há três tipos de requisitos, que serão descritos a seguir. O primeiro deles, os requisitos Indispensáveis, também são chamados de básicos, porque se não forem satisfeitos, o cliente estará sumamente insatisfeito, como por exemplo, na compra de uma bicicleta que, ao ser entregue, teve suas rodas retiradas porque elas “são compradas separadamente”. Há uma expectativa de que as bicicletas sejam vendidas com suas rodas. Mas, como o cliente percebe que são indispensáveis, fornecê-las não aumenta a sua satisfação. O segundo tipo, os requisitos Lineares, são os mais visíveis, os mais comuns. A satisfação do cliente é proporcional ao grau de sua satisfação, ou seja, quanto mais se atenda a estes requisitos, maior a satisfação do cliente e vice-versa. Por exemplo, a capacidade do porta-malas de um automóvel, ou a quantidade de quilômetros que podem ser feitos sem precisar reabastecer. Com frequência, os clientes exigem explicitamente estes requisitos. Há um terceiro tipo, chamado por Kano de requisitos Atrativos, também chamados “deleites”. São aqueles que o cliente nem espera nem requer explicitamente (surpresas positivas). Atende-los leva a satisfações excepcionais, mas se não são atendidos não há sentimento de insatisfação. Um exemplo pode ser uma cadeirinha para transportar crianças ou uma roda extra ao comprar uma bicicleta. O que se pode concluir a partir do modelo de Kano é que o que se define por qualidade é a percepção que o cliente tem do produto, não os atributos em si. Há atributos esperados, claro, que marcam o mínimo, mas além destes, o contexto é que define se há qualidade suficiente ou não. A consistência nas entregas é uma qualidade evidente do *software* para o cliente. Tanto o tempo, quanto as interfaces ou a densidade de defeitos precisa ser do nível esperado ou o cliente ficará insatisfeito. Esse é o argumento de Marcela a favor da normalização e da consistência.

Portanto, em uma reunião mensal de análise do portfólio de projetos, Marcela levanta a solicitação de discutir vários temas relacionados. Os gêmeos contribuem apontando que as equipes não aprendem de sua experiência e continua existindo muita diferença entre o tempo ideal e o real, o que traz não conformidades em cada *sprint* que acabam sendo significativas para o cliente. Por outro lado, apesar de existir uma norma sobre os recursos a serem utilizados pelos funcionários, nem sempre ela é seguida. Marcela intervém para afirmar que vai contratar uma pessoa para ajudá-la na garantia da qualidade, de maneira permanente, e que, a partir de agora, o processo do projeto deve ser definido como parte do plano antes do projeto ser iniciado. Ana pede que sejam expressos no processo, com toda clareza, a estrutura de decisões, porque viu, em suas visitas espontâneas às equipes de projetos, que as responsabilidades sobre as decisões podem levar a demoras se não são expressas corretamente com antecedência. Os gêmeos voltam a intervir, desta vez para criticar que as retrospectivas não são guardadas com fidelidade: as lições “aprendidas” são somente lições registradas que não se traduzem em experiência para os projetos porque não se usa a estrutura de palavras-chave que permite classificá-las para seu uso posterior. Quando termina a reunião, Marcela, que é a encarregada das minutas da reunião, sai com a lista de ações pendentes.

- a. utilizar o histórico dos *sprints* no jogo do planejamento para ter uma melhor aproximação ao esforço real

---

<sup>74</sup> [PIRSIG, 1974], Zen e a arte da Manutenção de Motocicletas. Pode-se dizer que é um dos ensaios mais importantes e profundos, já escritos, sobre a natureza e o significado de “qualidade” e, definitivamente, um calmante necessário para as consequências de um mundo moderno, patologicamente obcecado com a quantidade.

<sup>75</sup> <http://kanomodel.com/>

- b. fixar o uso<sup>76</sup> do padrão de recursos necessários, nos planos do projeto, para poder começar sua provisão antecipadamente (escritório, PC, SW, etc.)
- c. estabelecer normas de comportamento das equipes e fazer com que sejam respeitadas, com as variações necessárias
- d. dar às lições aprendidas e experiências o lugar importante que merecem e armazenar de forma adequada os resultados para aproveitá-los em futuros projetos
- e. estabelecer processos padrão e mecanismos de escolha entre eles, com critérios para escolher e adaptar.

Marcela sorri. Muitas das ações já foram incorporadas em sua biblioteca, mas agora conta com o mandato necessário para que sejam cumpridas. Confia que logo será materializado o progresso alcançado em uma avaliação formal de Nível E.

#### **Gerência de Projetos (GPR)**

- GPR4 (A partir do nível E) O planejamento e as estimativas das tarefas do projeto são feitos baseados no repositório de estimativas e no conjunto de ativos de processo organizacional;**
- GPR8 (A partir do nível E) Os recursos e o ambiente de trabalho necessários para executar os projetos são planejados a partir dos ambientes padrão de trabalho da organização;**
- GPR20 (A partir do nível E) Equipes envolvidas no projeto são estabelecidas e mantidas a partir das regras e diretrizes para estruturação, formação e atuação;**
- GPR21 (A partir do nível E) Experiências relacionadas aos processos contribuem para os ativos de processo organizacional;**
- GPR22 (A partir do nível E) Um processo definido para o projeto é estabelecido de acordo com a estratégia para adaptação do processo da organização.**

Tabela 7.11: Processo GERÊNCIA DE PROJETOS (a partir do nível E) [SOFTEX, 2012a]

#### **O QUE ACONTECEU ATÉ AGORA**

55. Na reunião mensal de análise de portfólio de projetos, são apresentados os problemas de desempenho que continuam diminuindo a capacidade da T<sup>2</sup>, como pessoal sem suficiente competência, equipes que não aproveitam a história ou experiências anteriores, indefinição de papéis nas equipes, etc.
56. Marcela fica encarregada de transformar a lista de ações em realidade, parcialmente implementadas na BiPro, mas ainda não realizadas nos projetos.
57. A T<sup>2</sup> se encaminha para uma avaliação do Nível E.

<sup>76</sup> Pouco vale ter uma diretriz se não é seguida!

## CAPÍTULO 8 - ELIMINANDO OS DEFEITOS

### 8.1 Nível D do MPS-SW

Mesmo antes da publicação do resultado da avaliação do Nível E, Marcela está agindo para fazer com que os resultados sejam os mais firmes possíveis, segundo seu plano. Depois de várias entrevistas de trabalho com potenciais colaboradores, Marcela encontra uma pessoa que possui as condições que está buscando: um passado com experiência administrativa e contábil, uma licenciatura em Psicologia Aplicada à Empresa e um Diploma em *Design Instrucional*. Jéssica, este é seu nome, terá como responsabilidade completar e melhorar o modelo de competências da T<sup>2</sup> e aperfeiçoar a base de conhecimentos e seu uso pelos colaboradores da empresa. Jéssica consegue se integrar perfeitamente com a equipe de direção, sendo, como eles, jovem, entusiasmada e com muito boa formação. Não demora muito para que seja convidada a fazer parte das reuniões mensais da direção.

### 8.2 Determinando o Problema

O primeiro assunto do dia nessas reuniões, desde que a BiPro e a base de dados estão sendo usadas na empresa, é rever os números. Mariano prepara um relatório sobre os dados de satisfação dos clientes e a correlação com componentes do produto. Isto é colocado como um indicador de tendência com os dados que são refletidos como séries temporais no gráfico: Densidade de Defeitos Identificados pelo Cliente por Componente, que é uma quantidade que muda mês a mês, ao se modificar o tamanho do componente e os defeitos identificados; e Porcentagem de Defeitos Identificados pelo Cliente relacionados com o componente. O primeiro indica a quantidade de defeitos introduzidos e não detectados antes de liberar o produto para o cliente e o segundo, qual dos componentes é o maior responsável pelos defeitos. A derivada da primeira série é o indicador de melhoria ou de piora, enquanto que o histograma indica onde buscar os problemas. Já a série temporal do módulo com pior desempenho indica a magnitude.

Desse modo, é possível criar a estratégia para identificar aqueles defeitos que valem a pena analisar na reunião. Se a variação desde o mês anterior é muito grande, Marcela se responsabiliza pela realização de uma análise mais profunda da causa-raiz com participação dos especialistas no tema. Se o componente responsável tiver muitos defeitos, é possível prever que será refeito.

A partir da análise dos defeitos encontrados pelo cliente e do grau de satisfação correspondente, é feita a iteração sobre um processo que Jéssica introduziu ao grupo, conhecido como a folha de '*Balanced Score Card*' (BSC)<sup>77</sup>, de uso comum em empresas de porte médio e grande. É utilizada para assegurar que a organização entenda sua missão, alinhar seus objetivos com a missão e canalizar adequadamente os recursos e as energias para atingir seus objetivos. O método consiste em identificar primeiro a visão de negócios, o 'para onde queremos ir', para poder encontrar significados nas atividades. Depois de esclarecer a visão, que no caso de um exercício tão frequente como o que é realizado na T<sup>2</sup> é simplesmente a verificação de cada oportunidade<sup>78</sup>, são analisados os resultados, fixados objetivos e discutidos os investimentos em diferentes categorias. A T<sup>2</sup> adotou como eixos de sua análise as categorias clássicas: 'Resultados Financeiros', 'Satisfação do Cliente', 'Processos Internos' e 'Conhecimento e Crescimento do Pessoal'. A Figura 8.1 resume os passos de sua construção.

Para cada uma das categorias listadas são escolhidos "temas" que estão alinhados com a visão de negócios da empresa. Por exemplo, para que os resultados financeiros estejam alinhados com a visão de crescimento há, pelo menos, quatro coisas que podem ser feitas: aumentar as vendas com os clientes atuais, (1) vendendo-lhes novos produtos ou (2) estendendo as prestações, (3) aumentando o portfólio de clientes ou (4) subindo os preços. Por outro lado, para analisar como estas metas financeiras são traduzidas em metas com os clientes, na T<sup>2</sup> são usados três temas para (1) expandir (que eles comprem mais licenças), (2) aprofundar (que nos solicitem novas prestações) ou (3) redefinir relações com os clientes (que

<sup>77</sup> [KAPLAN e NORTON, 1996], *The Balanced Scorecard: Translating Strategy into Action*.

<sup>78</sup> A visão se expressou como "Ter um crescimento sustentável de pelo menos 15% anual com aumento das margens proporcionais ou maior a cada ano".

forneçam melhores referências da empresa e estas se transformem em vias de vendas). Em todo caso, trata-se sempre de aumentar a satisfação dos clientes com os produtos.

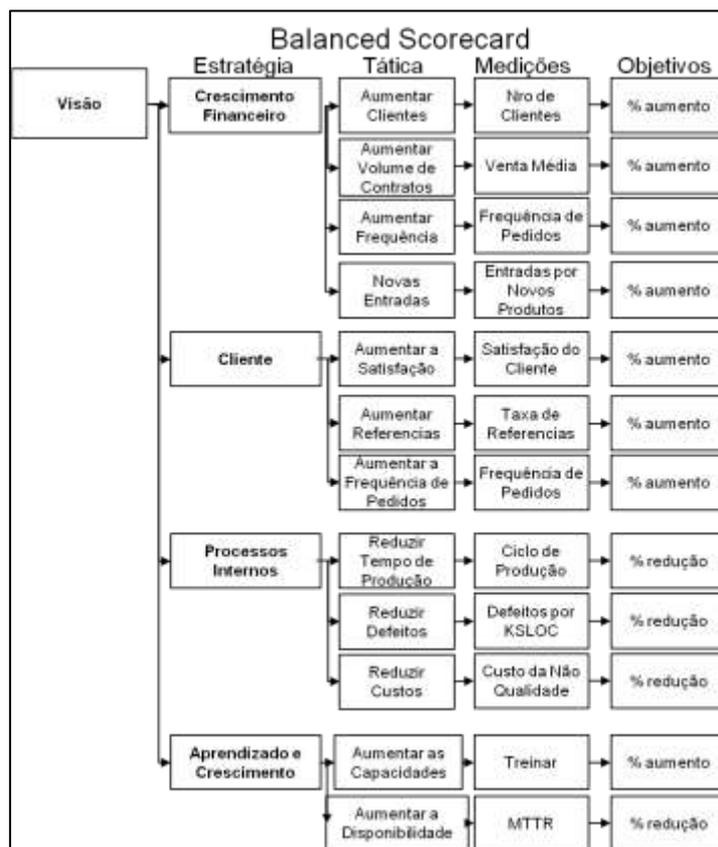


Figura 8.1: Estrutura Típica de uma Folha de Resultados Equilibrados

É evidente que se os objetivos de gestão com o cliente são cumpridos, eles têm um impacto claro nas metas financeiras, mas a análise não termina aí. A pergunta que se segue é: O que precisamos fazer para que estas metas de bom relacionamento com os clientes possam ser alcançadas, do ponto de vista de modificar o que já estamos fazendo? Claramente, os processos que utilizamos impactam na nossa capacidade. As duas variáveis que mais impactam nos clientes são o prazo de entrega e o número de defeitos que entregamos. É necessário diminuir ambos para aumentar significativamente a satisfação, a percepção do cliente. Se ao realizar ações relacionadas a estes objetivos, podemos alcançá-los diminuindo também os custos, melhor, porque ainda se as vendas não sobem muito, ao baixar os custos, aumentam as margens. A análise fica completa com o diagnóstico do aprendizado necessário para que os processos permitam reduzir prazos e defeitos para que os clientes tenham uma maior satisfação e para que os resultados financeiros melhorem. Como todo processo, isso requer que se ajuste de maneira constante o conjunto, porque o aprendizado, por exemplo, pode ser muito caro ou muito longo para que os resultados possam ser traduzidos em objetivos que tenham um prazo razoável.

Os dados coletados permitem, também, realizar uma análise objetiva dos resultados. Comparando a densidade de defeitos produzida pelas equipes da T<sup>2</sup> com as médias históricas na literatura<sup>79</sup>, a direção da T<sup>2</sup> está preocupada com os resultados: quer diminuir o número de defeitos. Um dos problemas detectados pelas retrospectivas é que, apesar da BiPro, há pouca preparação técnica. A BiPro ajuda muito quando a tarefa é de gestão ou de apoio, mas as técnicas de engenharia de *software*, apesar de que os engenheiros começaram a se agrupar em disciplinas, ainda são primitivas. A programação não é o problema, porque a Escola de Engenharia (como tantas outras boas universidades) se encarrega muito bem de formar bons programadores, mas as disciplinas correlacionadas, como a definição e análise de requisitos, o

<sup>79</sup> Capers Jones, [JONES, 1986], *Programming Productivity*

*design* em todos os níveis e a engenharia de testes deixam muito a desejar. A falta de conhecimento nessas disciplinas é notória.

#### O QUE ACONTECEU ATÉ AGORA

58. Jéssica é incorporada à equipe de Marcela e introduz o BSC<sup>80</sup> nas reuniões mensais.
59. Em uma análise equilibrada dos defeitos, detecta-se grande densidade de defeitos no produto, o que é um obstáculo para o alcance dos objetivos financeiros.
60. O uso do material gerado nas retrospectivas das equipes permite identificar uma série de problemas relacionados.

### 8.3 Busca da Solução

A solução poderia ser definir um método próprio e adaptado às necessidades de negócios específicas da T<sup>2</sup>; mas, tanto Ana quanto Marcela são partidárias de uma resposta que lhes permita crescer em número de funcionários selecionando pessoal já pré-capacitado na parte técnica. Portanto, essa solução é descartada. Consultados, os gêmeos sem nem pensar sugerem a capacitação integral em XP: programação entre pares, desenvolvimento dirigido por testes (*Test driven development* ou *TDD*), *design* incremental, integração contínua, propriedade coletiva do código, conhecimento compartilhado, padrões de codificação, passo sustentável e trabalho energizado, além das práticas comuns a *Scrum*, como os *sprint* e o jogo do planejamento. Vocês podem imaginá-los falando ao mesmo tempo e terminando as frases um do outro. Nessas partes onde diferem *Scrum* e XP, como a participação do usuário na equipe, obrigada nesta última e descartada em *Scrum*, os gêmeos preferem manter as técnicas do primeiro. Para Ana, XP é uma possibilidade, mas não a única. Propõe identificar outros métodos e compará-los, seguindo o esquema de análise que já é utilizado normalmente na empresa.

Mesmo quando, a princípio, está de acordo, a esta altura de sua trajetória, parece necessário para Marcela entender o que anda mal e o que quer corrigir antes de começar a decidir. Para isso conta com uma fonte inestimável de experiências: a base de conhecimento construída a partir das reuniões de retrospectiva. Daí surgem múltiplos temas que parecem às equipes problemas ou riscos dos projetos, relacionados com suas tarefas técnicas. Marcela convence os quatro a realizar uma análise desses temas.

Entre ela, Ana e os gêmeos vão lendo os problemas detectados e copiando-os em uma folha autoadesiva. Se eles encontram dois problemas com enunciados muito parecidos colam um sobre o outro. Quando todos os problemas foram reunidos na lousa, os participantes tentam agrupá-los em tipos de semelhança. Por exemplo, o enunciado 'a falta de cobertura na definição de critérios nos levou a tomar decisões equivocadas sobre a qualidade do produto demonstrada pela densidade de defeitos em testes' é agrupado com 'nem sempre a densidade de defeitos que calculamos está baseada em dados confiáveis' e ao fazer isso coloca ambos sob o título 'Problemas em Testes'. Ao finalizar ficaram cinco tipos. O primeiro é intitulado 'Problemas de Requisitos' e são listados os problemas com a mitigação ou solução à direita na Tabela 8.1<sup>81</sup>.

	risco ou problema:	mitigação:
1	os clientes nos dão informações incompletas sobre as necessidades e requisitos do produto, gerando muito retrabalho	seguir um processo sistemático para a identificação do que o cliente necessita para obter o que quer segundo o que diz

<sup>80</sup> *Balanced Score Card*

<sup>81</sup> As outras quatro são denominadas 'Problemas de *Design*', 'Problemas de Integração', 'Problemas de Verificação' e 'Problemas de Validação'. No contexto do modelo MPS-SW, a diferença entre verificação e validação é clara: a verificação é relativa aos requisitos, 'verifica-se' quando se compara um produto, final ou intermediário, com os requisitos dos quais deriva, levando em conta sua completude em relação a eles e sua consistência intrínseca.

2	a especificação de um módulo, componente ou sistema é às vezes incompleta, resultando em um produto abaixo da perfeição	construir um documento que permita priorizar os requisitos e especificá-los de forma a possibilitar sua revisão e análise.
3	costumam ficar funções sem especificar e os requisitos não funcionais são frequentemente acrescentados no final do projeto, resultando em muito retrabalho	a especificação dos requisitos deve permitir identificar funções, bem como os atributos de qualidade não funcionais do produto
4	há muito impacto nos compromissos pela quantidade de requisitos derivados que surgem depois do jogo de planejamento	antes de iniciar um <i>sprint</i> , devem ser definidos os detalhes do conjunto de funcionalidades que será incluído
5	a integração contínua fracassa porque não há uma clara definição das interfaces	parte da especificação deve ser dirigida para o ambiente de funcionamento do produto, sejam interfaces internas ou externas
6	as funções solicitadas são mal interpretadas porque não há um contexto onde situá-las, resultando em demonstrações fracassadas	como parte do mecanismo de entendimento dos requisitos, é necessário construir histórias de usuário, narrativas e/ou casos de uso que expliquem o uso esperado do produto
7	há certa pressa em passar à estimativa sem fazer uma análise mais profunda da funcionalidade que é implementada, resultando em oportunidades perdidas e ineficiências	a seleção de processos em um <i>sprint</i> deve estar baseada na análise de requisitos e necessidades do cliente equilibradas com as restrições de capacidade da equipe
8	Mesmo nos casos em que são construídos modelos do produto, estes poucas vezes são discutidos com o dono do produto	durante o jogo de planejamento, os requisitos menos comuns devem ser validados usando modelos e os casos de uso ou histórias de usuário

**Tabela 8.1: Problemas de Requisitos**

Este é o ponto de partida da futura tabela de decisão. A coluna da direita, 'mitigação:', dá a definição de atributos desejáveis da solução. Quando se compara com as práticas de eXtreme Programming fica claro que não há uma correspondência, salvo com a presença de tempo completo do usuário no meio da equipe, que poderia ajudar, mas que também poderia tornar as coisas mais difíceis: Um usuário que disponha de 8 horas diárias para trabalhar no desenvolvimento de *software* possivelmente não tem muito poder de decisão na empresa que representa, tornando mais complexa a elaboração dos requisitos.

Os gêmeos ficam um pouco frustrados, mas as reuniões de retrospectiva não mentem; tal é a densidade dos problemas derivados da má elaboração de requisitos que finalmente os dois se resignam a expandir o horizonte das práticas aplicáveis pelas equipes de desenvolvimento (sobretudo) e manutenção. Sem renunciar a XP, é necessário utilizar técnicas que tenham foco na primeira parte do processo de construção do produto, a captura e elaboração das especificações técnicas.

#### 8.4 Corrigindo os Processos de Requisitos

Como resposta, Marcela e Ana sugerem voltar a ferramentas que sempre são úteis quando se trata de analisar requisitos, ferramentas conhecidas como 'métodos' de criação de modelos. Os gêmeos não querem acreditar no que estão ouvindo, mas prestam atenção mesmo assim. Em seu mundo, modelar não é tarefa de programadores! Marcela explica as bases da construção de modelos, fazendo uma breve resenha de *A System Modeling Language*, ASML<sup>82</sup>, que depois ficou conhecido como *Yourdon Software Development Methodology* (YSDM). Marcela

<sup>82</sup> [WARD e MELLOR, 1986], *Structured Development for Real-Time Systems, Volume I: Introduction and Tools*

resgata que é mais fácil reconhecer as características a serem implementadas em um sistema se um modelo dele é construído, pois esse modelo funciona como ‘objeto de fronteira’<sup>83</sup> entre o cliente e o engenheiro de *software*. O modelo, então, é passível de análise, tanto pelos usuários quanto pelos técnicos. Uma simples narrativa ou caso de uso pode ser mal-interpretado sem que as diferenças sejam detectadas, porque a linguagem natural é muito ambígua. Um Diagrama de Contexto (ver Figura 8.2), por outro lado, não. É possível fazer um diagrama muito simples que descreva os atores envolvidos no ambiente do sistema e os estímulos (fluxos de entrada) aos que o sistema precisa dar uma resposta automática (fluxos de saída). Esta simples estrutura tem a excelente propriedade de poder ser “executada” com usuários para detectar diferenças de opinião. Como exemplo, Marcela realiza em poucos traços o diagrama da Figura 8.2 e o “narra”, mostrando como o sistema interage com clientes, proprietários e administrador em um sistema de administração de imóveis para alugar ou vender. O Proprietário se registra com seus dados cadastrais e registra seus imóveis. O sistema comunica cada solicitação de registro (“pede autorização para registrar um imóvel”) ao Administrador, que revê e autoriza ou não esse imóvel. Um Cliente se registra interativamente e procura um imóvel no sistema. O sistema recebe as ofertas do Cliente e assim sucessivamente. Os gêmeos não estão convencidos, já não existe no mercado quem trabalhe com ASML ou análise estruturada.



Figura 8.2: Diagrama de Contexto de um Sistema

As mulheres lembram que ASML é o antecessor direto da UML e que UML é ensinado nas Universidades. Ana toma, então, a liderança da reunião e começa a explicar sua experiência nas disciplinas de Arquitetura e *Design* com o que se conhece como FDD, mas enfatizando um de seus componentes, ‘*Java Modeling in Color*’<sup>84</sup> (JMC). Em JMC é utilizada a notação de UML<sup>85</sup>, uma norma de expressão de modelos que evoluiu das velhas linguagens e que continua sendo apoiada por ferramentas e bibliografia, na qual, em JMC, são aplicados abundantemente os padrões, que os autores denominaram ‘arquétipos’, que não devem ser confundidos com os estereótipos de UML.

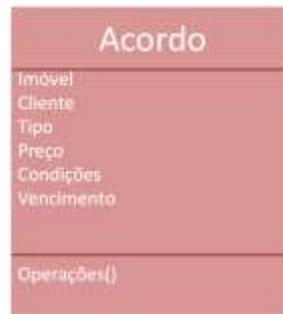
Em JMC, as classes de UML são “pintadas” de diferentes cores para expressar mais claramente sua função. As rosas são chamadas <<intervalo ou evento>> e representam eventos ou atividades para as quais temos que realizar um acompanhamento por razões de negócios ou jurídicas no sistema a ser desenvolvido. Se tomarmos como um caso ilustrativo uma empresa de bens imobiliários, exemplos de classes de cor rosa são Venda e Aluguel. Sempre as classes rosa são as mais importantes, porque se não houvessem transições e

<sup>83</sup> Um “objeto de fronteira” é definido como uma entidade que tem uso plástico, que varia de uma comunidade a outra. Desse modo sua interpretação permite detectar discrepâncias entre as comunidades.

<sup>84</sup> [COAD *et al.*, 1999], *Java Modeling In Color With UML: Enterprise Components and Process*. Atualmente o nome foi modificado para *Visual Paradigm*, abreviado VP UML.

<sup>85</sup> [FOWLER, 2003], *UML Distilled: A Brief Guide to the Standard Object Modeling Language (3rd Edition)*

eventos, não haveria necessidade de história, desta forma não haveria necessidade de sistemas de informação.



**Figura 8.3: Diagrama de Classe de Acordo**

Uma classe pintada de cor verde é uma <<entidade>> e são classificadas, além disso, em <<grupo>> (geralmente uma pessoa ou organização), <<lugar>> (o lugar onde o evento ou atividade é produzida), e <<coisa>> (os objetos do mundo real que participam no evento ou atividade). Exemplos para o nosso caso são Pessoa (que pode ser mais de um) e Imóvel.

As classes amarelas denotam <<papéis>> e representam uma forma de participar em um evento ou atividade por uma classe de entidade verde. Exemplos de classes amarelas são Comprador, Vendedor e AgenteDeVendas.

O quarto arquétipo é o azul, denominado <<descrição-como-entrada-de-catálogo>> (Descrição, para abreviar), e representa a diferença entre algo como um departamento em um edifício de vários andares e a descrição do tipo de apartamento no catálogo de vendas. A descrição é a classe azul, que contém uma série de valores e intervalos de valores que podem ser tomados para todos os apartamentos desse tipo; cada apartamento individual físico está representado por uma classe verde <<entidade>>.

Nestas classes não importa tanto a precisão com a qual se pode instanciar porque seu papel é permitir identificar classes faltantes e ajudar a realizar a análise dinâmica do modelo. As classes que pertencem a uma determinada classe arquétipo têm mais ou menos o mesmo tipo de atributos e operações, mas não precisam ser iguais. Estas classes particulares de arquétipos também tendem a interagir com as classes de outros arquétipos em formas geralmente previsíveis. Estes padrões de características e comportamento podem nos ajudar a construir rapidamente modelos de objetos que se tornam muito sólidos e são facilmente extensíveis, ao identificar rapidamente atributos e operações que de outro modo poderiam ser perdidos, e nos dará maior confiança na estrutura de nosso código.

Notavelmente, um modelo dinâmico, tal como um diagrama de transição de estados, pode ser deduzido da estrutura. Por exemplo, se pegarmos um Acordo de Aluguel, este começaria com uma busca de seu identificador único para acessar o Cliente e, por meio dele, os dados de seu domicílio legal. Esta sequência se repete se o objeto da busca é um Livro emprestado a um Leitor em uma biblioteca, para averiguar onde este mora. Esta característica dos modelos UML coloridos fazem com que a pesquisa seja muito mais sistemática e que até uma pessoa sem conhecimento do domínio seja capaz de conduzir entrevistas produtivas com usuários. Os arquétipos se combinam entre si de maneira natural, que os autores de JMC chamam “componentes independentes do domínio”<sup>86</sup> A Figura 8.4 mostra um componente deste tipo. Como já dissemos, a característica destes componentes independentes é que têm uma dinâmica implícita. Esta dinâmica implícita pode ser traduzida de modo semiautomático em um diagrama de transição de estados que mostra explicitamente as conexões dinâmicas entre as classes para o caso em que, por exemplo, quiséssemos contabilizar transações de um cliente.

A vantagem de contar com uma arquitetura pré-fabricada é evidente para Ana e Marcela, somando a isto o fato de que muitas ferramentas apoiam o *design* de diagramas UML, e que as universidades preparam pessoal com este conhecimento; não precisam de mais argumentos. Os gêmeos não estão totalmente convencidos e pedem tempo para ler mais sobre o tema. Sugerem que, enquanto isso, um método sistemático de *design* dos testes que inclua a descrição de casos de uso seria suficiente para mitigar muitos dos riscos. Estiveram seguindo

<sup>86</sup> *Domain Neutral Components*, tradução dos autores.

as ideias de Richard Denney desde sua primeira publicação em StickyMinds<sup>87</sup>, e acharam seu último livro<sup>88</sup> bastante útil. Apresentam então uma proposta alternativa: usar histórias de usuários para se manter dentro de *eXtreme Programming*, elaborar com elas um diagrama de casos de uso, identificar casos omissos usando uma matriz CRUD<sup>89</sup>, e desenvolver um perfil de operações seguindo as técnicas do livro de Denney e, para aqueles casos que mereçam<sup>90</sup>, desenvolver casos de uso completos seguindo os parâmetros do livro de Allistair Cockburn<sup>91</sup>. Poderão, assim, certificar-se de que todas as entidades necessárias estão sendo consideradas nos requisitos. Para assegurar que a transição a partir dos requisitos ao código se realize com fluidez, podem utilizar o Desenvolvimento Baseado em Testes (*Test Driven Development* ou *TDD*<sup>92</sup>). Desse modo, opinam, solucionarão a maioria dos problemas sem modificar muito o tratamento atual dos requisitos.

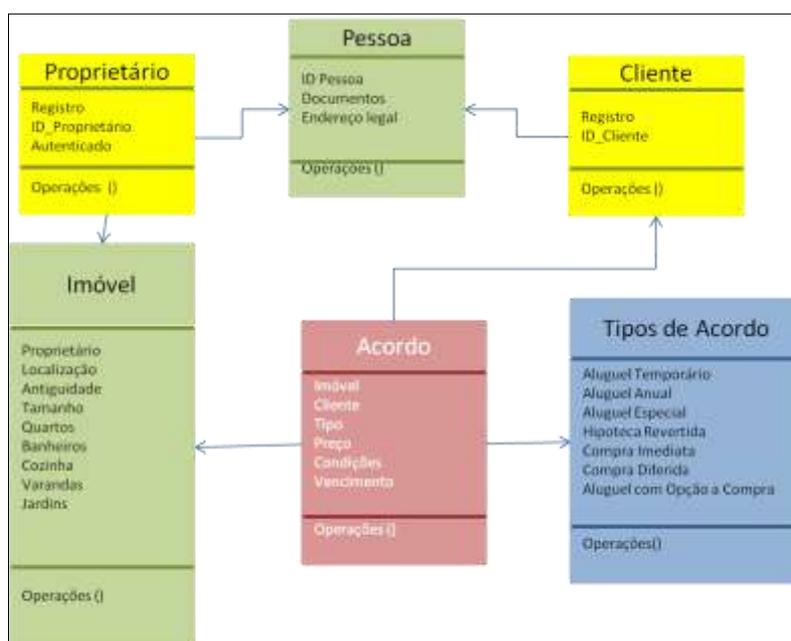


Figura 8.4: Diagrama de Classes de Acordo

Como  $T^2$  é uma organização que amadureceu muito, as opiniões não se discutem a não ser que sejam submetidas à análise e comparações. Os quatro montam uma matriz de Pugh que cruza os problemas com as soluções. Na interseção de cada fila (problemas) e cada coluna (soluções) coloca-se uma letra que indica a contribuição da coluna para essa solução. Por enquanto basta categorizá-las como A (alto), M (médio) ou B (baixo). Se não há nenhum impacto da solução no problema, deixa-se a célula em branco.

As duas soluções são colocadas nas colunas: Modelado em Cor com UML e XP ampliado com melhor *design* de testes. A Tabela 8.2, Comparação entre Métodos de Desenvolvimento pelo Risco, mostra os resultados como são entendidos pela equipe de análise. Da matriz fica claro

<sup>87</sup> <http://www.stickyminds.com/>

<sup>88</sup> [DENNEY, 2012], *Use Cases Levels of Test. A Four-Step Strategy for Budgeting Time and Innovation in Software Test Design*.

<sup>89</sup> CRUD é a sequência de iniciais das palavras inglesas *CREATE*, *READ*, *UPDATE*, *DELETE*. Estas atividades surgem do uso típico de um sistema de informação e a falta de uma dessas funções sugere uma especificação incompleta, sendo mais frequente a falta das eliminações de registros que já não são necessários.

<sup>90</sup> Seguindo Denney, os casos que tenham maior frequência de uso são desenvolvidos primeiro.

<sup>91</sup> [COCKBURN, 2000], *Writing Effective Use Cases*.

<sup>92</sup> Fundamental em XP, o TDD baseia-se em um ciclo curto de repetições: primeiro o desenvolvedor escreve um caso de teste automatizado que define uma melhoria desejada ou novas funcionalidades. O código sem modificar precisa fazer com que esse teste fracasse, a não ser que o rechace. O novo código que é produzido pode então ser validado por verificação por meio de um novo teste. Se for necessário reacomodar o código para que se cumpra com normas de legibilidade ou semelhantes, ele será redesenhado usando Refatoração e “*Smells*”.

que os dois enfoques são bastante poderosos, mas nenhum é completo. Há elementos de processo a acrescentar em todos os casos e é pouca a vantagem de um método sobre o outro. Por causa da paridade, a equipe decide incorporar mais uma análise, desta vez baseando-se na curva de aprendizagem e no risco da adoção.

<b>risco:</b>	<b>mitigação:</b>	JMC	XP + TST
informação incompleta sobre necessidades e requisitos	processo sistemático de identificação	A	M
especificações incompletas	documento onde priorizar requisitos e especificá-los	M	A
funções e requisitos não funcionais sem especificar	identificar funções, assim como atributos de qualidade	B	M
muitos requisitos derivados depois do jogo do planejamento	definir detalhes da porção de funcionalidades no <i>sprint</i>	M	M
não há clara definição das interfaces	especificar interfaces internas e externas	A	A
não há contexto onde situar as especificações	construir histórias de usuário, narrativas e/ou casos de uso	A	A
passa-se à estimativa sem fazer uma análise mais profunda da funcionalidade	equilibrar requisitos e necessidades do cliente contra restrições da equipe	M	A
os modelos não são discutidos com o dono do produto	validar o quanto antes os requisitos menos comuns	A	A

**Tabela 8.2: Comparação entre Métodos de Desenvolvimento pelo Benefício**

	JMC	XP + TST
curva de aprendizagem	A	B
risco de adoção	A	B

**Tabela 8.3: Comparação entre Métodos de Desenvolvimento pelo Risco**

Apesar de ser um pouco melhor usar UML com as extensões de modelamento em cor, é muito mais simples continuar como até agora, com o agregado dos casos de uso e as técnicas propostas por Denney.

Vencidas pelas provas, Ana e Marcela começam a trabalhar com os gêmeos na melhoria, resumindo as técnicas em um processo (ver Figura 8.5).

Alguns procedimentos do processo anterior são expandidos para que possam ser executados de maneira similar em todos os casos. Por exemplo, para definir histórias de usuário são agregados passos que aprofundam a funcionalidade no jogo do planejamento, em particular durante a discussão com o Dono do Produto. O Dono do Produto participa ativamente na criação destas histórias.

Uma história de usuário é uma ou mais frases na linguagem comum do usuário final, que capta o que faz ou precisa fazer como parte de sua função. Captura 'quem', 'o quê' e 'por quê' de um requisito, de uma forma simples e concisa, muitas vezes limitada em detalhes que podem ser manuscritos em um guardanapo de papel. Por exemplo, "Um cliente se registra no sistema para buscar imóveis para alugar ou comprar". As histórias de usuário são a principal forma de influenciar a funcionalidade do sistema a ser desenvolvido. Podem também ser escritas pelos mesmos engenheiros de desenvolvimento para expressar requisitos não funcionais (segurança, qualidade, desempenho, etc.) ou requisitos derivados, por exemplo aqueles que se omitiram na definição do *backlog* do *sprint* porque foram assumidos como óbvios (uma verificação de saldos em uma conta, a existência do cliente em um repositório, etc.) As histórias de usuário são traduzidas em casos de uso como diagrama de casos de uso.

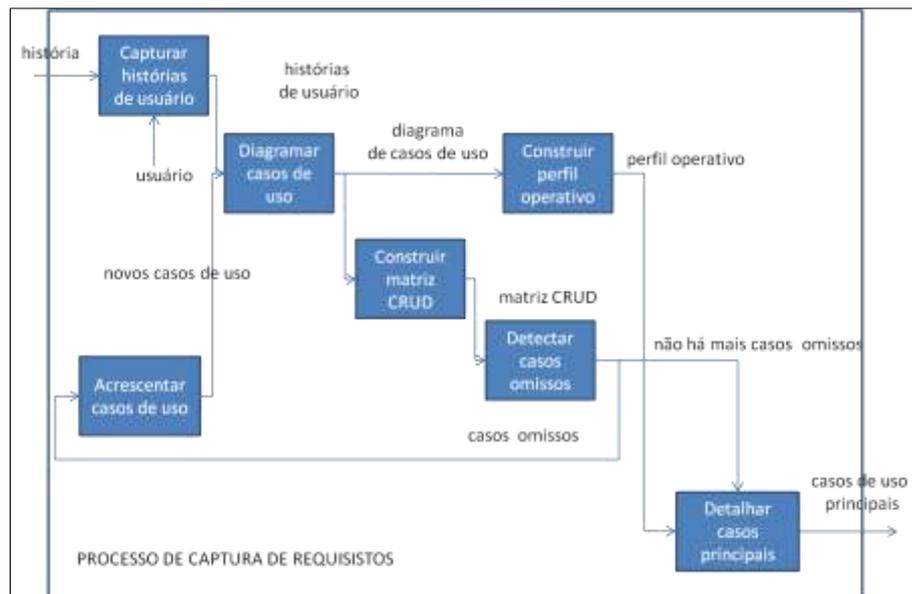


Figura 8.5: Processo de Captura de Requisitos

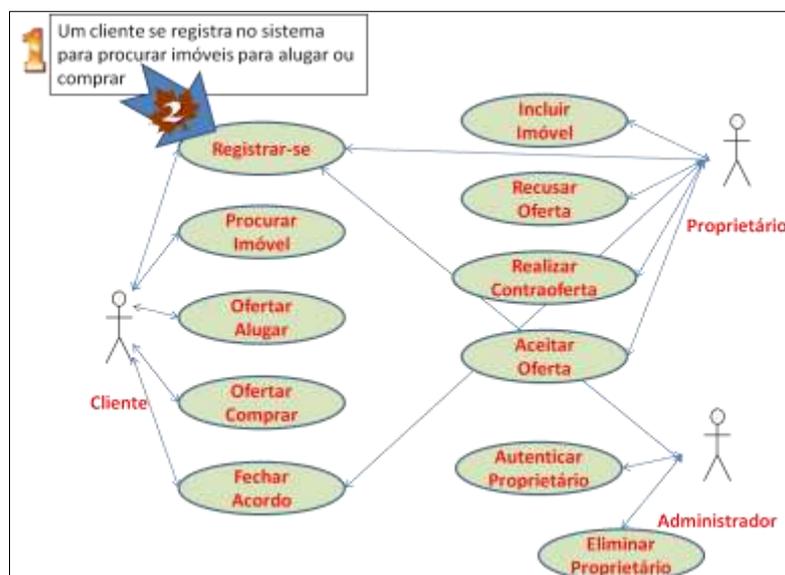


Figura 8.6: Resultado dos Passos 1 e 2

Segundo o novo processo, o próximo passo é gerar o diagrama de casos de uso com os atores, como ilustra a Figura 8.7.

Segundo Denney devemos nos assegurar, agora, que não existam mais casos de uso que estejam faltando. São listados primeiro os casos de uso: Registrar-se; Procurar imóvel; Ofertar Aluguel; Ofertar Compra; Fechar Acordo; Incluir Imóvel; Recusar oferta; Realizar contra-oferta; Aceitar oferta; Autenticar proprietário; Eliminar proprietário.

Depois são listadas as classes que surgem dos casos de uso e estas são analisadas para verificar se as quatro funções CRUD estão definidas para cada uma delas. Para continuar com o exemplo proposto, considerando o primeiro caso de uso, Registrar-se, sendo que tanto o ator Cliente quanto o Proprietário utilizam esse caso de uso, é razoável supor que teremos classes para cada um. São incluídos na lista de Classes. Na ordem dos casos de uso, segue Procurar Imóvel; Imóvel vai para a lista. Assim tem-se a seguinte lista de classes: A: Cliente; B: Proprietário; C: Imóvel; D: Acordo de Aluguel; E: Acordo de Compra; F: Oferta; Nossa matriz agora tem Atores na primeira coluna e casos de uso em cada uma das seguintes. O resultado é a Tabela 8.4.

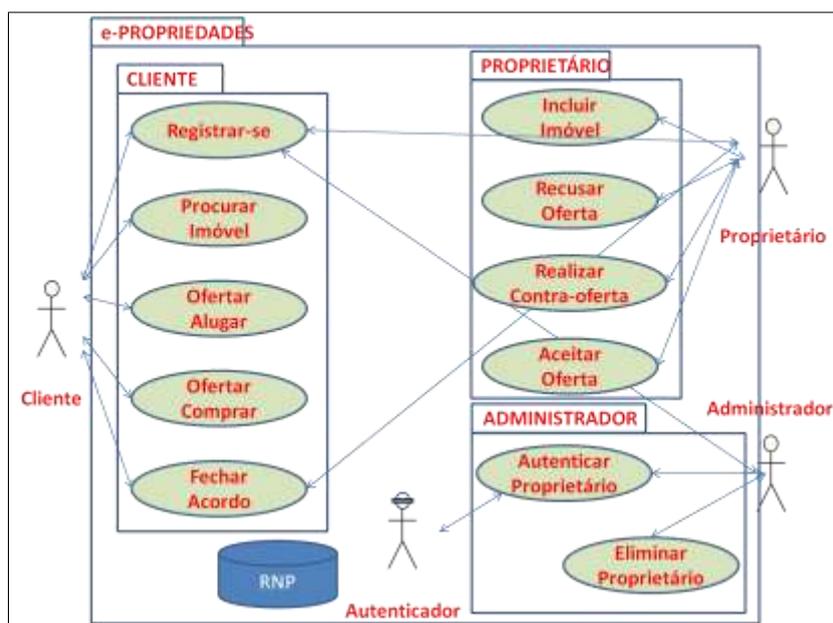


Figura 8.7: Diagrama de Arquitetura

CLASSES \ CASOS DE USO	CASOS DE USO										
	Registrar-se	Procurar imóvel	Ofertar Alugar	Ofertar Comprar	Fechar Acordo	Incluir Imóvel	Recusar oferta	Realizar contra-oferta	Aceitar oferta	Autenticar proprietário	Eliminar proprietário
Cliente											
Proprietário											
Imóvel											
Acordo de Aluguel											
Acordo de Compra											
Oferta											

Tabela 8.4: Matriz CRUD sem Completar

A matriz é completada colocando na interseção de cada interseção linha e coluna um C, R, U ou D segundo o caso de uso da coluna crie, leia, atualize ou elimine um objeto da classe da fila. A matriz assim completada é vista na Tabela 8.5.

CLASSES \ CASOS DE USO	CASOS DE USO										
	Registrar-se	Procurar imóvel	Ofertar Alugar	Ofertar Comprar	Fechar Acordo	Incluir Imóvel	Recusar oferta	Realizar contra-oferta	Aceitar oferta	Autenticar proprietário	Eliminar proprietário
Cliente	C					U					
Proprietário	C	R								U	D
Imóvel					U	C					
Acordo de Aluguel					C	U		U	C		

Acordo de Compra					C	U		U	C		
Oferta			C	C	D		D				

Tabela 8.5: Matriz CRUD já Completada

Todas as classes têm pelo menos um caso de uso que cria objetos de sua classe. Os casos de uso Aceitar Oferta e Fechar acordo são vistos estranhamente iguais. Talvez se trate de um requisito redundante, e será preciso separá-lo de forma detalhada com o Dono do Produto. A grande quantidade de espaços em branco sugere que devemos analisar, mais profundamente, com o Dono do Produto, as regras do negócio. Por exemplo, analisar se a criação de um acordo atualiza os objetos que se relacionam com ele, o cliente e o proprietário, além do imóvel. Parece lógico que seja assim e deveríamos confirmá-lo. Isto nos leva a pensar que o objeto Oferta cria relações similares, ainda não marcadas na matriz. Por outro lado, não há casos de uso que atualizem ou eliminem objetos da classe Cliente, de modo que depois de criado, o objeto é persistente e totalmente inerte. Objetos com essas características são pouco úteis em geral. Depois, em um caso real, perguntaríamos se existem regras de negócio que nos dizem o que fazer com um Cliente, como se atualiza sua informação e como, se for o caso, ele pode ser excluído no sistema.

Da mesma forma, falta informação sobre imóveis e acordos, que parecem ser modificados uma única vez. Certamente, os acordos se renovam ou vencem, os imóveis podem sair do sistema. Inclusive nos perguntamos se, ao excluir um Proprietário, não corresponderia excluir nos objetos relacionados com ele ou ela. Nosso conhecimento do sistema, que parecia tão completo quando olhávamos a Figura 8.7, agora parece muito resumido. Com um método simples, mas exaustivo, podemos assegurar que identificamos casos de uso que faltam ou são redundantes e regras de negócio incompletas.

### 8.5 Perfil Operacional

Descreveremos agora o passo seguinte, construir o primeiro nível do perfil operacional, que refinaremos depois para cada caso de uso. O propósito desta atividade é evitar detalhar casos de uso que são pouco frequentes e que, portanto, implicam pouco risco. Nas palavras de Denney<sup>93</sup>, “é melhor ter uma estratégia para utilizar o tempo sabiamente, e saber quais técnicas funcionam melhor (ou não) em diferentes problemas”.

Para simplificar, vamos supor que já consultamos com o cliente as regras de negócio e este nos disse que, por enquanto, só quer incluir mais dois casos de uso, Limpar Dados, que levará em conta eliminações de objetos que caducaram, e Atualizar Dados, que usarão Cliente e Proprietário para renovar suas identificações. Para construir nosso perfil de uso, devemos entender a frequência de uso de cada caso de uso para cada um dos atores. Obviamente, no caso de um produto existente, investigaremos o comportamento atual e o ajustaremos aos desejos ou necessidades futuras do cliente. Se o sistema é novo, deveremos utilizar a visão do Dono do Produto para conseguir os dados.

Nossa matriz tem agora Atores na primeira coluna e casos de uso em cada uma das seguintes, mas foi acrescentada uma coluna que identifica a quantidade de atores esperados de cada um dos tipos. Por exemplo, existe só um Administrador, mas se o sistema é novo, a quantidade de usuários dos outros dois tipos é desconhecida. Segundo Denney<sup>94</sup>, a precisão pode ser desprezada neste momento, basta conhecer a ordem de magnitude relativa. É importante reconhecer que ordens de magnitude são suficientes para identificar perfis de uso. Em nosso caso, diremos que há mil Proprietários para cada administrador e dez Clientes para cada Proprietário no sistema.

Agora é preciso quantificar o uso dos casos de uso pelos atores. Primeiro escolhemos (ênfase em “escolher”) um intervalo de tempo conveniente. Pode ser um segundo, um minuto ou um ano. O importante é que sirva para estimar números razoáveis. Em nosso caso, digamos que se conta com estimativas mensais para cada ator, de modo que o intervalo escolhido seja o mês. Na matriz que estamos gerando (chamada QFD<sup>95</sup> por Denney, dada a semelhança

<sup>93</sup> Use Case Level of Test, op.cit., página 6.

<sup>94</sup> Use Case Level of Test, op.cit., páginas 40 e 41.

<sup>95</sup> Desdobramento da função qualidade (QFD – *Quality Function Deployment*) é um método de gestão de qualidade baseado em transformar as demandas do usuário na qualidade do *design*, implementar as  
(a nota continua na próxima página)

existente entre sua matriz de quantificação de perfil com a usada pelo método desse nome) coloca-se em cada célula a estimativa de frequência de uso mensal do caso de uso (coluna) pelo Ator (linha). Este número é um número real positivo que é omitido quando a frequência é exatamente 0. Por exemplo, foi estimado em um estudo de mercado realizado pela empresa cliente, que serão incluídos 59 Proprietários por mês que no total (novos e já registrados) incluirão uma média de 0,27 propriedades cada um por mês, levando em conta publicações de avisos de aluguel na rede e outros meios (avisos classificados dos jornais). Seguindo com as estimativas de modo parecido ao modelo de tomar como referência ordens de magnitude, o resultado é o da Tabela 8.6; Estimativas de Atividade.

por mês	
600	clientes novos
59	proprietários novos
63,13	imóveis novos
4000	buscas compra
8500	buscas aluguel
1000	oferta compra
3000	oferta aluguel
80	contraoferta compra
12	contraoferta aluguel
2	recusas
3830	acordos fechados
59	autenticações de proprietário
0,01	Eliminações de proprietário

**Tabela 8.6: Estimativas de Atividade**

Com essas estimativas, a matriz de QFD fica com os seguintes dados:

Quantidade	CASOS DE USO ATORES	Registrar-se	Procurar imóvel	Ofertar Alugar	Ofertar Comprar	Fechar Acordo	Incluir Imóvel	Recusar oferta	Realizar contraoferta	Autenticar proprietário	Eliminar proprietário	Limpar Dados	Atualizar Dados
		10000	Cliente	0,06	0,13	0,03	0,01	0,04					
1000	Proprietário	0,06				0,38	0,06	0	0,02				
1	Administrador									59	1	110	330
	execuções por mês	659	1250	300	100	766	63	2	20	59	1	110	330
	peso relativo	18%	34%	8%	3%	21%	2%	0%	1%	2%	0%	3%	9%
	ranking	3	1	5		2							4
	% de esforço	20%	38%	9%		23%							10%

**Tabela 8.7: Perfil Operacional dos Casos de Uso**

Os valores de cada célula surgem da divisão dos dados correspondentes a cada caso de uso pela multiplicidade correspondente do ator que o executa. Por exemplo, se há cem ofertas de

funções que apótem mais qualidade, e implementar métodos para alcançar qualidade do *design* em subsistemas e componentes e, em última instância, aos elementos específicos do processo de fabricação. [http://en.wikipedia.org/wiki/Quality\\_function\\_deployment](http://en.wikipedia.org/wiki/Quality_function_deployment) (N.A.: a redação desta página indica sua origem em alguma tradução automática, o que deixa com um baixo valor estético, mas ainda é legível).

compra por mês e são dez mil atores, cada um deles executa 0,01 vezes o caso de uso correspondente. As execuções por mês resultam da soma dos produtos desse número por esta multiplicidade. Por exemplo, se há 0,06 execuções de Registrar-se por parte de Clientes e 0,059 execuções do mesmo caso por parte de Proprietários, o produto de 0,06 x 10000 somado ao produto de 0,059 por 1000 dá 659. Somando todas as execuções, o resultado é 3660 que, usado como divisor dos números anteriores, dá a porcentagem que corresponde a cada um. Assim, 1250 dividido por 3660 dá aproximadamente 34%, que é o caso de uso de mais alto *ranking*. Descartando os casos de uso de baixa frequência, pode-se recalculá-lo pelo esforço esperado e conseguir calcular a dedicação a cada caso de uso.

Dados os números assim calculados, faz sentido investir nos casos de uso que ocupam os primeiros 4 lugares porque, em conjunto, representam 82% do uso do sistema. Os 5 primeiros somam 90%, enquanto que os 3 primeiros somam 73%. Para quem fizer a escolha, o método é claro: são eleitos para serem detalhados os casos de uso que mais peso têm em termos de utilização pelos usuários. A exceção ou exceções podem vir do risco que colocam alguns casos de uso em particular, por exemplo, pode ser que o caso de uso Autenticar Proprietário seja muito importante em termos de negócio, de modo que, apesar de seu relativo baixo peso, seja considerado para ser detalhado de todos os modos.

### 8.6 Detalhando Um Caso

Cada caso de uso deve descrever como se utiliza o sistema em um aspecto único do negócio, quer dizer, descreve, a partir do ponto de vista do usuário, o comportamento do sistema quando este ajuda um usuário a alcançar um objetivo de negócio com ele. Para a maioria de projetos de *software*, isto significa que talvez seja necessário especificar centenas de casos de uso para definir completamente o novo sistema. Os projetos ágeis, com sua definição de 'sashimi' para cada *sprint*, não precisam de tantos. Se, além disso, são selecionados, como vimos na seção anterior, aqueles que são mais representativos, o resultado é manipulável pela equipe em poucas horas.

Um caso de uso contém uma descrição textual de todas as maneiras que se espera que os atores interajam com o sistema. Os casos de uso não descrevem funcionalidades internas do sistema, nem expõem como serão implementadas. Por outro lado, mostram os passos da interação do sistema com o ator. Para ser útil, um caso de uso deve descrever uma única tarefa do negócio que sirva a um objetivo de negócio, porque, se há muitos objetivos, o resultado é um produto muito complexo. É importante ter um nível de detalhe apropriado, nem muito detalhado nem muito simples, e ser bastante simples a ponto de que um desenvolvedor o elabore em um período curto. Para evitar escrever longos casos de uso, há objetivos e subobjetivos, de modo que um caso de uso estende outro caso de uso; ou um caso de uso pode chamar outro caso de uso. O detalhamento de cada caso selecionado implica seguir certas convenções para evitar ter um conjunto de casos de uso díspares em tamanho e extensão e, conseqüentemente, incomparáveis. Neste caso, Marcela dá preferência ao formato proposto por Allistair Cockburn em seu livro<sup>96</sup>. Frequentemente, é proposta a seguinte estrutura no *design* de casos de uso (Tabela 8.8):

Componentes de um Caso de Uso	Definição
ID	Segundo nomenclatura de configuração
NOME	Descritivo do caso de uso (CDU)
REFERÊNCIAS CRUZADAS	Outros materiais que são utilizados para entender o CDU
CRIADO POR	Nome do Autor
ÚLTIMA ATUALIZAÇÃO POR	Nome do Revisor ou Atualizador
DATA DE CRIAÇÃO	Óbvia
DATA DA ÚLTIMA ATUALIZAÇÃO	Óbvia
ATORES	Todos os que intervêm, robôs inclusive

<sup>96</sup> [COCKBURN, 2000], *Writing Effective Use Cases*.

DESCRIÇÃO	Texto explicativo de alto nível (opcional)
DISPARADOR	Evento que dispara o CDU
PRÉ-CONDIÇÃO	Estado do sistema que permite executar o CDU
PÓS-CONDIÇÃO	Estado do sistema que surge após executar o CDU
FLUXO NORMAL	Série de passos que são considerados normais
FLUXOS ALTERNATIVOS	Série de passos que são considerados especiais
INCLUSÕES	Outros CDU que são referenciados por este
FREQUÊNCIA DE USO	De acordo com o perfil operacional
REGRAS DE NEGÓCIO	Aquelas que se aplicam ao CDU
REQUISITOS ESPECIAIS	<i>Hardware</i> ou ambiente de execução ou desempenho
NOTAS E ASSUNTOS	Comentários para revisores ou futuros usuários

**Tabela 8.8: Componentes Sugeridos dos Casos de Uso**

A planilha anterior é uma entre muitas variações. Para ver um exemplo dos campos de Fluxo ver Tabela 8.15 mais abaixo. Possivelmente um caso de uso complexo possa requerer todos esses campos e alguns mais. Às vezes são separados os fluxos alternativos em duas categorias, uma delas dedicada ao tratamento de exceções, como quando uma condição intermediária não é cumprida. É o caso do fluxo que atende ao usuário que esqueceu sua senha no CDU de Entrada. Outras vezes, estes casos são derivados de Casos de Uso especialmente criados e incluídos. A decisão de qual planilha usar depende exclusivamente das necessidades de cada situação, basta que os campos sejam utilizados uniformemente para que possam ser revisados.

#### O QUE ACONTECEU ATÉ AGORA

61. Marcela, Ana e os gêmeos analisam os problemas e os agrupam para procurar soluções.
62. Fazendo uma análise de causas profundas decide-se incorporar métodos e técnicas para o desenvolvimento dos requisitos no *Sprint*.
63. Comparando métodos por seu impacto e custo de implementação, decide-se utilizar uma mistura de XP-TDD com técnicas de desenvolvimento de casos de teste propostas por Denney.

Neste ponto, o processo de captura de requisitos já tem todos seus componentes descritos. Os gêmeos colocam em prática nas quatro equipes que dirigem, e ao final de quatro *sprints*, controlando os riscos detectados, pode-se apreciar que todos eles foram considerados.

<b>risco:</b>	<b>XP + TST</b>
os clientes nos dão informação incompleta sobre as necessidades e requisitos do produto, gerando muito retrabalho	São utilizados diagramas de CDU e matriz CRUD
a especificação de um módulo, componente ou sistema é incompleta, resultando em um produto subótimo	As prioridades são fixadas com o DP e analisadas mediante CDU
costumam sobrar funções sem especificar e os requisitos não funcionais são muito frequentemente acrescentados ao final do projeto, resultando em muito retrabalho	As planilhas de CDU identificam funções e permitem acrescentar requisitos não funcionais
há muito impacto nos compromissos pela quantidade de requisitos derivados que surgem depois do jogo do	O jogo do planejamento passa a incluir o detalhe dos requisitos.

planejamento	
a integração contínua fracassa mais do que o necessário porque não há uma clara definição das interfaces	A planilha de CDU contempla esse ponto
são mal interpretadas as funções pedidas porque não há um contexto onde situá-las, resultando em demonstrações fracassadas	Os CDU cobrem este risco
há certa pressa em passar a estimativa sem fazer uma análise mais profunda da funcionalidade que é implementada, resultando em oportunidades perdidas e ineficiências	Os perfis operacionais se ocupam de estabelecer esse equilíbrio quando há pressões de recursos
ainda naqueles casos em que se constroem modelos do produto, estes são poucas vezes discutidos com o dono do produto	O jogo do planejamento passa a incluir o detalhe dos requisitos e sua revisão

**Tabela 8.9: Lista de Controle de Mitigação de Riscos em Requisitos**

Marcela incorporou, aos passos de aprovação do procedimento, a criação de um mapa dos resultados esperados do MR-MPS-SW que a mudança traz. Isto é realizado sem exigência de completar necessariamente todos os resultados, mas para entender melhor a lacuna entre o que já foi implementado e o que resta para cumprir com o requisitos de uma avaliação. O resultado é detalhado na Tabela 8.10.

<b>Resultados Esperados do DRE no MPS</b>	<b>Evidência de implementação na Tahini-Tahini</b>
<b>DRE1 As necessidades, expectativas e limitações do cliente, tanto do produto quanto de suas interfaces, são identificadas;</b>	segue-se um processo sistemático para a identificação do que o cliente necessita, para que tenha o que quer segundo o que diz
<b>DRE2 Um conjunto definido de requisitos do cliente é especificado e priorizado a partir de necessidades, expectativas e limitações identificadas;</b>	constrói-se um documento que permite priorizar os requisitos e especificá-los, de modo que sua revisão e análise seja possível
<b>DRE3 Um conjunto de requisitos funcionais e não funcionais, do produto e dos componentes do produto que descrevem a solução do problema a ser resolvido, é definido e mantido a partir dos requisitos do cliente;</b>	a especificação dos requisitos deve permitir identificar funções, assim como os atributos de qualidade não funcionais do produto
<b>DRE4 Os requisitos funcionais e não funcionais de cada componente do produto são refinados, elaborados e designados;</b>	antes de iniciar um <i>sprint</i> , devem ser definidos os detalhes das funcionalidades que serão incluídas
<b>DRE5 Interfaces internas e externas do produto e de cada componente do produto são definidas;</b>	parte da especificação deve ser dirigida expressamente ao ambiente de funcionamento do produto, sejam interfaces internas ou externas
<b>DRE6 Conceitos operacionais e cenários são desenvolvidos;</b>	como parte do mecanismo de entendimento dos requisitos, é necessário construir histórias de

	usuário, narrativas e/ou casos de uso que expliquem o uso esperado do produto
<b>DRE7 Os requisitos são analisados, usando critérios definidos, para equilibrar as necessidades dos interessados com as restrições existentes;</b>	a seleção de processos em um <i>sprint</i> baseia-se na análise de requisitos e necessidades do cliente, equilibradas com as restrições de capacidade da equipe
<b>DRE8 Os requisitos são validados.</b>	durante o jogo do planejamento, os requisitos menos comuns serão validados usando modelos e os casos de uso ou histórias de usuário

Tabela 8.10: Implementação de DESENVOLVIMENTO DE REQUISITOS na T<sup>2</sup>

## O QUE ACONTECEU ATÉ AGORA

64. Dirigindo os procedimentos em quatro *sprints*, verifica-se que foram controlados todos os riscos detectados
65. Marcela constata que os resultados esperados para DRE foram cobertos com a implementação dos novos procedimentos

**8.7 Detectando Defeitos nos Produtos**

Os riscos controlados pelas mudanças realizadas no processo de requisitos deveriam diminuir a densidade de defeitos. Na reunião posterior aos quatro meses de acompanhamento em projetos pilotos, os números são mistos. Apesar de terem sido eliminados os problemas que são introduzidos em um *sprint* que chega ao usuário, os problemas legados de *sprints* anteriores à reestruturação do processo começaram a surgir com força e são o foco da atenção neste momento.

Este fenômeno é frequente e uma das causas mais habituais de frustração com a melhoria de processos. Resolver um problema faz com que seja evidente a existência de outro que antes era escondido pela existência do primeiro problema. O fenômeno, conhecido pelos *designers* de sistemas operacionais, denomina-se ‘problema de engarrafamento’, porque é ilustrado com esta conhecida moléstia do trânsito. A história é assim: suponhamos que um ponto de um caminho produz engarrafamentos diários a um custo anual ao público de cem mil horas de espera, somadas todas as partes envolvidas. Com relação aos cofres públicos calcula-se que uma hora de espera tem um custo social de cerca de 100 reais. Assim, o ponto em questão ‘custa’ à sociedade dez milhões de reais anuais. O Estado inicia planos para eliminar o problema que custam vinte milhões de reais e, depois de concluídos, espera-se que o que foi gasto seja recuperado em dois anos. Depois que as obras foram entregues, o engarrafamento não acontece mais nesse ponto, mas em uma parte um pouco mais abaixo na mesma estrada surge um novo engarrafamento, que antes não era detectado porque o trânsito não chegava a esse segundo ponto na densidade suficiente para criar um engarrafamento. As horas perdidas são agora quase as mesmas, mas em outro ponto.

Este desmascaramento ocorre na melhoria de processos também. Podemos considerar um projeto como uma série de estações unidas por traçados pré-definidos (os processos) que levam os produtos de uma a outra. Em cada estação o produto é transformado (de necessidade do cliente em caso de uso, de caso de uso em código, de caso de uso em caso de teste, de código não testado em código testado, e assim por diante). Se o que oferta o serviço está ocupado, o produto espera para ser liberado. Isto produz filas. Acelerar o serviço em uma estação, neste caso a detecção de defeitos, faz com que outra estação mais adiante tenha agora uma fila de espera. Vimos o tratamento que ocorre em *Kanban* sobre este tema no Capítulo 3, focado em liberar o serviço da melhor maneira possível e o quanto antes. Nesta seção nos limitaremos a mostrar as atividades que são necessárias para melhorar o serviço que segue.

Neste caso o problema é que se eliminou a introdução de problemas derivados de requisitos incompletos, ambíguos ou inconsistentes, mas o usuário continua encontrando defeitos porque o produto continua mostrando defeitos que não foram detectados em seu momento e que foram introduzidos em muitos *sprints* anteriores. A T<sup>2</sup> enfrenta o problema de detectá-los antes que o cliente. Revisando as planilhas analisadas junto com os Problemas de Requisitos, que se

conglomeraram em outras quatro, a saber: 'Problemas de *Design*', 'Problemas de Integração', 'Problemas de Verificação' e 'Problemas de Validação', os quatro protagonistas de nossa história na T<sup>2</sup> concluem que há motivos para continuar modificando o que fazem. Primeiro consideram os problemas de Validação, porque no sentido mais estrito afetam o cliente. Das notas que se tomaram na análise resgatam a Tabela 8.12.

<b>risco ou problema:</b>	<b>mitigação:</b>
em algumas ocasiões preparamos validações com o cliente que não cumpriam as expectativas do que eles queriam ver, por exemplo omitimos documentação, perdendo credibilidade e tempo	decidir com o cliente e a equipe a estratégia de validação, os produtos a avaliar, o cronograma de validação, os critérios de entrada e de aceitação e os ambientes de aplicação
em algumas ocasiões preparamos validações que seguiam uma ordem incorreta e confundiam o cliente, fazendo com que tivéssemos que refazer a apresentação	
frequentemente os clientes não compartilham nossos critérios de qualidade, sobretudo quando os requisitos foram disputados várias vezes	
em algumas ocasiões o produto "roda" em nosso ambiente, mas não no ambiente do cliente	
para resolver o problema com o cliente, fizemos correções durante o desenvolvimento que não ficaram registradas e tiraram de sincronização a <i>baseline</i>	reforçar o seguimento do processo de aceitação do usuário, aumentando a frequência das auditorias e ajudando a prevenir não conformidades, visando assegurar que os critérios de aceitação sejam cumpridos, baseados em documentação fidedigna
resolvemos de mais e de menos em muitas ocasiões, desperdiçando esforço ou perdendo qualidade	
em algumas ocasiões liberamos código sem o suficiente respaldo	

**Tabela 8.11: Problemas de Validação**

Comparando as ações de mitigação desta tabela com as da Tabela 8.12, Problemas de Verificação, Marcela encontra semelhanças que sugerem que a melhoria do processo de engenharia de testes pode atender tanto à Validação quanto à Verificação.

<b>risco ou problema:</b>	<b>mitigação:</b>
apesar do plano de testes ser geral, a inspeção é seletiva e nem sempre escolhemos bem os produtos ou partes deles que devem ser inspecionados, ou então escolhemos métodos mais fracos do que deveríamos	decidir com a equipe a estratégia de verificação, incluindo: os produtos a avaliar e com quais métodos; o cronograma das diferentes avaliações; os critérios de entrada, descontinuidade e aceitação (garantindo que a cobertura mínima seja um deles); e os ambientes de teste, quando aplicável
nem sempre planejamos com antecipação suficiente as atividades de teste ou verificação e perdemos participantes importantes	
a falta de cobertura na definição de critérios nos levou a tomar decisões equivocadas sobre a qualidade do produto demonstrada pela densidade de defeitos	
quando o <i>sprint</i> está por terminar, cortou-se caminho omitindo testes importantes, como o de estresse	aplicar com todo rigor auditorias de processo e produto até que seja fixada a conduta de teste sistemático
nem sempre a densidade que calculamos está	automatizar todos os testes

baseada em dados confiáveis	
deveríamos poder justificar perante a equipe o retorno do investimento de verificar usando inspeções	os dados obtidos dos testes e das inspeções, <i>walkthroughs</i> e revisões técnicas, devem ser analisados e discutidos na reunião mensal de portfólio de projetos

Tabela 8.12: Problemas de Verificação

Desde o começo, Marcela deixou clara a diferença entre Validação e Verificação. Apesar de que há escolas opostas no que diz respeito ao significado destes dois termos, Marcela, seguindo os conselhos de Máximo, utiliza as interpretações de Barry Boehm em sua obra<sup>97</sup>. Para eles, então, verificar é comparar se o que é realizado cumpre com o especificado, enquanto que validar é garantir que o produto satisfaz as necessidades do cliente.

Para verificar seus produtos, a T<sup>2</sup> utiliza múltiplas ferramentas. As revisões<sup>98</sup>, sejam revisão por pares, inspeções, *walkthroughs*<sup>99</sup>, revisões estruturadas ou revisões contínuas<sup>100</sup>, são usadas para constatar que o produto sob revisão tem a qualidade esperada e é consistente, não só em si mesmo, mas com os produtos que o antecedem, fundamentalmente com os requisitos do sistema. Os testes de programa são utilizados para provocar quedas ou falhas do sistema com o propósito de detectar defeitos no produto<sup>101</sup>, mas também como ferramenta de *design*<sup>102</sup> e guia para a construção do programa, como já vimos no Capítulo 3. Desenvolveremos tanto as revisões em geral quanto métodos e técnicas de teste de programas.

Para validar seus produtos, a T<sup>2</sup> usa atividades nas quais participa o DP<sup>103</sup>. Ao planejar o *Sprint*, a equipe cria o modelo que representa a funcionalidade desejada e o “executa” na frente do DP. Isto permite validar a visão da equipe, no que ela traduziu o que entendeu um objeto de fronteira que o DP pôde interpretar por sua vez, achando então diferenças ou até esclarecendo sua própria visão. Ao finalizar o *Sprint*, a equipe apresenta uma demonstração do produto ao DP, para constatar que a visão compartilhada se materializou corretamente. Estas duas atividades são efetivamente atividades de validação, voltadas para avaliar a eficiência do produto para o cliente.

### 8.8 Procedimentos de Verificação

Já cobrimos no Capítulo 3 a revisão contínua, como programação por pares ou em dupla. Aqui desenvolveremos em detalhe os procedimentos de revisão clássicos: *walkthroughs*, revisão formal estruturada e inspeção.

As revisões são uma parte fundamental de toda atividade de criação de um produto. Em particular, na engenharia de *software* são indispensáveis porque, se adiamos o teste do produto até o último momento, colocamos em perigo a totalidade do projeto. Vamos argumentar este último começando por uma definição que foi evoluindo na década de 60 do século passado, pela qual o processo de construção de um artefato de *software* pode ser visto como a tradução de um conteúdo semântico de uma sintaxe para outra, com preservação da semântica e o possível acréscimo de detalhes em cada passo. Por exemplo, o documento de especificação com casos de uso pode ser visto como um refinamento do documento de histórias de usuário, com uma nova sintaxe (a do caso de uso), mas preservando o significado (a semântica) do documento anterior. Da mesma forma, o caso de teste pode ser visto como

<sup>97</sup> De maneira muito geral, pode-se dizer que a verificação se ocupa de constatar se o produto está sendo desenvolvido corretamente, enquanto que a validação procura assegurar que se está desenvolvendo o produto correto, quer dizer, aquele produto que o cliente necessita [BOEHM, 1981].

<sup>98</sup> Veja para maior detalhe [FREEDMAN e WEINBERG, 1990].

<sup>99</sup> N. dos A. Traduzimos *walkthroughs* como “percursos”, mantendo a consistência histórica com [BORIA, 1987].

<sup>100</sup> Seguindo a [BECK, 2000] consideramos a programação a dois uma forma extrema de revisão contínua.

<sup>101</sup> Testar o programa para “assegurar que funcione” é um erro de enfoque que diminui a eficiência do teste.

<sup>102</sup> *Test Driven Design* em [BECK, 2000]

<sup>103</sup> Dono do Produto, Capítulo 3.

um refinamento do caso de uso com outras regras sintáticas, e o código gerado como uma nova iteração desse processo de traduzir e refinar com outra sintaxe. Isto é visualizado na parte esquerda da Figura 8.8, sob o título Atividades de Tradução.

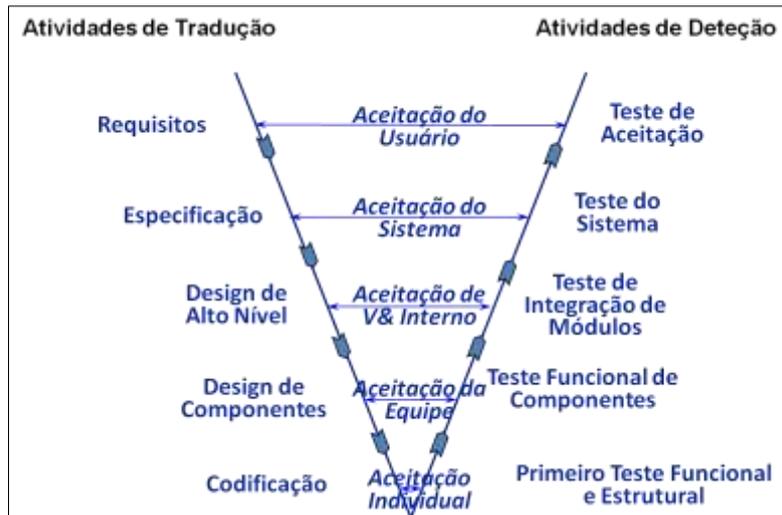


Figura 8.8: Modelo V de Desenvolvimento de Software

O ramo direito do modelo em V define a correspondência entre as atividades de tradução e as atividades de detecção de defeitos por elas introduzidos. A hipótese é que cada tradução introduz “ruído” no sentido que esse termo tem em teoria de informação, quer dizer, um elemento que dificulta a recepção da mensagem. Cada atividade, então, introduz defeitos. Se esperarmos até o código ser escrito para remover defeitos, o resultado é caótico. Certamente o projeto tem poucas reservas de tempo e esforço ao chegar a esse ponto, como é bem conhecido pelos engenheiros de teste que, consistentemente, veem seu calendário deslizar para o futuro porque o desenvolvimento ainda não alcançou o ponto onde o produto pode ser entregue a eles. Nessas circunstâncias, a detecção de defeitos não é diferente em primeira instância, mas a falta de tempo conspira contra a resolução: as modificações são feitas rapidamente e sem pensar muito. O resultado é que a correção introduz novos defeitos, com maior probabilidade do que se tivessem sido corrigidos com tempo adequado<sup>104</sup>. Se a detecção de todos os defeitos introduzidos é postergada, pode-se esperar que o resultado seja uma zona de grande atrito entre os envolvidos até o final do esforço, como mostra a Figura 8.9.

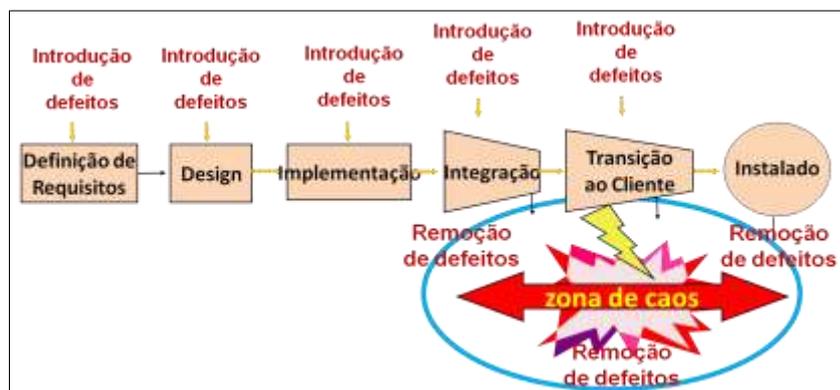


Figura 8.9: Zona de Caos por Postergação de Atividades de Remoção

Se, em contraste com este enfoque, é feito um esforço para detectar e eliminar defeitos assim que tiverem sido introduzidos, como mostra a Figura 8.10, onde foram acrescentadas atividades de detecção mais cedo (revisões indicadas pela letra R em um círculo, onde a flecha ‘CRUCIAL’ mostra as duas mais importantes), o resultado é um menor esforço de retrabalho, realizado no momento correto.

<sup>104</sup> [MYERS, 1979],

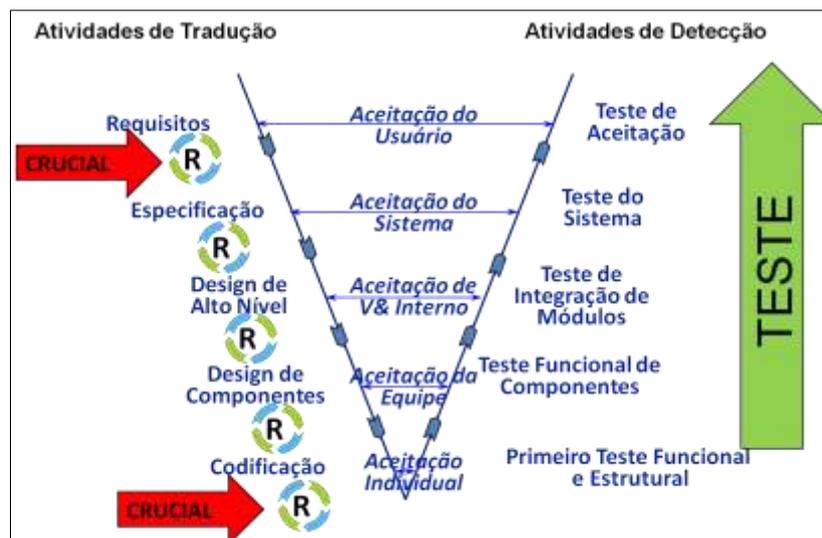


Figura 8.10: Modelo em V com Revisões entre Atividades de Tradução

### 8.9 Revisões

Os três métodos de revisão que são aconselhados, além da programação em pares, são a inspeção, que detalharemos a seguir, a revisão formal estruturada e o *walkthrough*. Os produtos que se aconselha revisar são os que estão associados a requisitos, como casos de uso ou documentos de especificação, e o próprio código. Sempre é conveniente identificar os componentes mais arriscados para cobri-los antes de ficarmos sem tempo.

Vamos começar então pelas inspeções, para depois, por diferença, definir os outros dois métodos. O que se ganha ao fazer inspeções? Qualidade, porque ao inspecionar produtos são detectados e eliminados defeitos, mas também comunicação, porque escolhendo convenientemente os participantes, consegue-se comunicar conteúdos com precisão, assim como melhorar o conhecimento dos mais novos. Ao inspecionar materiais gerados por especialistas, os mais novos aprendem com eles melhores práticas. Uma inspeção é um processo muito formal que identifica um produto, escolhe uma equipe de inspetores, escolhe as ferramentas de inspeção, detecta os defeitos no produto e garante a qualidade resultante. Uma equipe de inspeção é formada com papéis bem definidos. Deve haver um Moderador que conduza o processo e capacite, se for necessário, os inspetores. O autor dos materiais a serem inspecionados participa sem voz, escutando o que os Revisores opinam, também chamados de inspetores. Um deles, ou o Moderador, atua como Escrivão, tomando nota do que foi feito. O Moderador é alguém com habilidades de facilitador e que recebeu treinamento para dirigir inspeções. A equipe é pequena, não mais do que sete pessoas no total, contando o Moderador, o Autor, e dois a cinco Inspectores. Os inspetores são escolhidos de modo a conseguir o máximo benefício da inspeção, por exemplo, conhecem ou são os autores do produto anterior ou serão os autores do produto que se segue. Individualmente são representantes de categorias importantes dentro da organização, como arquitetos ou testadores. O que deve ser assegurado é que compreendam o produto e o processo, e são especialistas ou vão receber treinamento. Possivelmente alguém de Garantia da Qualidade participe com funções especiais, mas também é útil que a Garantia da Qualidade tenha feito uma auditoria do produto antes da inspeção.

### 8.10 Atividades do Processo de Inspeção

O processo de inspeção é ativado quando o autor do produto avisa que ele está completo. O encarregado que recebe este aviso consegue a participação de um Moderador entre o conjunto de Moderadores treinados da empresa. Este se comunica com o Autor e começam a primeira atividade de preparação, que consiste na Seleção do Material. Nela, o Autor com o Moderador decidem, juntos, quais partes do produto serão inspecionadas. Como critério de seleção, as partes a inspecionar devem estar completas, não devem ser tão extensas que incorram em grandes esforços dos inspetores, mas devem ser críticas para que valha a pena avaliá-las. Então, o Autor e o Moderador escolhem e preparam materiais acerca do produto (lucro, apresentação, etc.) e as listas de verificação e guias a serem aplicadas ou padrões que ajudem a entender o material, assim como produtos referenciados que servem de apoio.

A logística é preparada também entre o Moderador e o Autor. Ambos escolhem a equipe, designam papéis distintos (pelo menos o escrivão) e o moderador fixa as durações para revisão do material, que deve ser no máximo umas 2 horas de esforço; pode ser que em 2 dias de duração; para a reunião de instrução (uns 30 minutos); para a reunião de unificação de defeitos (não mais que 2 horas) e comunica as decisões à equipe. São consideradas as respostas ao pedido de participação e, depois de estabilizada a equipe e o calendário de atividades, procede-se à Reunião de Instrução.

### **Reunião de Instrução**

Na reunião de instrução, o Moderador “instrui” a equipe de inspetores sobre o que buscar e por que é importante, inclusive o que acontece se passar um defeito (impacto no projeto, impacto no cliente). O Moderador instrui os participantes nos processos a seguir e, se são novos em relação ao processo, são formados à parte. A Reunião também serve para distribuir os materiais a revisar e as referências, discutir os papéis que cada inspetor vai ter, ou seja, se escolhe-se fazer com que cada um “desempenhe” um papel (por ex., cliente, usuário final, *designer*, engenheiro de testes, garantia de qualidade). Nesse caso, cada papel pode ter sua própria lista de itens de verificação. O autor fornece informações sobre o produto, como resumo do material a inspecionar, responde perguntas, esclarece significados e relaciona entre si os produtos. Tudo isto é realizado em, no máximo, 30 minutos.

Os materiais distribuídos são cuidadosamente preparados. O produto terá as linhas numeradas ou identificadas de forma clara, e se não for todo o produto, destaca-se o que é preciso revisar. Entrega-se também a planilha usada para preparar o produto, assim como os padrões e referências, os materiais relacionados e as listas de itens de inspeção que servirão para identificar os defeitos. Para juntá-los, entrega-se uma planilha de registro de defeitos.

### **Inspeção Individual do Produto**

Cada inspetor revisa os materiais por sua conta e tenta encontrar TODOS os defeitos. Para isso utiliza os guias, planilhas e as listas de itens e se coloca na perspectiva de seu papel. Registra uma a uma as observações encontradas na planilha de registro de defeitos, onde marca a localização do problema, o tipo de item (pergunta, defeito, ambiguidade), e a severidade do defeito. Quando completa todos os itens que encontrou, registra o tempo e o esforço na planilha e realiza um resumo dos problemas. Se, no prazo de duas horas, não alcançou o final do produto a revisar, pode estender o tempo de inspeção pessoal, mas não além de meia hora. Em qualquer caso, quando termina o prazo sem ter conseguido terminar a tarefa, marca até onde inspecionou e dá por concluída a inspeção pessoal. Se o produto não está pronto para ser revisado pela sua baixa qualidade, ou se existe algum outro motivo para que não seja feita a reunião de unificação, caso o produto seja redundante ou esteja desatualizado, informa-se ao Moderador. Se segue o processo normal, cada Inspetor envia a planilha preenchida ao Moderador para que este a unifique dentro da que apresentará na Reunião de Unificação.

### **Reunião de Unificação**

As diferentes questões encontradas individualmente são unificadas em uma reunião especial, cujo objetivo é incrementar a qualidade do produto. A sinergia obtida da discussão incrementa em 20% o número de questões documentadas na sessão.

É o Moderador que conduz esta reunião. O Moderador a prepara verificando que todos enviaram suas planilhas preenchidas. Se alguém não fez isso, a reunião é adiada, o que é um grave problema porque o projeto demora. Seria possível fazer sem o Inspetor atrasado, mas o precedente que seria colocado faria com que a inspeção perdesse a prioridade que deve ter.

Na reunião, o Moderador lembra a todos o objetivo da reunião, que consiste em assumir a responsabilidade conjunta pelos defeitos do produto, não a caça às bruxas dos autores. A equipe de inspeção empregou valiosos recursos na análise do produto para que este saia como um resultado coletivo, com a melhor qualidade possível. Para reforçar a consciência do trabalho em comum, o Moderador apresenta as estatísticas dos inspetores, tempo, esforço e quantidade de questões levantadas. Depois, ajudado por um projetor, vai percorrendo um a um os defeitos em sua planilha unificada. Ordenou-os por localização, de modo a ir do geral ao particular e do princípio ao fim. De cada questão dá a descrição, tipo e severidade. Depois de lida uma questão faz uma pausa para permitir comentários dos participantes. Os inspetores podem fazer perguntas entre si, por exemplo, se tal ou qual questão está relacionada com

outra, ou se é realmente tão severo quanto foi dito. Podem fazer perguntas ao autor sobre suas decisões, mas estas não são julgadas. Se surgem novos problemas, estes são registrados no momento, com toda a equipe, menos o Autor, concordando na redação da questão levantada. Estes podem ser problemas em outros produtos (melhorias da planilha, por exemplo, ou questões não detectadas previamente em outros documentos que foram oferecidos como apoio) inclusive outras questões levantadas na sessão. É aceito que a equipe faça comentários, ofereça sugestões e alternativas, que sejam feitas perguntas que sirvam para esclarecer e propostas de soluções, mas não são discutidas, só registradas. O Autor responde perguntas se forem dirigidas a ele, mas pode também perguntar em cima das questões levantadas, sem defender sua posição, para que sejam esclarecidas.

A discussão é limitada para que a sessão não dure mais de uma hora. Geralmente, o Moderador tenta concluir em noventa minutos para que seja viável que a reunião termine antes das duas horas. Se são completadas as duas horas, a inspeção é cancelada, o que dá um estímulo muito grande para que a equipe tente encerrar. Na Reunião de Unificação não se discutem nem se resolvem as questões, que são apenas anotadas. Talvez se esclareça o tipo ou a severidade, que pode mudar se quem levantou a questão concordar. Como já dissemos, pode-se pedir esclarecimento sobre por que isso é considerado uma questão ou um defeito.

### **Decisão sobre o Produto**

Quando já não há mais questões, a equipe decide sobre o produto. Isto significa que escolhe entre opções de: aceitá-lo completamente, não há retrabalho indicado; ou há retrabalho, mas basta que seja verificado pelo moderador; ou há retrabalho, mas é necessária outra inspeção por esta ou outra equipe; ou não há retrabalho, o produto é recusado porque é preciso refazê-lo inteiramente. Nos casos em que a decisão não seja definitiva (aprovar ou recusar) a equipe precisa fixar critérios de aprovação para o Autor, que o Moderador garantirá. O Moderador facilita esta discussão de fechamento e a conduz. O resultado completo é enviado com a decisão e as questões levantadas a todos os participantes, principalmente ao Autor.

### **Retrabalho e Conclusão**

Para cada questão levantada, o Autor se reúne com o Moderador para definir se a corrige, caso seja um defeito, ou a documenta e arquiva, se for uma melhoria pedida para esse ou outro produto e ainda não está disposto a realizá-la. Em todos os casos, anota sua decisão na planilha da inspeção que recebeu do Moderador. Se o Moderador o autoriza, pode mudar a severidade. Isto é bom porque, se há pessoas que se comportam com fanatismo<sup>105</sup> na reunião, é possível parar de discutir em um ponto e avançar. Se necessário completa-se, então, o trabalho, e o moderador verifica a completude do trabalho realizado, quer dizer, se foram corrigidos os defeitos severos, se o critério de aprovação fixado pelos inspetores foi alcançado ou superado. Podem então ocorrer algumas das seguintes alternativas: que o Moderador revise ou aprove o retrabalho, porque a disposição escolhida pela equipe permite; que a equipe revise partes selecionadas do produto, guiada por sugestões do Moderador, seja trabalhando individualmente (que é a opção preferida) ou em uma nova reunião. Também pode ocorrer que todo o produto seja reinspecionado quando os problemas forem substanciais e o produto for crítico para o projeto.

### **Informe Final**

A inspeção é completada com um Informe de Fechamento ou Informe Final de Inspeção. O Autor atualiza os itens da planilha de inspeções, deixando claro o status de cada item (pois algumas questões podem ter ficado sem resolver), a classificação final de severidade e tipo para cada item (pois podem ter mudado com anuência do Moderador), e envia a planilha atualizada ao Moderador. O Moderador ou o Autor se colocam de acordo para ver quem é que envia pedidos de mudança aos autores dos documentos de apoio relacionados com itens levantados na planilha. É o Moderador quem é responsável por emitir o informe final, que descreve a disposição final do produto de trabalho e inclui as estatísticas finais: severidade por tipo, esforço, etc. Estes números são consolidados e analisados especialmente para medir a efetividade e eficácia das inspeções.

---

<sup>105</sup> “Um fanático é uma pessoa que não está disposta a mudar de opinião nem mudar de tema”. Winston Churchill.

### 8.11 Fatores Críticos de Sucesso

O fato de que uma revisão seja denominada uma “Inspeção” não a transforma em uma atividade efetiva. É necessário planejá-la bem e levar em conta os fatores de sucesso. Ao planejar uma inspeção, escolha os participantes com cuidado, seguindo as pautas anotadas (ao final da seção Revisões, na página 121). Sempre é útil contar com um engenheiro de testes, porque é possível treiná-lo para encontrar problemas. Incluir pessoal novo permite mostrar a eles boas práticas. Assegure-se que a quantidade de material a inspecionar é limitada, para que tudo possa ser revisado. O Moderador deve exigir que os projetos forneçam tempo suficiente de preparação aos Inspectores para a revisão inicial. A duração da reunião de unificação deve se restringir a duas horas, em uma organização com problemas de disciplina as inspeções começaram a ser usadas para outro tipo de reuniões porque eram as únicas reuniões que os engenheiros respeitavam, convertendo-as em maratonas de decisões até que deixaram de assistir a todas elas. Por último, monitore e controle o processo. O que não se controla, se dispersa.

### 8.12 Fatores de Fracasso

As pautas para que uma inspeção seja bem-sucedida não acabam aqui. Há certas decisões que matam a inspeção. Se a alta gerência participar da reunião é muito provável que não se levantem questões para não ferir a reputação do Autor.

Se toneladas de material são revisadas, as questões levantadas não serão significativas e as inspeções perderão força rapidamente. Se a reunião de unificação se transforma em uma reunião de solução, degenerará rapidamente em um concurso de opiniões, deixe que o Autor seja o responsável por escolher as correções. A inspeção não é uma proposta de *design* por comitê, mas de qualidade por apoio da equipe. Se a reunião de unificação dura para sempre, as pessoas começarão a colocar objeções e desculpas para não participar. Se durante a reunião, o Moderador aceita comentários personalizados ou carregados de emoção (“Isto é uma porcaria de produto” ou “Quem escrever assim não pode trabalhar comigo”) as brigas serão moeda corrente e o propósito da inspeção, que é a que um grupo ajude uma pessoa a melhorar seu produto para bem de todos, perde-se de forma total.

### 8.13 Diferenças Entre Inspeções, *Walkthrough* e Revisões Estruturadas

Utilizando as inspeções como ‘gênero’, revisaremos os outros dois tipos mostrando somente as diferenças. Seguimos aqui o material de três livros que coincidem no essencial a respeito destes três tipos: [FREEDMAN, 1990], [GILB, 1994] e [EBENAU, 1994]. Primeiro começaremos com a mais fraca das três, em termos de capacidade de encontrar defeitos, o *walkthrough*. Há um livro de Yourdon<sup>106</sup> que descreve com detalhe o processo de *walkthrough*, mas que em nossa opinião descreve uma revisão estruturada, de modo que se estivermos interessados nesta última, essa é uma boa referência.

Ao contrário da inspeção, o *walkthrough* não exige um moderador. O próprio Autor organiza e facilita. A equipe não está limitada a um número máximo e não precisa ser convocada com antecipação alguma. O Autor pode reunir um grupo que dispõe de tempo e pedir que se apresentem para uma reunião em minutos ou horas. A duração da reunião também é flexível, e os papéis são designados, se for o caso, no momento de começar a reunião. Se a inspeção é uma tarefa premeditada, o *walkthrough* é uma atividade apenas formal. O propósito é obter retroalimentação rápida de um grupo de pessoas para melhorar a qualidade de um produto no momento. As anotações, captura de dados e resolução ficam a cargo do autor. É possível que não fiquem rastros desta atividade e que se considere que a falta de história não é um problema. Entre os dois tipos se encontra a revisão estruturada que, como já dissemos, está bem documentada na literatura. Nossa descrição coincide com a realizada no livro de Gilb já citado e em muitas partes com o livro de Yourdon, também citado.

Na revisão estruturada, quem convoca é o encarregado do projeto, ou seja, o líder técnico, líder de projeto ou gerente de projeto. O propósito é conhecer o estado de um produto. O encarregado designa o facilitador e este pode escolher se o autor participa na logística da atividade. Quer dizer, pode não levá-lo em conta para as decisões de composição da equipe e outras decisões. O facilitador cumpre um papel mais direto que o Moderador na inspeção

<sup>106</sup> YOURDON, Edward, 1989, *Structured Walkthroughs*. Prentice-Hall.

porque, apesar de seu nome, ele toma mais decisões que o Moderador. Em geral, segue-se o procedimento descrito para a inspeção, exceto porque a reunião de instrução é opcional, sendo substituída frequentemente por uma comunicação eletrônica para os participantes. As listas de verificação são semelhantes, mas é menos frequente que sejam diferentes para os distintos participantes segundo seus papéis, que tampouco são designados com a mesma frequência do que nas inspeções. Muda também a capacidade da equipe de decidir sobre o produto. Ao invés de escolher a disposição, a equipe recomenda ao encarregado do projeto um caminho a seguir, que este não está obrigado a considerar.

Uma das aplicações mais interessantes é seu uso combinado em projetos de certo risco. Para mais informação ver [BORIA, 2010].

Resumindo as diferenças em uma tabela, são as seguintes:

	<i>Walkthrough</i>	Revisões Estruturadas	Inspeções
<b>Objetivos</b>	Obter retroalimentação inicial sobre a maneira de avançar com o produto e corrigir defeitos, se existirem	Mostrar que o produto tem a qualidade esperada	Localizar todos os defeitos possíveis e corrigi-los, até alcançar um nível de qualidade prefixado
<b>Critério de Entrada</b>	Identifica-se a necessidade no calendário do projeto ou o autor ou outra pessoa decide	A gerência estima que o produto está pronto, foram publicados os critérios de qualidade, o autor confirma que se pode começar	O autor solicita a inspeção e confirma que o produto cumpre com os critérios para ser inspecionado
<b>Decisões</b>	As decisões são tomadas pelo autor	A equipe de inspeção informa e recomenda, a gerência decide	A equipe decide sobre o produto e ações futuras
<b>Fechamento das Questões</b>	Fora do escopo do procedimento	A gerência controla como parte do projeto	O moderador controla de acordo com os critérios da equipe
<b>Tamanho</b>	Duas ou mais pessoas, sem limite	Ao menos três e não mais de sete	Três pelo menos e não mais de sete
<b>Composição da Equipe</b>	Quem estiver disponível	Líderes técnicos e colegas do autor	Colegas do autor escolhidos com critérios previamente definidos
<b>Liderança</b>	Autor	Geralmente um líder técnico do projeto	Moderador capacitado
<b>Preparação</b>	Opcionalmente é distribuído o material com horas de antecipação	Há uma revisão individual prévia seguindo critérios estabelecidos na lista de verificação	Há uma revisão individual prévia seguindo critérios estabelecidos na lista de verificação e com materiais de apoio
<b>Apresentador</b>	Normalmente o autor	Autor ou leitor designado	Leitor designado ou ninguém
<b>Medidas Registradas</b>	Opcional, raramente é feito	Geralmente são registradas como em inspeções, mas não são requeridas, depende da empresa	Requeridas formalmente, esforço, tempo, tipo e número de questões
<b>Relatórios</b>	O autor produz de acordo com sua decisão	Relatório com a lista de defeitos, questões e ações realizadas	Relatório com detalhe do detectado, investido e realizado
<b>Captura de Dados</b>	Não é exigida	Não é exigida	Coleta, com fins estatísticos, de todos os dados cadastrais e esforço, tempo, custo, etc.

**Tabela 8.13: Comparação entre Revisões**

### 8.14 Usos Ágeis

As revisões, assim descritas, parecem mais adequadas em um projeto longo do que para serem úteis em um ambiente ágil. No entanto, vejamos como foram adaptadas por Marcela e os gêmeos na T<sup>2</sup>. Em primeiro lugar, a revisão escolhida pelos gêmeos é revisão contínua, chamada na literatura de programação em pares, à qual se podem juntar os dados que mudaram no processo: quando o módulo é crítico o programador com mais experiência atua de *coach* e captura dados em uma planilha adaptada usada habitualmente em inspeções (ver Tabela 8.14). Com esses dados é possível iniciar uma análise estatística dos defeitos detectados. Da mesma forma, Marcela e Jéssica adaptaram o método de revisões estruturadas para seu uso em *sprints*. Recordando o início da T<sup>2</sup>, quando se realizava o “leilão” de tarefas, quando uma equipe de trabalho termina um produto que mereça ser revisado, o que se define no jogo do planejamento do *Sprint*, ela convoca mediante a Intranet da T<sup>2</sup> (chamada festivamente ITT) as outras equipes para uma revisão. Cada equipe tem a obrigação moral de liberar um de seus membros, nas duas horas que se seguem ao chamado, que possa rever o material. No fim do dia, os revisores se reúnem com os autores para apresentarem suas questões. O *Scrum Master* da equipe que solicitou a revisão é o Moderador e tem as mesmas responsabilidades em relação ao relatório do que se fosse uma inspeção. Desse modo, as revisões são ágeis e produzem resultados. Os dados das revisões são armazenados em um repositório central para serem analisados na reunião mensal.

#### O QUE ACONTECEU ATÉ AGORA

66. Os problemas introduzidos em um *sprint* foram eliminados, mas os problemas legados escalaram.
67. Marcela verifica que uma melhoria do processo de engenharia de testes pode atender tanto à Validação quanto à Verificação.
68. Marcela, Jéssica e os gêmeos adaptaram o processo de revisões estruturadas a seus *sprints* e modificaram o procedimento de programação em pares para que produza evidência de revisão contínua.

Questões Levantadas Pela Inspeção				
Projeto	_____	Autor	_____	
Produto	_____	Moderador	_____	
Data de Entrada	_____	Revisores	_____	
Hora de Começo	_____		_____	
Hora de Fim	_____		_____	
Duração	_____		_____	
Localização (Página, Linha)	Descrição da Questão (Se o item foi levantado na reunião marque com *)	Tipo	Severidade	Comentários da Correção

Tabela 8.14: Planilha de Registro de Questões

### 8.15 Testes de Produto

Para continuar com a melhoria da percepção da qualidade do produto pelo cliente, os gêmeos lideram uma equipe de especialistas em testes. Jéssica incorporou entre as melhorias ao sistema de compensações a avaliação contínua, que consiste em descartar a avaliação anual de desempenho e, em troca, realizar avaliações do resultado de cada tarefa, inspirada pelo material de [CULBERT, 2010]. Como parte do sistema de recompensas, os mais destacados por seu desempenho recebem um diploma honorário de ‘Campeões’ e um curso de sua

escolha de treinamento anual pago pela T<sup>2</sup>. Como consequência, Evelina Kahn acabou a Campeã de Testes e como recompensa escolheu um curso de engenharia de processos e técnicas de testes, e voltou de lá disposta a aplicar o que aprendeu. Como a oficina está inspirada no livro de Denney, citado acima, a compatibilidade entre o que procuram os gêmeos e o que Evelina quer aplicar é total.

Revisando as ações para as questões de validação e verificação, Marcela já se ocupou de reforçar o seguimento do processo de testes, tanto para a aceitação do usuário quanto para os testes que o precedem, aumentando a frequência das auditorias e ajudando a prevenir não conformidades, de modo que este ponto é considerado fechado. Dá então a maior prioridade à necessidade de definir com o cliente e a equipe as estratégias de teste, os produtos a serem avaliados, o cronograma dos testes, os critérios de entrada e de aceitação e os ambientes de aplicação.

### 8.16 Critérios Relacionados com Testes

Entende-se por critério, segundo o dicionário Aurélio, “princípio que se toma como referência e que permite distinguir o verdadeiro do falso”. Em segunda acepção um critério é “ponderação, medida, equilíbrio, discernimento”. Em ambos, o significado considera o critério como um elemento de juízo que permite conhecer a verdade. Quando trabalhamos em um ambiente de *software*, é importante responder à pergunta: como saberemos que o produto está pronto para passar à próxima fase ou ser liberado ao usuário? Questões deste tipo são denominadas de Critérios de Aceitação.

Para cada tipo de teste, define-se:

1. O conjunto de casos de teste que deve ser usado
2. Os dados que devem ser utilizados cada vez que forem executados os casos de teste
3. Os ambientes nos quais devem ser executados os testes
4. Os testes de regressão incluídos neste tipo de testes
5. O número de defeitos tolerados por categoria de severidade
6. O mínimo de funcionalidades cobertas pelos testes (cobertura)
7. Objetivos de desempenho dos testes a serem alcançados (*performance*)
8. Objetivos específicos de volume (se aplicáveis)
9. Objetivos de confiabilidade (se aplicáveis)
10. Objetivos de usabilidade (se aplicáveis)

Outros dois critérios úteis servem para responder à pergunta: como sabemos que o produto está pronto para ser testado? Saber isto é importante porque, se começamos o teste com um produto imaturo, os defeitos se acumularão sem benefício, pois será preciso realizar muitas modificações. Depois é importante definir e confirmar que o critério de entrada para a fase de testes correspondente será cumprido.

Para cada tipo de teste, define-se:

Qual é o estado que precisa ter o produto para ingressar na fase (geralmente o critério de aceitação do produto na fase anterior, quer dizer, precisa ter sido provado em tal ambiente com tais casos que deram tal cobertura e com os resultados de não mais do que tantos defeitos segundo a severidade). Garantir que o critério de entrada seja cumprido implica em um bom investimento de recursos. Isto é especialmente importante nas primeiras etapas de teste, porque é aí onde são produzidos os maiores atrasos, quando os desenvolvedores entregam apressadamente produtos ainda sem terminar.

O seguinte critério a incluir é o que responde à pergunta: quando é preciso deixar de testar um produto porque tem muitos defeitos? Deixa-se de testar o produto e o devolvemos à fase de desenvolvimento para corrigi-lo quando não faz sentido continuar investindo recursos para testar código que fatalmente irá mudar de forma significativa. É bastante comum conectar este critério com o de aceitação que diz respeito à quantidade de defeitos por severidade. Claramente continua-se testando o produto enquanto o critério de aceitação relacionado com a densidade de defeitos (o 5 em nossa lista) continuar sendo cumprido e o critério de cobertura ainda não (o 6 em nossa lista).

### 8.17 Cobertura

A cobertura fornecida pelos testes é um aspecto sumamente importante na definição de critérios. Podemos ser mais específicos do que simplesmente falar da funcionalidade coberta. Nesse caso, o que se expressa como ‘cobertura’ é a porcentagem dos casos de uso que os casos de teste cobrem. Mas se lembrarmos a redação de um caso de uso (Tabela 8.8) é possível que existam fluxos alternativos. Por exemplo, vejamos o primeiro de todos os CDU, Registrar-se como Usuário do Sistema.

FLUXO NORMAL	1. O sistema espera um usuário novo mostrando a tela de entrada
	2. Um usuário escolhe entrar no sistema
	3. O sistema apresenta a tela de opções para usuários registrados ou novos
	4. O usuário escolhe a opção para usuários novos
	5. O sistema apresenta a tela de registro de dados
	6. O usuário introduz seus dados pessoais e confirma apertando “ENTER”
	7. O sistema armazena os dados do novo usuário em sua base de dados e passa ao CDU de Entrar
FLUXOS ALTERNATIVOS	7a. O sistema encontra um usuário com a mesma identidade (nome, tipo e número de documento), mas outros dados de cadastro (endereço e bancos)
	7b. O sistema consulta o usuário sobre os dados existentes, solicitando permissão para atualizá-los com os dados recém-ingressados
	7c. O usuário autoriza o sistema a realizar a atualização
	7d. O sistema atualiza os dados em sua base de dados e passa ao CDU de Entrar

**Tabela 8.15: Exemplo de Sequência Principal e Alternativa de um CDU**

É claro que um caso de teste que siga a sequência “feliz” (o usuário é totalmente novo) não cobre toda a funcionalidade do caso de uso, já que há pelo menos um fluxo alternativo quando o usuário se registrou anteriormente e em consequência uma regra de negócio (atualizar os dados de cadastro) que não é testada. Portanto, é importante definir o que se quer dizer com cobertura dos casos de teste. Obviamente, no nível mais alto, espera-se que haja pelo menos um caso de teste para cada caso de uso. Pode-se requerer que a cobertura de casos de uso seja 100% na maioria dos casos, mas nem sempre é assim, se o perfil operacional nos diz que alguns dos CDU são de muito baixa importância. Se alguns casos de uso são importantes, devemos saber quais coberturas das diferentes alternativas e exceções, de cada um, estamos cobrindo com nossos casos de teste.

Na realidade, a literatura clássica de *design* de casos de teste associa cobertura dos casos de teste com os métodos de caixa de cristal, ou caixa branca como eram chamados no princípio. Os métodos de *design* de testes são classificados em dois grandes tipos: o *design* de casos de teste omitindo o código que foi ou será construído, chamados “caixa preta” por sua semelhança com métodos de *design* de testes que ignoram o *design* dos componentes a testar, e caixa de cristal, ou “caixa branca” originalmente (até que alguém levou em consideração que o problema é distinguir entre opacidade e transparência, e não entre cores de caixas igualmente opacas). Os métodos de caixa de cristal são associados facilmente à cobertura, porque as categorias de cobertura são: sentenças (porcentagem de linhas de código exercitadas durante a execução do conjunto de dados de teste), condições (porcentagem das condições testadas em ambos os sentidos, quer dizer Falso/Verdadeiro, exercitadas durante a execução do conjunto de dados de teste), e caminhos (porcentagem de todos os caminhos possíveis exercitados durante a execução do conjunto dos dados de teste). As categorias são crescentes, nas quais a cobertura de caminhos implica na cobertura de condições, que por sua vez implica na cobertura de sentenças. Não posso atender todas as condições sem atender todas as sentenças A NÃO SER QUE HAJA CÓDIGO INALCANÇÁVEL, quer dizer, código que não pode ser executado em condições normais de uso. É o caso de porções de código que se separaram do conjunto colocando um salto (GOTO) à primeira sentença que vem depois da porção na sentença anterior à qual ele começa ou que estão condicionados por uma condição inalcançável.

Voltemos, agora, a Denney e seu livro *Use Case Levels of Test*. Nós o abandonamos com a construção do Perfil Operacional do sistema e a construção daqueles casos de uso que se justificavam. Agora já podemos elaborar sobre estes casos de uso que valem a pena, para analisar quais passos é preciso atender. Do perfil operacional já não podemos tirar mais nada, a não ser que refinemos os passos que são percorridos dentro de cada caso de uso e analisemos sua frequência relativa. É isso precisamente o que nos propõe Denney, convertendo primeiro o caso de uso em um diagrama de fluxo de controle. A tentativa de construção do diagrama permite ver que há caminhos sem definir no caso de uso. Por exemplo, O que acontece se o usuário não dá permissão para atualizar seus dados? No diagrama tomamos a decisão de finalizar sem atualizar.



Figura 8.11: Diagrama de Fluxo do Caso de Uso da Tabela 8.14

Um diagrama de fluxo de controle requer ter um único ponto de entrada e um único ponto de saída. Estes nos permitem fechar regiões no diagrama, marcadas como 1, 2 e 3, de baixo para cima. Isto é importante porque nos permite conhecer o número de casos de teste que necessitaremos gerar.

Começamos por fazer abstração do que acontece em um nó para poder mostrar, progressivamente, o método de construção de casos de teste mais a cobertura à medida que acrescentamos casos ao conjunto. Tomando novamente emprestado de Denney usaremos seu exemplo da Figura 8.12.

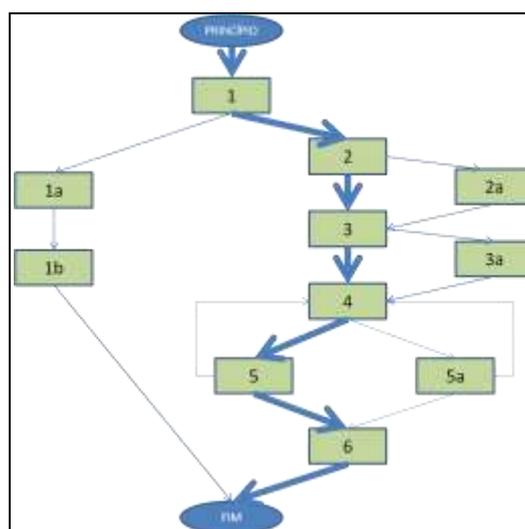


Figura 8.12: Diagrama de Fluxo de Controle com Funcionalidade Abstraiada

Dependendo se for um caso de uso de alta frequência, os caminhos que valem a pena atender com casos de teste serão diferentes. Imaginemos o pior caso, no qual a cobertura de todos os casos de uso for parte dos critérios de aceitação de uma fase de testes, mas o caso de uso

assim representado tem baixa frequência de utilização. Então tentaremos fazer o menos possível para cumprir com o critério, mas sem usar recursos demais. Quando é preciso escolher um único caso de teste para um caso de uso, geralmente, cria-se sobre a base de que o uso mais frequente deve ser no caso em que todas as condições acontecem de forma perfeita. É por isso que este caso é denominado “o caminho feliz”, porque não ocorrem exceções. Ainda para o caminho feliz, poderíamos ter uma mesma sequência de testes com diferentes dados, mas falaremos disto mais adiante. Um caso de teste que percorra os nós 1, 2, 3, 4, 5 e 6 da Figura 8.12 exercita o caminho feliz. Se dentro de cada nó não há caminhos alternativos, a execução de um nó implica a execução de todas as sentenças que formam parte desse nó no código, por isso a cobertura de nós no diagrama de fluxo de controle é equivalente à cobertura de sentenças e no código. Para conseguir cobertura de sentenças, precisamos de dois casos de teste. Um pode se aproveitar da existência de dois ciclos (entre 5 e 4 e entre 5a e 4) para que o caso de teste que percorre 1, 2, 2a, 3, 3a, 4, 5, 4, 5a e 6 dê cobertura aos ramos da direita do diagrama e depois com 1, 1a e 1b atender o ramo da esquerda.

Mas a cobertura de sentenças não garante que os testes sejam definitivos. Para ter a maior cobertura possível, a de caminhos, podemos partir do diagrama de fluxo de controle e sistematicamente produzir todos os caminhos. Começamos com o primeiro caminho, o Caminho Feliz, que como já vimos é 1, 2, 3, 4, 5 e 6 (Figura 8.13) e indo do último nó para cima vamos escolher um passo que não tenhamos explorado. Como já passamos pelo 5, a única opção é o 5a com seus dois passos, de 4 até ele e dele até o 6. Isto delimita a primeira zona do diagrama, a zona 1 (Figura 8.14). Repetindo o algoritmo chegamos ao nó 4, e agora escolhemos qualquer arco de entrada. Damos preferência aos que são numericamente posteriores, depois escolhemos o nó 5 e o arco que vai do 5 ao 4 (Figura 8.15). Fica delimitada a segunda zona, a 2. Repetindo o processo aparecem sucessivamente todas as zonas até a 7. São necessários 8 casos de teste para atender todos os caminhos.

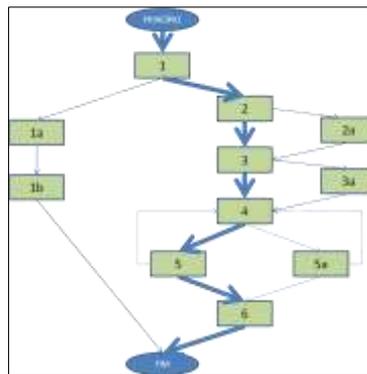


Figura 8.13 Caminho Feliz

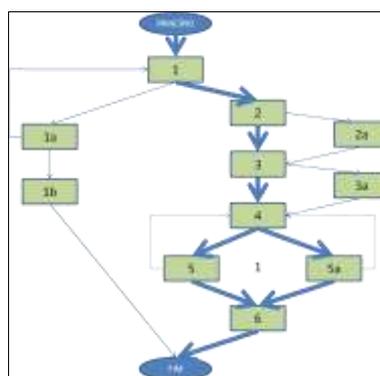


Figura 8.14 Primeira Área Fechada

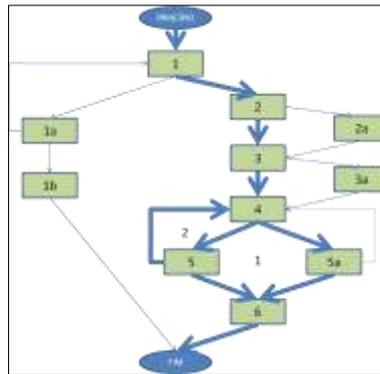


Figura 8.15 Segunda Área Fechada

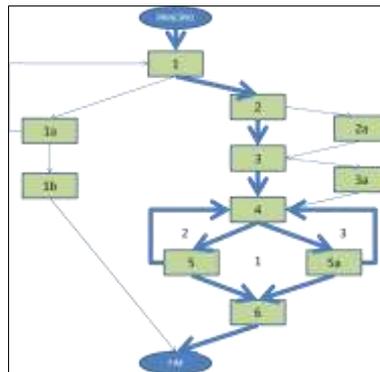


Figura 8.16 Terceira Área Fechada

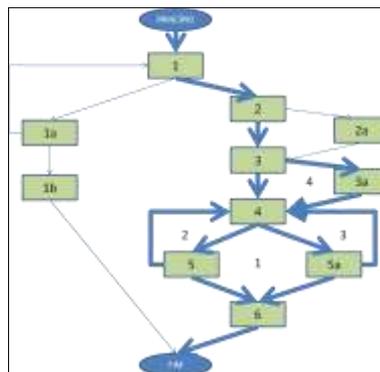


Figura 8.17 Quarta Área Fechada

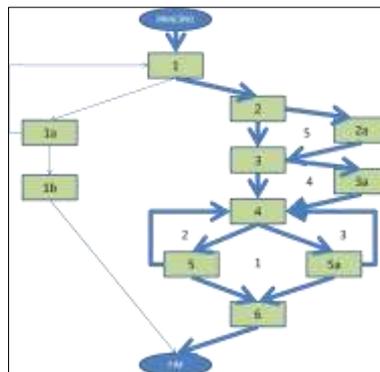


Figura 8.18 Quinta Área Fechada

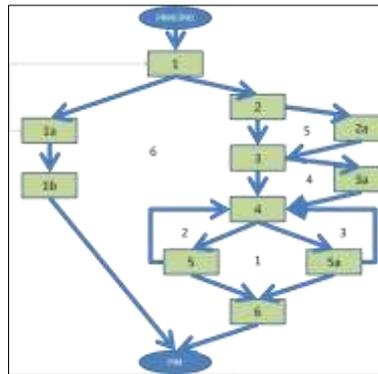


Figura 8.19 Sexta Área Fechada

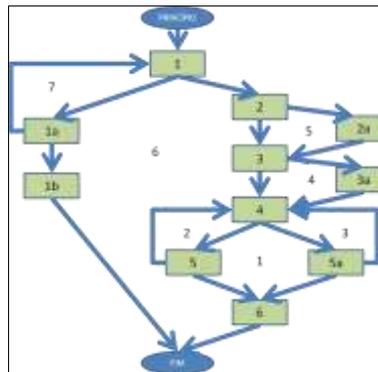


Figura 8.20 Todas as Áreas Fechadas

Só resta definir como se selecionam os caminhos que valem a pena serem detalhados com casos de teste. O método de Denney sugere... Adivinhar!, mas não sem fundamentos. Baseando-se na conhecida distribuição dos bens do matemático italiano Pareto, Denney propõe atribuir uma probabilidade de 80% ao passo mais provável e 20% ao outro, se são dois. Por exemplo, na Figura 8.13 o passo de saída (1, 2) teria 80% de probabilidade e o (1, 1a) teria 20%. Cada passo do caminho feliz tem 80% e as alternativas 20%. Quando há só um caminho de saída este tem um peso probabilístico de 100%, obviamente. Desta maneira as probabilidades são designadas a cada passo. Desprende-se então que a frequência de uso de um caminho é o produto de todos os passos desse caminho. O caminho feliz tem frequência  $(0,8 \times 0,8 \times 0,8 \times 0,8 \times 0,8) = 0,32768$ . O caminho 1 -> 1a -> 1b tem frequência 0,16. (Figura 8.21).

Marcela e as equipes técnicas investigam a possibilidade de utilizar critérios de cobertura fortes para garantir que os testes deem uma alta segurança de que os defeitos remanescentes são muito poucos e dentro do objetivo de processo estabelecido pelo comitê de gestão. Finalmente, decide-se que os critérios serão estabelecidos pelo dono do produto em conjunto com a equipe durante o jogo do planejamento, como parte da atividade de estabelecer os requisitos não funcionais do *sprint*. Mas quando um caminho dentro de um caso de uso utilizado com alta frequência tiver risco alto, o caso de teste correspondente deve ser incluído entre os casos a criar e implementar para o teste de integração contínua do *sprint*. Como parte do *design*, concorda-se em seguir os passos descritos por Denney. Para o *sprint* de estabilização serão fixados critérios especiais relacionados aos objetivos de negócio da empresa a respeito do produto em questão.

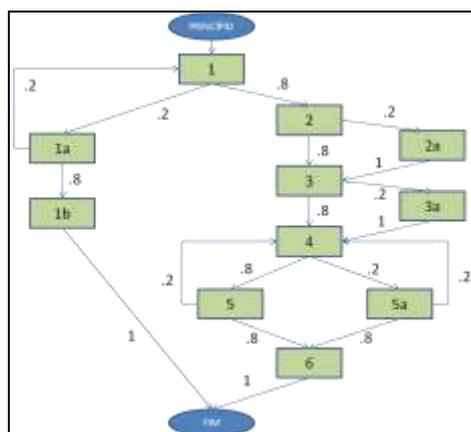


Figura 8.21: Probabilidade de Cada Transição do Grafo

#### O QUE ACONTECEU ATÉ AGORA

69. Jéssica incorporou entre as melhorias ao sistema de compensações a avaliação contínua.
70. Os gêmeos encabeçam um grupo de especialistas para modificar e melhorar as técnicas de testes de produto.
71. Evelina terminou Campeã de testes e volta de um curso de engenharia de testes disposta a aplicar o que aprendeu recentemente.
72. São adotadas as técnicas de *design* de casos de teste guiadas por riscos e frequência de uso.
73. Os critérios de cobertura, com diferente ênfases, são usados na fixação de objetivos para cada *sprint*.
74. O *Sprint* final de estabilização receberá seus próprios critérios de cobertura.

Como último passo na melhoria de processos da engenharia de testes, Marcela e Jéssica exploram a lacuna existente entre os procedimentos em uso pela T<sup>2</sup> e os resultados esperados do MR-MPS-SW para os processos de Verificação e Validação.

Resultados Esperados de VER no MPS	Atividades Internas na Tahini-Tahini
<b>VER1 Os produtos de trabalho a serem verificados são identificados;</b>	acordar com a equipe a estratégia de verificação, os produtos a avaliar e com quais métodos, o cronograma das diferentes avaliações, os critérios de entrada, descontinuidade e aceitação (garantindo que a cobertura mínima seja uma delas) e os ambientes de teste, quando forem aplicáveis
<b>VER2 Uma estratégia de verificação é desenvolvida e implementada, estabelecendo o cronograma, revisores envolvidos, métodos para verificação e qualquer material a ser utilizado na verificação;</b>	
<b>VER3 Critérios e procedimentos para verificação dos produtos de trabalho a serem verificados são identificados e um ambiente para verificação é estabelecido;</b>	
<b>VER4 Atividades de verificação, incluindo testes e revisões em pares, são executadas;</b>	aplicar com todo rigor auditorias de processo e produto até que seja fixada a conduta de teste sistemática
<b>VER5 Os defeitos são identificados e registrado;</b>	automatizar todos os testes

<b>VER6 Os resultados de atividades de verificação são analisados e postos à disposição das partes interessadas.</b>	os dados obtidos dos testes e das inspeções, walkthroughs e revisões técnicas, devem ser analisados e discutidos na reunião mensal de portfólio
--	---

Tabela 8.16 Resultados Esperados de VERIFICAÇÃO e Atividades Internas em T<sup>2</sup>

Resultados Esperados de VAL no MPS	Atividades Internas na Tahini-Tahini
<b>VAL1 Os produtos de trabalho a serem validados são identificados;</b>	acordar com o cliente e a equipe a estratégia de validação, os produtos a serem avaliados, o cronograma da validação, os critérios de entrada e aceitação e os ambientes de aplicação
<b>VAL2 Uma estratégia de validação é desenvolvida e implementada, estabelecendo um cronograma, participantes envolvidos, métodos para validação e qualquer material a ser utilizado na validação;</b>	
<b>VAL3 Critérios e procedimentos para a validação dos produtos de trabalho a serem validados são identificados e um ambiente para validação é estabelecido;</b>	
<b>VAL4 As atividades de validação são executadas para garantir que o produto esteja pronto para seu uso no ambiente operacional previsto;</b>	
<b>VAL5 Os problemas são identificados e registrados;</b>	reforçar o seguimento do processo de aceitação do usuário, aumentando a frequência das auditorias e ajudando a prevenir não conformidades, assegurando que os critérios de aceitação sejam cumpridos, baseados em documentação fidedigna
<b>VAL6 Os resultados das atividades de validação são analisados e postos à disposição das partes interessadas;</b>	
<b>VAL7 As evidências de que os produtos de software desenvolvidos estão prontos para seu uso previsto são fornecidas.</b>	

Tabela 8.17: Resultados Esperados de VALIDAÇÃO e Atividades Internas em T<sup>2</sup>

Os resultados são muito bons, o que a direção de T<sup>2</sup> recebe com muita alegria. A pergunta que se fazem é se vale a pena fazer uma avaliação para o Nível D ou esperar até ter o C implementado. Marcela lembra que ainda não revisaram todos os processos do nível D, falta ocupar-se de Integração do Produto – ITP e de Projeto e Construção do Produto – PCP.

### 8.18 Projeto e Construção

Seguindo a linha de atuação que surgiu dos requisitos, a construção de casos de teste, pode-se ir até ITP considerando que a integração tem um forte componente de testes, ou até PCP, porque a equipe escolheu *Test Driven Design* como técnica de *design*. Este último os guia até revisar a lacuna com PCP. Os resultados de PCP estão listados na :

#### Projeto e Construção do Produto (PCP)

**PCP1 Alternativas de solução e critérios de seleção são desenvolvidos para atender aos requisitos definidos de produto e componentes de produto;**

PCP2	Soluções são selecionadas para o produto ou componentes do produto, com base em cenários definidos e em critérios identificados;
PCP3	O produto e/ou componente do produto é desenhado e documentado;
PCP4	As interfaces entre os componentes do produto são desenhadas tomando como base critérios pré-definidos;
PCP5	Uma análise dos componentes do produto é conduzida para decidir sobre sua construção, compra ou reuso;
PCP6	Os componentes do produto são implementados e verificados de acordo com o que foi desenhado;
PCP7	A documentação é identificada, desenvolvida e posta à disposição de acordo com os padrões estabelecidos;
PCP8	A documentação é mantida de acordo com critérios definidos.

Tabela 8.18 Processo PROJETO E CONSTRUÇÃO DO PRODUTO [SOFTEX, 2012a ]

Revisando os materiais derivados das reuniões de retrospectiva são analisados os problemas que se vinculam ao *design* e construção dos produtos. Reflete-se esta análise na Tabela 8.19.

<b>Design (Projeto) e Construção do Produto – PCP</b>	
<b>risco ou problema:</b>	<b>mitigação:</b>
nos dedicamos a explorar a primeira opção de <i>design</i> e se não há razão para descartá-la, a implementamos sem considerar os objetivos do <i>design</i> e buscar alternativas melhores, arriscando sub-otimizá-lo	desenvolver critérios universais para iniciar uma análise de alternativas de <i>design</i> baseados nos objetivos de negócio da T <sup>2</sup> e do projeto
o <i>sprint</i> de estabilização está tendo problemas porque a equipe não reconhece algumas decisões de <i>design</i> que impactam nos testes	documentar o <i>design</i> , sobretudo o porquê da seleção de componentes, apoiado nos requisitos, sobretudo os não funcionais
a integração, frequentemente, fracassa na primeira tentativa, fazendo perder um dia de testes a cada vez	realizar uma análise de causas para identificar ações
tentamos reinventar a roda em várias ocasiões	aplicar a análise de reuso, ampliando iniciativas de reuso com componentes do mercado
em algumas ocasiões, a equipe se desviou significativamente das decisões de <i>design</i> , em favor de melhorias hipotéticas, aumentando a probabilidade de precisar refazer os casos de teste no último momento.	reforçar o sentimento de YAGNI <sup>107</sup> utilizando só testes que foram aprovados pela equipe no momento de aprovar o plano de <i>sprint</i>
em casos particulares entregou-se a documentação incompleta ou em formato diferente do decidido	acrescentar uma planilha de revisão de documentação para ser utilizada para afirmação da qualidade antes do fechamento do <i>sprint</i>
	refazer a revisão da documentação no <i>sprint</i> de estabilização

Tabela 8.19: Problemas de *Design*

Comparando as soluções com os resultados esperados do processo, Marcela e Ana vislumbram grandes oportunidades de resolver o problema detectado em cada caso e, ao mesmo tempo, fazer uma implementação que alcance esses resultados esperados. Algumas ações são muito fáceis de implementar, por exemplo, ampliar a análise de reuso que já

<sup>107</sup> *You Ain't Gonna Need It* (Você não vai precisar).

vimos<sup>108</sup> para incluir entre as opções componentes que podem ser adquiridos fora da T<sup>2</sup>. Isto dá frutos de imediato porque Mariano mantém um caderno de tecnologia no mesmo wiki em que são guardadas as informações de componentes reutilizáveis. O motivo pelo qual faz isso é porque quer ter muito claro quais são os produtos que competem com as linhas de produto da T<sup>2</sup> e orientá-la para as necessidades do mercado. Desse modo, as equipes não precisam mudar nem uma linha de seus processos, pois uma pequena mudança no método de busca na wiki dá o resultado esperado. Mas a investigação sobre reuso trouxe um presente inesperado: pode-se adaptar o método de análise de decisões para reuso, para que seja um método de análise para *design* em geral, de modo que, ao começar o *design*, já se esteja pensando em alternativas. Desse modo, ataca-se a raiz do problema de *design*, o primeiro dos pontos. Modificada a análise que deve ocorrer no jogo do planejamento, fica assim:

1. Enunciado de objetivos, quer dizer, para que se realiza o *design* do produto do *sprint*. Alguns exemplos são encurtar prazos sem perda de qualidade, permitir manter a duração do *sprint* desenvolvendo mais pontos de usuário, diminuir custos, etc.
2. Escolha de atributos desejáveis, onde a equipe discute, a partir dos requisitos, quais atributos deve ter o *design*, por exemplo, deve ser integrável facilmente, compatível com o *design* atual, fácil de testar, executar muito rápido, etc.
3. Identificação de soluções candidatas, que é realizado automaticamente usando o algoritmo neural baseado nos atributos escolhidos para componentes existentes ou no mercado. No caso de não existir ou de considerar a equipe necessária, são descritos ao menos três soluções alternativas enfatizando o impacto do risco na identificação.
4. Avaliação de candidatos, que é realizado por testes *ad-hoc*, que são criados em relação aos atributos escolhidos, e a história do componente, quando existente, ou por métodos menos formais, quando se trata de soluções originais.
5. Análise de opções, realizado seguindo um método pré-estabelecido que utiliza uma matriz de Pugh como a que já foi visto no Capítulo 4. Uma das opções é “nunca utilizar um componente reutilizável”.
6. Seleção de alternativas, realizado também seguindo o método anterior.
7. Adoção ou Design de componente, que realiza o ajuste do componente, de acordo com as necessidades da equipe, se aplicável. Pode ser que, ao chegar a este ponto, o resultado da avaliação de alternativas seja totalmente negativo e seja preciso revisar novamente o *design*.
8. Avaliação do resultado, que compara os resultados obtidos com os objetivos enunciados no primeiro passo para: continuar montando a história do componente e acrescentar os conhecimentos adquiridos, quando se trata de reuso, ou capturar o aprendido na wiki, quando se trata de um *design* novo. Se aplicável, registrar o componente como útil para reuso.

**Tabela 8.20: Análise de Opções sobre Design**

Deste modo, estende-se a aplicação de reuso a componentes ainda não construídos ou já existentes no mercado.

Realizada uma análise de causas sobre os problemas do fracasso da integração contínua na primeira tentativa, determina-se que o problema é que houve mudanças nas APIs sem que todos os interessados tivessem sido comunicados. A solução é que as APIs fiquem sob controle do *Scrum Master* depois de aprovadas e que todas as mudanças sejam comunicadas. O uso de ferramentas de controle de configuração faz com que esta pequena mudança seja fácil de implementar. Apesar de tudo, os primeiros *Sprints* utilizando a mudança ainda mostram traços desse problema, por isso o grupo de Qualidade, que agora depende de Jéssica, volta a realizar auditorias iniciais para garantir que as APIs e as interfaces de usuário estejam sob controle de mudança e designadas ao *Scrum Master*. Apesar de isto violar de certo modo os

<sup>108</sup> Capítulo 7, Figura 7.2: Análise de Opções sobre Reuso, página 99.

princípios ágeis, a proposta é bem recebida pelas equipes: poucas coisas são mais sentidas do que chegar pela manhã a um escritório vazio e descobrir que os resultados da noite anterior são nulos.

Já no tema das auditorias de qualidade, incorpora-se a sugestão das retrospectivas de acrescentar um item de controle que permita conhecer o estado da documentação decidida com o usuário em cada caso. Todas as auditorias pré-demo as incluem, assim como as de entrada e saída ao *Sprint* de estabilização.

Voltando ao tema de *design* de testes, como os testes se desenvolvem de modo sistemático agora, torna-se mais desejável utilizá-los como elemento de *design*. As equipes ratificam sua política de usar os testes criados por engenheiros de teste para desenvolver o produto. Neste ponto, Marcela considera que os itens principais surgidos das retrospectivas que se relacionam com o *Design* foram cobertos. Para verificar isso, constrói a Tabela 8.21.

<b>Resultados Esperados de PCP no MPS</b>	<b>Cobertura na Tahini-Tahini</b>
<b>PCP1 Alternativas de solução e critérios de seleção são desenvolvidos para atender aos requisitos definidos de produto e componentes de produto;</b>	desenvolver critérios universais para iniciar uma análise de alternativas de design baseados nos objetivos de negócio da T2 e do projeto
<b>PCP2 Soluções são selecionadas para o produto ou componentes de produto, com base em cenários definidos e em critérios identificados;</b>	
<b>PCP3 O produto e/ou componente do produto é criado e documentado;</b>	documentar o design, sobretudo o porquê da seleção de componentes, apoiado nos requisitos, sobretudo os não funcionais
<b>PCP4 As interfaces entre os componentes do produto são criadas tomando como base critérios predefinidos;</b>	controlar as interfaces de todo tipo para que suas modificações se propaguem aos interessados
<b>PCP5 Uma análise dos componentes do produto é conduzida para decidir sobre sua construção, compra ou reuso;</b>	aplicar a análise de reuso, ampliando-o com componentes do mercado
<b>PCP6 Os componentes do produto são implementados e verificados de acordo com o que foi criado;</b>	reforçar o sentimento de YAGNI utilizando somente testes que foram aprovados pela equipe no momento de aprovar o plano de sprint
<b>PCP7 A documentação é identificada, desenvolvida e disponibilizada de acordo com os padrões estabelecidos;</b>	acrescentar uma planilha de revisão da documentação para ser utilizada para afirmação da qualidade antes do fechamento do sprint
<b>PCP8 A documentação é mantida de acordo com critérios definidos.</b>	refazer a revisão da documentação no sprint de estabilização

Tabela 8.21: Cobertura de Resultados Esperados de PCP

### 8.19 Integração

As retrospectivas também deixaram lições sobre os procedimentos de integração. Nelas foram detectados vários problemas relacionados, e da mesma forma, foram propostas soluções. Algumas destas soluções propostas estão vinculadas a procedimentos já implementados para engenharia de testes, de maneira que acabam sendo muito fáceis de implementar.

<b>risco ou problema:</b>	<b>mitigação:</b>
a integração contínua está sendo usada muito limitadamente, sem considerar os componentes já desenvolvidos, o que pode trazer consequências no <i>sprint</i> de estabilização (e já trouxe em alguns casos)	ampliar o uso da integração contínua para conseguir que os testes de regressão ocorram com certa frequência, visando minimizar o esforço de estabilização no <i>sprint</i> final
quando o produto começa a estabilizar-se e a crescer, o ambiente de desenvolvimento é muito lento e a integração cria obstáculos ao desenvolvimento, resultando em horas perdidas.	garantir que os ambientes de integração definidos para os projetos estejam de acordo com suas necessidades, a partir de uma definição básica padrão e critérios de ajuste
a integração pode fracassar porque as interfaces não são compatíveis, o que pode levar a perdas de eficácia no uso dos recursos	assegurar a compatibilidade das interfaces mediante uma mini-inspeção antes de subir um módulo para ser integrado
as mudanças nas interfaces têm efeitos negativos nos testes de regressão, e nem sempre são consideradas na análise do impacto	dedicar parte do procedimento da análise de impacto à análise de interfaces
cedendo a pressões próprias, os programadores costumam subir código defeituoso ao ambiente de integração	utilizar inspeções breves para analisar os produtos e auditar que os testes unitários foram realizados e os critérios foram cobertos
a integração é realizada espontaneamente sem seguir muitas pautas, salvo o que está nos scripts da ferramenta	orientar as equipes para que gerem suas próprias regras sempre que incluam as normas organizacionais ou recebam dispensa para não fazer isso
no passado, acreditamos ter aprovado uma integração quando na realidade não havia sido executada ainda	os dados da integração devem ser cuidadosamente inseridos na ferramenta e os resultados devem ser analisados como parte do <i>scrum</i> diário
alguns casos da base de testes de regressão deixaram de ser úteis e novos casos que deveriam ter sido incluídos não foram	acrescentar à descrição do papel de engenheiro de testes o procedimento de manutenção da base de regressão
nem sempre se desenvolveu a documentação prevista por contrato ao mesmo tempo do produto em si, o que frequentemente provocou sufoco no final	orientar os <i>scrum masters</i> para que revisem o <i>backlog</i> e questionem a falta de tarefas relacionadas com a documentação contratualmente obrigatória, quando for pertinente

**Tabela 8.22: Ações Relacionadas com Integração, Derivadas de Retrospectivas**

Como parte do procedimento habitual de integração contínua, foi acrescentada a execução de testes de regressão todas as sextas-feiras ao sair para o fim de semana, a fim de minimizar o esforço de estabilização no *sprint* final.

Ao contar com ambientes de teste bem definidos para os projetos de acordo com suas necessidades, incorporou-se naturalmente o ambiente de integração, a partir de uma definição básica padrão e critérios de ajuste que integram a BiPro.

Quando se sobe um programa à *baseline* para sua integração contínua, o colega que realizou os testes correspondentes deve assegurar a compatibilidade das interfaces mediante uma miniauditoria antes de subi-lo.

No jogo do planejamento, dedica-se agora parte do tempo à análise de interfaces.

São realizadas miniauditorias de configuração e processos para assegurar que foram realizados os testes unitários e foram cobertos os critérios de aprovação dos módulos antes que possam ser integrados.

As equipes adaptam as normas organizacionais para realizar procedimentos de integração ou recebem dispensa para não precisar realizá-los sob justificativa bem documentada.

Os dados da integração surgem da ferramenta e os resultados são analisados à primeira hora, como parte do *scrum* diário.

O papel de engenheiro de testes tem agora documentada sua responsabilidade pelo procedimento de manutenção da base de testes de regressão.

É tarefa dos *Scrum Master* revisar diariamente o *backlog* e questionar a possível falta de tarefas relacionadas com a documentação obrigatória quando for pertinente pelo acordo com o cliente.

Com todas estas melhorias, o Comitê de Gestão espera obter benefícios significativos, manifestados nos objetivos de negócio.

#### O QUE ACONTECEU ATÉ AGORA

75. As mudanças introduzidas permitem supor que os resultados esperados de VER e VAL estão cobertos.
76. A direção está em dúvida se deve esperar para implementar o Nível C ou realizar antes uma avaliação do Nível D.
77. Os resultados esperados de PCP e ITP são revisados, junto com ações resultantes de retrospectivas, e são também implementados aproveitando mudanças anteriores.

## CAPÍTULO 9 - AMPLIANDO A CAPACIDADE DE DECISÃO (NÍVEL C do MPS-SW)

### 9.1 Uma Gestão Ambiciosa

Até aqui a gestão da T<sup>2</sup> foi, nos dois anos e um mês que a acompanhamos, um processo sólido, baseado em decisões estruturadas e com um profundo sentido das opções possíveis e os riscos que implicavam. O sucesso obtido com a implementação dos ajustes à engenharia, realizados com ênfase nos resultados esperados dos processos de Nível D do modelo MPS, convenceram Marcela que a fonte onde bebem é boa, e essa é a mensagem que ela transmite à direção. Na reunião mensal de dezembro, a mais importante do ano porque coincide com o fechamento do exercício anual, Marcela faz seu anúncio oficial: a já não mais tão pequena Tahini-Tahini vai realizar uma avaliação conjunta do Nível C do MPS-SW e do Nível 3 (Definido) do CMMI-DEV no ano que vem. Ana, que já sabia do projeto, apoia animada. Alfredo olha para uma e para outra sem saber o que dizer. Os gêmeos perguntam se os três novos projetos que estão realizando, assim como as extensões aos sistemas de farmácia e hospital que estão sendo produzidas a cada semestre, poderão receber o apoio dos recursos que necessitam ou se a demanda pelas avaliações vai dificultar o sucesso deles. Marcela se dirige a Mariano, interrogando-o com o olhar.

*“Não há dúvida”, diz Mariano, “as vantagens de entrar no mercado internacional com produtos inovadores como os nossos é incomparável. Podemos atrasar projetos e pagar as consequências com juros se formos avaliados no Nível 3 e isso nos trouxe só um cliente do Hemisfério Norte”.*

*“Mas isso não tem por que ser assim”, diz Jéssica, “porque podemos alcançar esse cobiçado prêmio sem pararmos os projetos nem um segundo. Até chegar o momento da avaliação, precisamos somente de fundos para dois recursos, um funcionário em tempo integral e o outro, Máximo, para nos conduzir nas decisões mais importantes”.*

### 9.2 Líder em Ação

*“Managers are people who do things right and leaders are people who do the right thing”<sup>109</sup>*

Marcela apresenta um quadro comparativo dos lucros ao usarem uma ferramenta nova (mais uma) que Jéssica introduziu, a Árvore de Decisão. Um dos caminhos da árvore representa o *status quo*, com os investimentos normais e os lucros esperados. Outro dos ramos mostra os investimentos a serem realizados para alcançar o nível C e os benefícios esperados. Estes são derivados de um exercício que realizaram Ana, Mariano e Cláudio, analisando a expansão de mercado possível com base nas melhorias de qualidade, assim como os benefícios internos imediatos vindos da diminuição dos custos. Um terceiro ramo mostra os custos e benefícios de alcançar tanto o nível C do MPS quanto o nível Definido do CMMI. Nesse ramo também se acrescenta a expansão do mercado do hemisfério norte.

Mariano intervém para colocar que já iniciaram contato com uma importante farmacêutica que quer levar o produto da T<sup>2</sup> ao mercado do Canadá. Obter o Nível de Maturidade 3, segundo Mariano, terminaria de convencê-los e fecharia o negócio. A votação é unânime a favor da avaliação conjunta. A Tabela 9.1 mostra os cálculos realizados pela árvore da Figura 9.1.

Marcela elabora mais sobre os três aspectos que são muito importantes para o nível C do MR-MPS-SW: Formalizar a gestão de riscos, a gestão das decisões e desenvolver métodos para que o reuso seja sistemático.

*“Os três pontos”, afirma Marcela, “já estão em nossos processos. É só uma questão de detalhar ainda mais, colocá-los no papel e documentar seu uso de maneira mais explícita”.*

Os gêmeos ficam contra o expressão usada por Marcela: Alberto a acusa de ser “ecologicamente incorreta” por falar em papel. Armando explica que em um mundo sem árvores, o oxigênio passa a ser um bem dos muito ricos. O ambiente fica mais tranquilo e a reunião termina com um brinde: os Galarraga fazem aniversário e trouxeram para a mesa de trabalho *Champagne Mumm Cordon Rouge*, que acaba em poucos minutos.

<sup>109</sup> [BENNIS, 1997], *Learning to Lead*

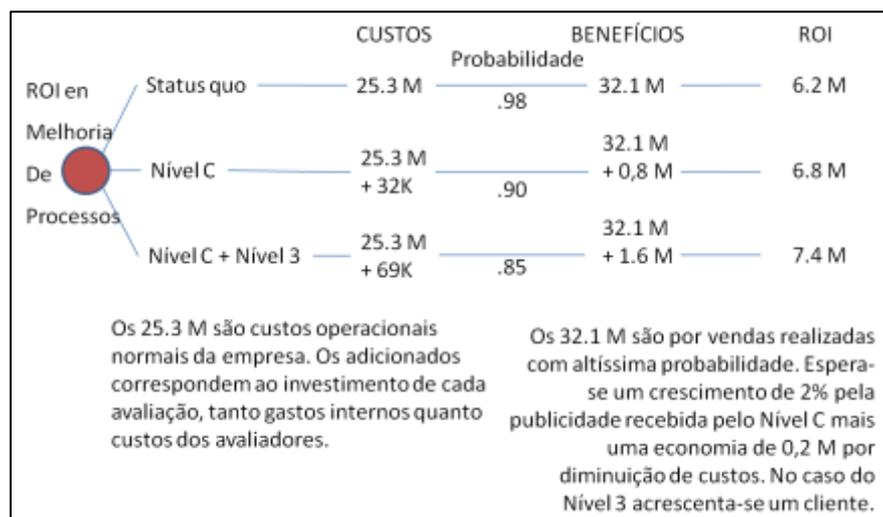


Figura 9.1: Árvore de Decisão

	Status Quo		Nível C		SCAMPI <sup>110</sup>	
	Entradas	Gastos	Entradas	Gastos	Entradas	Gastos
	32,1	25,3	0,8	0,032	1,6	0,069
Probabilidade	0,98	1	0,9	1	0,85	1
Expectativa	31,458	25,3	0,72	0,032	1,36	0,069
ROI	6,158		0,688		1,291	

Tabela 9.1: Tabela de Pagamentos da Árvore de Decisão

### 9.3 Visão Estratégica dos Riscos

A T<sup>2</sup>, há mais de um ano, começou a aproveitar os conhecimentos adquiridos pelo pessoal, criando ativos em seus arquivos. A biblioteca de processos é continuamente atualizada e melhorada. Os engenheiros, que já passam de cento e vinte, formam naturalmente grupos de interesse em cada uma das disciplinas. As anomalias na entrega dos produtos e os atrasos nos projetos estão começando a serem vistos como exceções e não como regra. Quando aparecem problemas, eles são rapidamente identificados e resolvidos. Os engenheiros estão alertas, detectando os padrões que permitem se anteciparem aos problemas e introduziram um método simples para capturar estes padrões em uma tabela. Sobre essa base, Marcela elabora um procedimento de riscos que é compatível com o atual e cobre os resultados esperados do processo Gerência de Riscos do MPS-SW.

Consciente dos desvios e excessos no uso da palavra “risco”, como ilustra seu abuso em várias tabelas que já vimos no Capítulo 8<sup>111</sup>, Marcela define com precisão o significado e o uso do termo dentro da T<sup>2</sup>. Seguindo o uso do IEEE, do MPS e do CMMI, “Risco” na T<sup>2</sup> é um problema potencial, quer dizer, algo que ainda não ocorreu. Existe risco em todas as atividades, porque o futuro é incerto. É atribuída a Niels Bohr<sup>112</sup> a frase: “Nada é mais difícil de prever que o futuro” e ironias à parte, nada é mais certo de acordo com nosso conhecimento científico. Marcela escolhe a forma mais ampla de definir um risco. Seu uso na T<sup>2</sup> parte da definição de problema. Habitualmente, reconhecemos um problema como uma situação incômoda, chata, um obstáculo em nossa vida ou no processo de negócios. É, definitivamente, algo ruim. Se olharmos o problema como um problema intelectual, algo que nos desafia a

<sup>110</sup> SCAMPI é a sigla de *Standard CMMI Appraisal Method for Process Improvement*, o método padrão e oficial de avaliar a maturidade de processos em relação ao modelo CMMI.

<sup>111</sup> Nas tabelas derivadas das retrospectivas, a primeira coluna é intitulada “riscos”, apesar de que são problemas já detectados ou incidentes. A segunda se intitula “mitigação” embora na realidade seja um plano de ação para resolver o problema.

<sup>112</sup> <http://www.aps.org/policy/reports/popa-reports/energy/modeling.cfm>

encontrar uma melhoria mesmo quando a situação não é tão ruim, teremos uma melhor definição de problema. Para classificá-lo de algum modo que ajude a esclarecer, podemos falar de: problemas de desempenho quando a situação atual é pior do que a esperada; problemas de melhoria quando a situação atual não é tão boa quanto desejamos; e problemas de manutenção quando a situação atual deve ser mantida. Nesse caso, há riscos de três tipos, o primeiro dos tipos coincide com a definição habitual: Algo ruim que pode acontecer. A pergunta que é feita por aquele que tenta identificá-los é “o que pode nos impedir de obter o resultado esperado?” O segundo tipo é o risco de perder uma oportunidade. Está relacionado com a inovação e o chamaremos “risco de oportunidade”. A pergunta que se faz para tentar identificá-lo é “o que estamos deixando de considerar que pode nos dar uma vantagem competitiva?” Está claro que esta segunda pergunta será muito menos frequente do que a primeira, mas de qualquer modo, Marcela propõe estender o escopo a todos os aspectos do negócio, por exemplo as contratações, os fornecedores, os salários, a tecnologia, a localização dos escritórios e até os métodos de avaliação<sup>113</sup>. Uma modificação na política de qualidade da T<sup>2</sup> define o escopo da aplicação da gestão de riscos com precisão.

O que Marcela propõe é que todos os que trabalham na T<sup>2</sup>, de Alfredo ao engenheiro mais recentemente incorporado, vejam seu trabalho como uma série infinita de tomada de decisões, cada uma delas com suas consequências. Ao colocar opções para essas decisões, a pessoa precisa perguntar-se, ou perguntar, no caso daqueles que possuem capacidades de direção: qual é o risco? Marcela pensa em uma organização consciente de seus riscos, não em uma que evita os riscos em todos os casos. Ser consciente de um risco é compreender o impacto que ele traz caso se materialize, quer dizer, deixar de ser um risco para se converter em um problema. A probabilidade de que um risco se materialize é importante para fazer uma avaliação de como atuar a respeito. Por exemplo, não se pode descartar que, antes de finalizar o ano, um meteorito destrua a cidade em que moramos e trabalharmos, com as óbvias nefastas consequências para os indivíduos que moram nela e nossos clientes, mesmo aqueles que vivem em outros países. Mas o evento tem baixíssima probabilidade de ocorrer, por isso preferimos ignorá-lo. Esta é uma das possíveis respostas frente a um risco: assumi-lo e continuar com o plano. Há dois motivos pelos quais se assume um risco: ou a probabilidade é tão baixa que qualquer investimento em mitigá-lo ou planejar contingências é muito caro, ou estes gastos são tão altos que não justificam sua realização, porque o remédio é pior do que a doença. Se um risco é muito grande, talvez não haja mitigação possível, mas isso coloca em questão se vale a pena continuar com o projeto.

Marcela define o processo de gestão de riscos desde o primeiro contato de vendas. Parte do processo de Mariano deve ser analisar os riscos antes de apresentar uma proposta. Cláudio colaborará na análise financeira, valor presente, período de amortização e outras maneiras de analisar o risco monetário. A cada momento, os passos definidos por Boehm<sup>114</sup> são seguidos: Há duas etapas, a da avaliação e a do controle dos riscos. Na etapa de avaliação são distinguidas as seguintes atividades: Identificação, Análise e Priorização. Na de controle, as atividades são: Planejamento, Resolução e Monitoração. Nos detalhes de cada atividade, Marcela tomou a liberdade de adaptá-las às necessidades da empresa.

Fase	Fonte de riscos a avaliar	Categorias		
		alto	médio	baixo
pré-venda	cliente	novo, processos fracos, teoria X de gestão <sup>115</sup>	novo, processos OK, gestão moderna	conhecido, processos OK, gestão moderna
	domínio	desconhecido, baixa tecnologia, alto impacto em clientes do cliente	conhecido em parte, tecnologia moderna, impacto médio em clientes do cliente	conhecido, tecnologia avançada, impacto administrável no cliente
	tecnologia	inovadora, novas API, novas ferramentas	conhecida, algumas novas ferramentas	conhecida e compatível 100%
	concorrência	existem bons fornecedores com	existem bons fornecedores com	dominamos o mercado

<sup>113</sup> Já vimos no Capítulo 8 que a avaliação anual foi eliminada, substituída pela avaliação contínua.

<sup>114</sup> BOEHM, B., 1989, *Software Risk Management*.

<sup>115</sup> [http://en.wikipedia.org/wiki/Theory\\_X\\_and\\_Theory\\_Y](http://en.wikipedia.org/wiki/Theory_X_and_Theory_Y)

		muita experiência	nossa própria experiência	
	reuso	escassa probabilidade de reuso	algo de reuso, mas não em certos módulos críticos	fundamentalmente adaptações a nossos produtos
<i>kickoff</i>	<i>tradeoff</i> de <i>sprints</i> com requisitos	muitas funcionalidades para o release planejado	é possível modular a funcionalidade para reduzir a duração	é preferível qualidade no que é entregue do que muitas funções
	dono do produto	não entende o problema e está totalmente impedido de tomar decisões	entende o problema, mas está limitado em sua capacidade de decisão	operacional, resolutivo, independente, capaz
	custo do projeto	muito alto, relacionado com a tecnologia a ser incorporada e a curva de aprendizagem	um pouco mais alto que o habitual por sua duração	habitual
<i>sprints</i>	desenvolvimento novo x correções	nunca há tempo para corrigir porque o novo sempre tem maior prioridade	é preciso se esticar um pouco para poder corrigir o <i>sprint</i> anterior, mas é administrável	o dono do produto entende os problemas e apoia que sejam solucionados antes de chegar à estabilização
	muita funcionalidade	há pressão por incluir sempre “um pouco mais”	há pressão para incluir sempre “um pouco mais”, mas há possibilidade de diminuir o ritmo a cada dois <i>sprints</i>	pode-se descartar alguma história se não se chega com a data
	requisitos fracos	o dono do produto não sabe definir com clareza o que precisa	o dono do produto duvida do que necessita	o dono do produto sabe o que necessita e sabe explicar
estabilização	custo de manutenção	não houve muito tempo para corrigir e o <i>sprint</i> de estabilização é curto	ficam coisas para corrigir, mas entram no <i>sprint</i> se não surgirem alguns defeitos mascarados	chegamos com muito pouco para corrigir, é possível usar tempo para refazer
	entrega atrasada	o dono do produto é rígido a respeito da data de entrega e não há negociação	o dono do produto negocia qualidade por data de entrega	o dono do produto prefere um produto estável que seja entregue a tempo

Tabela 9.2: Estratégia de Riscos de Alto Nível, Fontes e Categoria

A estratégia é refinada depois da pré-venda. A aquisição de conhecimento acerca dos riscos abarca as fontes, categorias e as medidas associadas a sua eliminação, mitigação, transferência, diminuição ou tratamento da contingência. Já existem certas mitigações que foram detectadas e formam parte do conhecimento armazenado. Por exemplo, a primeira decisão de um projeto é o método organizacional que vai ser usado. Na T<sup>2</sup> os ciclos de vida autorizados estão associados ao tipo de projeto, para mitigar os riscos: *Kanban* para projetos muito simples com tarefas técnicas difíceis de separar em *sashimi* e que duram algo mais do que o habitual, *Scrum* para a maior parte dos projetos e FDD para aqueles projetos onde o volume de trabalho é tão grande que uma pirâmide de Scrums faria com que, de fato, estivessem trabalhando em uma cascata simples. De fato, FDD não foi utilizado ainda; a organização não quer correr o risco de introduzi-lo quando o cliente não é suficientemente compreensivo.

#### 9.4 Definição de um Risco

Detectar o risco não é equivalente a defini-lo corretamente. Para poder fazer a gestão do risco mais efetiva e eficiente é importante dedicar tempo à redação da definição do risco. Deve-se indicar claramente a situação atual, presente ou potencial que dispara o risco. Esta formulação diferencia entre “Já que...” e “Pode ser que...”. Por exemplo, “Já que nossa única especialista em sistemas de bases de dados persistentes ficou grávida” é do primeiro tipo, a situação já está presente. O segundo caso é em si mesmo provável, por exemplo, “Pode ser que os componentes A e B, por terem sido comprados de fabricantes diferentes, não sejam compatíveis entre si”. Neste caso não se sabe com certeza se isso é real, mas existe a

suspeita de que pode ser. A segunda parte do enunciado do risco deve apontar as consequências. Nos casos em que nosso enunciado do risco começa com “já que”, o resto da definição deve deixar claro que há uma probabilidade de que a consequência se materialize sem termos certeza. Por exemplo, o enunciado “Já que nossa única especialista em sistemas de bases de dados persistentes ficou grávida, seguramente não poderemos terminar no prazo” não é um risco, é um problema manifestado. Além do mais, está descrito de tal maneira que não é possível analisar as condições resultantes na perda (não terminar no prazo). Por outro lado, “Já que nossa única especialista em sistemas de bases de dados persistentes ficou grávida é provável que tenha que parar de ir à empresa durante os meses finais, por isso é possível que nosso *sprint* se ressinta de estabilização ao não poder contar com ela 24x7” está redigido em termos de risco e nos dá claras indicações de por onde atacá-lo. É possível montar um ambiente de tal maneira que ela possa participar no *sprint* de sua casa, ou treinar mais pessoas para que ela possa dirigi-los remotamente com pouca interação, ou conseguir alguém que possa substituí-la para esse *sprint*. Como se pode ver, é importante redigir o enunciado do risco de maneira que seja claro qual é a fonte do problema, não a gravidez nem a ausência, mas a falta de experiência no momento do *sprint* da estabilização.

A definição em termos de “pode ser que” é semelhante, seguindo o exemplo, “Pode ser que os componentes A e B, por terem sido comprados de fabricantes diferentes, não sejam compatíveis entre si, e tenhamos que investir muito esforço para fazer envoltórios<sup>116</sup> que montem API compatíveis entre eles”. Perceba que o enunciado “Dado que os componentes A e B, por terem sido comprados de fabricantes diferentes, não são compatíveis entre si, devemos investir muito esforço para fazer envoltórios que montem API compatíveis entre eles” não faz menção à probabilidade, é um problema que já existe e até anuncia o plano de ação a seguir.

## 9.5 Captura dos Riscos

Marcela adaptou uma planilha para que sirva na captura e gestão de riscos.

Planilha de Definição e Controle de Riscos											
								Nome do Projeto: _____			
								Data: _____			
								Preparado por: _____			
ID - Identificador do risco											
Adicione as colunas que sejam necessárias para monitorar a evolução dos riscos.											
Descrição do Risco - problema potencial   condição e consequência				Prob - Probabilidade de que o risco se transforme em um problema			Perd - Perda - potencial relativo da perda (monetária) ou um número entre 1 e 100		Exp - Exposição - produto entre prob e perd		
Prioridade neste análise - ranking por exposição				Prioridade a última vez - mostra a mudança acontecida			# Vezes na lista - resistência às ações				
Ação - ações a tomar para lidar com o risco				Quem - pessoa responsável pelas ações			Quando - data na que se revisarão as ações				
ID	Descrição do Risco		Prob	Perd	Exp	Prioridade neste análise	Prioridade a última vez	# Vezes na lista	Ação: eliminação, mitigação, transferência, diminuição, ou tratamento da contingência	Quem	Quando
	Condição	Consequência									
1					0						
2					0						
3					0						
4					0						
5					0						
6					0						
7					0						
8					0						
9					0						
10					0						

Figura 9.2: Planilha de Definição, Monitoração e Controle de Riscos

Ao explicar a planilha ficará claro o procedimento de definição, análise, priorização, monitoração e controle de riscos na T<sup>2</sup>. A planilha começa com uma lista de dados correspondentes ao projeto em questão. As colunas da esquerda para a direita contêm: o índice do risco, um número natural; o enunciado do risco, dividido em Condição e Consequência; a probabilidade de que o risco se materialize, quer dizer, que se transforme em um problema, calculada arbitrariamente como um número real entre zero e um (nem zero, porque então não seria necessário controlar esse risco por ser impossível que afete o projeto, nem um, porque então já é um problema); a perda estimada para o projeto caso o risco se materialize, medida em uma escala de um a cem; a exposição que traz o risco, definida como o

<sup>116</sup> Na Tecnologia da Informação, um envoltório (*wrapping*) são os dados que precedem ou marcam os principais dados de um programa, ou um programa que coloca em marcha outro programa para que possa executar corretamente. Em particular, em programação, um envoltório é um programa ou script que estabelece as bases e torna possível o funcionamento de outro programa mais importante.

produto entre probabilidade e perda, e que é o valor usado para priorizar o risco entre os demais (uma maior exposição implica em maior prioridade); a ordem de prioridade que ocupa esse risco nesta análise, onde 1 é o de maior prioridade e 10 o de menor; a mesma informação a respeito da última vez que se realizou o exercício de controle de riscos; o número de vezes que o risco está na lista, o que é indicativo de sua maturidade ou não; a ação, seja eliminação, mitigação, transferência, diminuição ou tratamento da contingência que será tomada, incluindo, quando é um plano de contingência, um disparador para iniciar a contingência; e quem vai levar adiante as ações e quando vai informar seu resultado.

## 9.6 Estratégias de Controle de Riscos

Em toda estratégia, o que conta é a sequência dos esforços. A primeira parte de uma análise estratégica consiste em identificar os riscos e priorizá-los. Depois de conhecido “o inimigo”, agora é preciso estimar os esforços que vamos dedicar a combatê-lo. Cada atividade tem seu custo e este custo não pode ser tal que o projeto se ressinta por eles. Em primeiro lugar, as estratégias da  $T^2$  (a  $T^2$  tem em seus guias de controle de riscos duas estratégias para serem aplicadas nos projetos) começam por analisar se é possível ou desejável a eliminação do risco. Dado o exemplo da seção anterior, “Pode ser que os componentes A e B, por terem sido comprados de fabricantes diferentes, não sejam compatíveis entre si e devemos investir muito esforço em criar envoltórios que montem API compatíveis entre eles” deveríamos ver se podemos eliminar esse risco. Uma alternativa sugerida é que construamos os componentes internamente. Isso modifica o projeto, os custos e a estrutura da equipe, trazendo seus próprios riscos. Outra é que desistamos do projeto. Uma terceira é buscarmos obter mais informação. Se fizéssemos um pequeno ensaio em um ambiente especialmente criado para isso e víssemos se A e B são realmente incompatíveis, poderíamos tomar melhores decisões e eliminar o risco de nossa lista.

Caso seja impossível eliminá-lo, o próximo passo é tentar transferi-lo. Caso seja o cliente quem exige o uso dos componentes, ou se é quem tem parâmetros rígidos sobre a data de entrega ou o custo, deveríamos ser capazes de ir até ele e explicar o risco, pedindo uma cláusula de risco que cubra o trabalho extra de construir os envoltórios para gerar a compatibilidade entre eles, cláusula que entra em vigor se o risco se materializar. Se formos nós mesmos que tomamos a decisão apressadamente<sup>117</sup>, então é abusivo que tentemos fazer isto. No entanto, é possível que as vantagens de usar A e B sejam tão grandes que possamos negociar algum compromisso. Por exemplo, pode ser que se, ao invés de B usarmos C, o resultado seja uma perda menor de desempenho, mas um lucro muito grande em compatibilidade. Deste modo, transfere-se o risco de incompatibilidade a um risco de desempenho.

Quando não se pode eliminar ou transferir o risco, o procedimento pede que se tente mitigá-lo. A mitigação é bem-sucedida desde que reduza a exposição que o risco causa ao projeto. Isto pode ser realizado de duas maneiras: ou é reduzida a probabilidade de que o risco se materialize ou é reduzida a perda que o risco provoca. Seguindo com nosso exemplo, já que a situação de incompatibilidade é nosso “gato de Schrödinger”<sup>118</sup>, já que a incompatibilidade existe ou não, só que não sabemos, é impossível mitigar a probabilidade de que sejam incompatíveis. Nossa única esperança é mitigar o impacto, a perda. A perda que expressamos como o esforço investido para construir artificialmente essa compatibilidade. É possível diminuí-lo? Deixemos este exercício da imaginação a cargo do leitor, porque não nos ocorre como.

Por último, se todos os demais fracassam ou se suspeitamos que as ações podem não ser efetivas, o procedimento da  $T^2$  chama o tratamento da contingência. Entende-se por isto que se planeja tomar ações que atuam contra o risco como problema. No caso do exemplo anterior, a contingência é que sejam efetivamente incompatíveis. Nesse caso será preciso construir os envoltórios para que se comuniquem por meio deles. Quanto é que podemos esperar antes de

<sup>117</sup> Tarefa para Marcela: modificar o método de decisão de aquisições para incluir na lista de verificação a compatibilidade entre componentes quando mais de um é comprado.

<sup>118</sup> Gato de Schrödinger: é um experimento imaginário concebido em 1935 pelo físico Erwin Schrödinger para expor uma das consequências menos intuitivas da mecânica quântica. Um gato, junto com um matraz (recipiente de vidro) que contém um veneno e uma fonte radiativa, são colocados em uma caixa fechada. Se um contador Geiger detecta a radiação, o frasco se rompe, liberando o veneno que mata o gato. A interpretação da mecânica quântica da Escola de Copenhague implica que, depois de um tempo, o gato está ao mesmo tempo vivo e morto.

incorrer nesse gasto-extra? Como nosso enunciado é “pode ser que”, isso indica que não estamos seguros da incompatibilidade, só temerosos de que ocorra. Começar a trabalhar nos envoltórios antes de estarmos seguros é uma perda de esforço. É escolhido um evento ou uma data como disparador desse esforço e possivelmente será revisto o plano de *sprints* para contemplá-lo.

### 9.7 Estratégia Conjunta

O guia de controle de riscos contempla o procedimento de tratamento a cada risco, mas a vida não é tão simples. Como organizar os esforços se os riscos que é preciso atender são múltiplos? Marcela identificou dois métodos, um simples e fácil de aplicar, a matriz de riscos (Figura 9.3), e outro mais sofisticado, que se chama o espectro de riscos. Vejamos primeiro o mais simples.

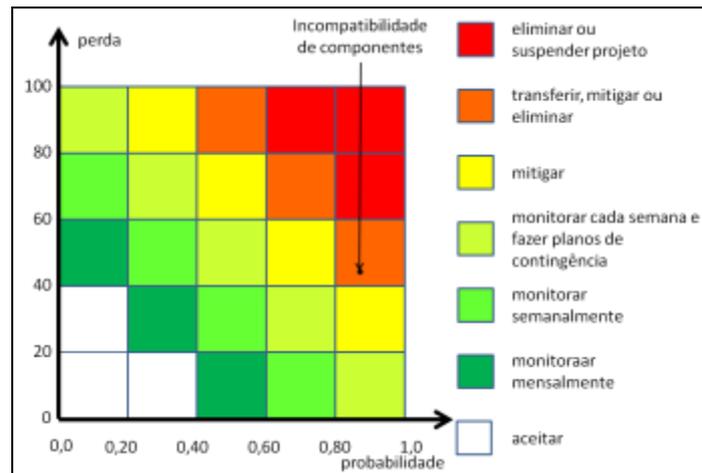


Figura 9.3: Matriz de Riscos

Os riscos estão localizados na matriz da Figura 9.3, construída como se vê sobre dois eixos, a perda na vertical e a probabilidade de ocorrência no horizontal. Foram escolhidos cinco intervalos em cada eixo, embora pudesse ter sido outro número. A estratégia conjunta consiste em adaptar as atividades ao grau de risco que possui cada um. Desse modo não se investe tempo em tentar eliminar defeitos que têm pouco impacto. As atividades relacionadas com os intervalos de cores diferentes são listadas à direita do desenho.

O outro método é o do espectro de riscos. O guia de controle de riscos o propõe para projetos que tenham muitos riscos, mas cuja importância estratégica faz com que valha a pena levá-los adiante de todos os modos. Nesse caso, é designada uma quantidade de dinheiro no orçamento ou de esforço para o tratamento dos riscos, e se avalia qual a melhor forma de investi-lo. Costuma ocorrer que a distribuição da exposição siga um padrão parecido com o espectro luminoso, não sendo distribuído uniformemente; mas há segmentos com um número maior deles do que outras de igual tamanho. Simplesmente dividir o orçamento de modo que atribua mais ao primeiro risco, menos ao segundo e assim sucessivamente, não cumpre com o objetivo de usar o esforço da melhor maneira possível. Vejamos um exemplo:

ID	Probabilidade	Perda	Exposição
1	0,8	90	72
2	0,8	84	67,2
3	0,8	69	55,2
4	0,8	66	52,8
5	0,6	64	38,4
6	0,5	71	35,5
7	0,5	40	20
8	0,5	32	16

9	0,3	37	11,1
10	0,3	33	9,9

Tabela 9.3: Exemplo de Tabela (Parcial) de Riscos

Os dois primeiros riscos estão muito próximos entre si. Como os números foram estimados para a perda e a probabilidade em cada caso, sem contar com um raciocínio muito sólido por trás, é difícil assegurar que o risco 1 é maior em exposição que o 2. Uma técnica que pode ser aplicada é a análise de sensibilidade, que veremos depois, para entender melhor as variáveis em jogo. A melhor estratégia é considerá-los da mesma categoria e tratá-los como iguais. O mesmo acontece com os dois seguintes. Isto sugere que é possível dividir o orçamento para tratamento de riscos entre as duas primeiras categorias e simplesmente monitorar as restantes. A proporção em que os recursos são repartidos fica a critério de cada equipe.

Marcela introduz os gêmeos aos pontos difíceis do MPS e recruta sua ajuda para estabelecer a lacuna com a Gestão de Riscos. Os gêmeos concordam com ela de que o escopo da gerência de riscos está bem estabelecido na política de qualidade e que pode ser aplicada universalmente na T<sup>2</sup>. Colocam GRI 1 ao lado dos resultados alcançados. O Guia de Riscos descreve as fontes e categorias dos riscos, assim como os parâmetros para analisá-los, priorizá-los e controlar o esforço que é realizado (GRI 2). As estratégias completam algumas destas coisas com ainda maior detalhe (GRI 3). Cada projeto que usou este método e a documentação conforme é discutida nas reuniões mensais do comitê de gestão estratégica (GRI 4, GRI 5, GRI 6). É revisada a cada semana, a cada quinzena ou a cada mês, segundo dita a estratégia fixada, utilizando a planilha para esse fim (GRI 7 e GRI 8). As ações correspondentes, sejam de eliminação, mitigação, transferência ou contingência, são documentadas e monitoradas para assegurar sua efetividade (GRI 9). Marcela está satisfeita, o Guia de Gestão de Riscos da T<sup>2</sup> está de acordo com o MPS.

#### O QUE ACONTECEU ATÉ AGORA

78. Depois de um período de seis meses de estabilidade, Marcela anuncia os planos para passar ao Nível C em uma avaliação conjunta.
79. Jéssica introduz uma nova ferramenta de decisões, a árvore de decisão.
80. Baseando-se nas apresentações de Marcela e nos resultados da árvore da decisão, o Comitê de Gestão aprova de maneira unânime o investimento para alcançar o Nível C do MPS-SW em uma avaliação conjunta com um SCAMPI de Nível Definido (3).
81. A primeira implementação do Nível C é a do processo gerência de riscos, que é construído e implantado em tempo recorde.
82. Depois de divulgado o uso do Guia de Gestão de Riscos da T<sup>2</sup>, Marcela e os gêmeos constataam que cobre todos os resultados esperados de GRI no MPS.

### 9.8 Nota de Cautela

O especialista em risco, economista e financista, Nassim Nicholas Taleb dedicou sua vida a analisar questões relacionadas com o acaso e a probabilidade. Seus dois últimos livros, [TALEB, 2010] e [TALEB, 2012] exploram o imprevisível. Sua tese é que a normalidade das coisas não é importante, que o que realmente modela o mundo é o acaso, os grandes acontecimentos imprevisíveis, que ele chama “cisnes negros”. Seus livros estão cheios de exemplos, incluindo a peste negra e a ocupação da América pelos europeus no século XVI, de acontecimentos que pareciam impossíveis à geração que os vivia e que mudaram as vidas das pessoas e as trajetórias das nações. Esta visão do acaso, compartilhada em outras ciências<sup>119</sup>, força as organizações a repensar sua estrutura e a administração do risco. Já que um “cisne negro” não pode ser previsto, é imprescindível organizar-se para que a sua chegada não seja destrutiva no curto, médio e longo prazo. No curto prazo, as empresas e organizações deveriam criar planos de contingência alinhados com sua sobrevivência ante desastres, ao estilo de [TOIGO, 2002]. No médio prazo, as empresas e organizações precisam ter a

<sup>119</sup> É a tese dominante no neodarwinismo, ver por exemplo, *Dinosaur in a Haystack: Reflections in Natural History*, [GOULD, 1996].

flexibilidade e a adaptabilidade para não só tolerar as novas circunstâncias, mas aproveitá-las. No longo prazo, as empresas e as organizações devem saber que se ficarem paralisadas, o próximo cisne negro poderá destruí-las e, por isso, devem se segmentar, dividir e multiplicar sem endurecer seus canais de informação.

### 9.9 Decisões Formais

Do mesmo modo que os riscos são analisados seguindo procedimentos bem definidos para agregar valor, recolhendo novos riscos e as soluções correspondentes que devem ser consideradas, há decisões especiais que são tomadas, quando necessário, com base em métodos formais<sup>120</sup>. O propósito de usar métodos formais ao tomar decisões é que, seguindo um padrão formal, podem armazenar, comparar, analisar e reutilizar as decisões baseando-se nos resultados que produzem. A princípio, a T<sup>2</sup> tinha só métodos muito básicos, baseados em matrizes de Pugh com pesos entre alternativas, mas a importância das decisões tornou clara a necessidade de ferramentas mais potentes, incluindo árvores de decisão, simulações, modelos e correlações simples. A T<sup>2</sup> começou, corretamente, o avanço no caminho da alta maturidade em seus processos estratégicos.

Vejamos como foi feito. Novamente Marcela com sua bagagem do Mestrado em Administração recorreu a um professor seu, o Dr. Zito, que recomendou várias leituras<sup>121</sup>, das quais extraiu Marcela poderosas ideias que transformou em um Guia de Aplicação de Decisões da T<sup>2</sup>.

O Guia é parte do material usado na integração de novo pessoal. A justificativa reside em que, em todo projeto de *software*, frequentemente a equipe se enfrenta com a necessidade de tomar uma decisão. Em quase todas essas ocasiões, as alternativas são claras e as variáveis que as diferenciam são visivelmente observáveis e fáceis de comparar. Selecionar entre as alternativas acaba sendo simples e justificável a decisão tampouco implica um custo extra para a equipe: as características da solução são óbvias, a geração de alternativas simples e a decisão é uma consequência lógica de aplicar bem os passos anteriores. Nesses casos, a aplicação documentada de um processo formal é desnecessária. Mas em algumas ocasiões, a decisão significa um risco muito alto para o projeto e acaba sendo pouco aconselhável tomá-la de maneira informal. As decisões “difíceis” não mudarão por serem tomadas formalmente, mas se o procedimento formal é executado passo a passo, a organização em seu conjunto consegue aprender de seus erros quando estes são cometidos. De outra maneira, as mesmas decisões podem ser tomadas várias vezes em diferentes projetos sem que haja necessariamente ocorrido uma aprendizagem de um para outro. O Guia formaliza o processo de tomada de decisões para garantir que sejam geradas e capturadas todas as informações necessárias, de modo que, se a decisão for acertada, pode ser repetida e se não for, que possa ser analisado o que foi decidido para melhorar essa decisão em futuros projetos.

O Procedimento de Análise de Alternativas e Tomada de Decisões (PAAeTdD) permite à organização tomar decisões de maneira consistente. Em particular, o PAAeTdD ajuda os participantes a tomarem decisões difíceis, aquelas que implicam riscos a bens ou pessoas. O PAAeTdD é um procedimento sistemático que descreve passo a passo as atividades a serem realizadas para poder formalizar a geração, documentação, avaliação e seleção de alternativas entre o espaço de soluções de um problema. Consiste em identificar claramente o problema a resolver, estabelecer objetivos da solução (seus atributos), gerar (identificar) alternativas, descompor o problema e modelá-lo, medir as alternativas contra o método de avaliação e selecionar a melhor entre elas. O Procedimento de Análise de Alternativas e Tomada de Decisões (PAAeTdD) não foi construído com o propósito de ser aplicado a todas as decisões que são tomadas em uma equipe. A lista que segue pretende ser um guia da aplicação do procedimento, não necessariamente exaustiva. O PAAeTdD pode ser utilizado em múltiplas ocasiões, conforme seja útil, a critério da equipe ou da gerência da organização.

Para a aplicação do PAAeTdD, o Guia estabelece parâmetros claros: deve ser utilizado quando a equipe enfrenta decisões que têm uma ou mais das seguintes características: (a) A decisão está relacionada com um tópico que é considerado de alto risco e que está sendo monitorado formalmente, ou (b) A decisão afeta o plano de trabalho de modo que impacta em mais de 5%

<sup>120</sup> De fato, na literatura há muitas publicações que falam ao mesmo tempo de riscos e decisões formais.

<sup>121</sup> Em particular, o Dr. Zito recomendou o livro de CLEMEN, 1997, mas Marcela também achou útil um livro muito mais simples, *Decision Analysis in Projects. Learn to Make Faster, More Confident Decisions*, de [SCHUYLER, 1996].

da *baseline* do produto, ou (c) A decisão afeta o plano de trabalho de modo que impacta em mais de 5% do orçamento, ou (d) A decisão afeta o plano de trabalho de modo que impacta em mais de 5% a data de entrega do produto, ou (e) A decisão afeta o plano de trabalho de modo que impacta em mais de 5% na qualidade final do produto, ou (f) O cliente requer que a tomada de decisões fique integralmente documentada para a decisão em questão, ou (g) O impacto estimado da decisão supera em mais de 100% o custo esperado de aplicar o procedimento PAAeTdD. O PAAeTdD contém os seguintes passos:

1. Definir corretamente o problema a ser resolvido.
2. Estabelecer atributos desejáveis da solução.
3. Definir métodos de medição dos atributos selecionados.
4. Criar métodos de avaliação das soluções.
5. Gerar ou descobrir soluções alternativas.
6. Aplicar a cada solução gerada a medição dos atributos desejáveis.
7. Avaliar mediante os métodos selecionados as soluções geradas.
8. Selecionar a melhor alternativa.

**Tabela 9.4: Definição dos Passos do PAAeTdD**

Desenvolveremos cada um dos passos com um pouco mais de detalhe. Começamos pelo passo 1, Definição do Problema. O propósito deste passo é estabelecer o enfoque do problema e assegurar que se pretende resolver o problema correto. Busca-se alinhar o enfoque com os objetivos de negócio da organização e identificar as causas principais do problema, para usá-las como entrada ao passo da seleção de soluções alternativas. As tarefas envolvidas são Fixar os objetivos de Resolução do Problema; Identificar e listar diferentes pontos de vista sobre o que constitui o problema; Reduzir por agrupamento o número de temas a uma quantidade administrável; Organizar os temas em ordem de importância para o objetivo do projeto; Realizar uma análise das causas profundas dos possíveis problemas; Listar cada uma das causas profundas consideradas e reduzi-las por agrupamento; Dividir as causas em três grandes categorias: imediatas, próximas, a considerar alguma vez. Os passos exigem que os participantes estejam capacitados para realizar *brainstorms*, agrupamento de temas e o uso de ferramentas como o diagrama de causa-efeito, hierarquia de objetivos, diagrama de Ishikawa ou diagrama de espinha de peixe. Também devem ser suficientemente responsáveis para poder realizar uma triagem dos problemas.

O passo 2, Atributos da Solução tem como propósito estabelecer os objetivos que precisam ser cumpridos por uma solução, quer dizer, definir os atributos de uma solução aceitável em termos dos objetivos. A única tarefa é identificar os atributos de uma boa solução.

O passo 3, Métodos de Medição tem como propósito fixar medidas objetivas dos atributos que devem pertencer à solução escolhida. As tarefas envolvidas são as seguintes: Definir o indicador adequado para o atributo em questão, Descrever os critérios de análise envolvidos no modelo de medição, Listar as medidas derivadas das medidas básicas e as funções de composição requeridas, Listar as medidas básicas e seu método de medição, e Definir claramente as unidades de cada medida em cada nível.

O passo 4, Métodos de Avaliação tem como propósito descrever o(s) método(s) que permitam discriminar entre soluções alternativas. As tarefas envolvidas são as seguintes: Hierarquizar e dar pesos relativos aos diferentes atributos das soluções, Ponderar possíveis relações entre atributos considerados independentes, e Definir funções que ponderem os resultados obtidos por meio dos indicadores para que as soluções possam ser comparadas. Em princípio, os atributos escolhidos para representar uma boa solução são independentes uns dos outros. As diferentes soluções exigem que seja realizada uma análise dos pesos relativos entre os atributos para poder ponderá-los, porque se não for assim os resultados, ao se encontrarem em um espaço multidimensional, acabam sendo não comparáveis. Este mecanismo é conhecido como "*trade-off analysis*".

O passo 5, Soluções Alternativas tem como propósito definir um conjunto de soluções que possam resolver o problema. As tarefas envolvidas são as seguintes: Revisar a lista de causas profundas de realização imediata, Gerar um *ranking* por prioridade de causa, e Sugerir soluções alternativas a cada causa de problemas detectada. Apesar de esta atividade poder ser realizada de maneira imediata depois da identificação do problema, é aconselhável postergar sua realização até que haja um conhecimento mais profundo da natureza da solução, coisa que as atividades listadas antes podem fornecer. Por outro lado, é possível que durante a realização da atividade "1. Definição do Problema" seja natural a identificação de possíveis

soluções como resultado da tarefa “Identificar causas profundas”. Cada equipe e cada circunstância devem indicar qual das duas opções é a correta para a ocasião.

O passo 6, Medições das Alternativas tem como propósito obter indicadores para cada solução alternativa definida a partir da medição dos atributos predefinidos para poder avaliá-las. Há só uma tarefa envolvida: Aplicar as medidas definidas em 3 (Métodos de Medição) às alternativas identificadas em 5. (Soluções Alternativas).

O passo 7, Avaliação das Alternativas tem como propósito gerar a tabela final de resultados para poder selecionar a solução a ser adotada. As tarefas envolvidas são: Calcular o “valor” de cada uma das alternativas utilizando os valores obtidos na medição dos atributos e a função de avaliação antes definida, Ordenar as soluções em ordem decendente e apresentá-las em forma de tabela, Revisar os resultados e corrigir as ferramentas quando a ordem obtida contradiz o sentido comum. Chegando a este ponto, é bom revisar o realizado ao longo de todo o processo, julgando-o a partir da tabela gerada. Se os resultados contradizem o sentido comum, é bom se perguntar se é este que deve ser modificado, ou se foram introduzidos erros de conceito nas funções definidas para medir e combinar medidas na construção do valor da solução individual.

O passo 8, Seleção da Melhor Alternativa tem como propósito o que diz o título, obviamente. As tarefas envolvidas são: Revisar e explicar os resultados da tabela final, Identificar e expor prós e contras da melhor alternativa, Definir com precisão o impacto da alternativa em termos de planos e tarefas, e Obter consenso em adotar a solução de melhor valor para a organização. Um traço da maturidade organizacional é a implementação inteligente de soluções escolhidas por meio de métodos objetivos. Neste ponto, a solução deve ser implementada com detalhe para que a direção tome a decisão de maneira consciente e responsável.

Marcela se propõe a acrescentar, além disso, os métodos ao repertório de tomada de decisões da T<sup>2</sup>. Primeiro, redige instruções para construir uma hierarquia de objetivos. Dado o problema que é preciso identificar, procura-se o efeito que se tenta conseguir. Por exemplo, temos que decidir entre dois fornecedores de um produto. Qual é o objetivo? Talvez haja muitos, de modo que entender qual é o mais importante é crucial para a decisão. Digamos que a equipe se divide entre os que acreditam que “entregar a tempo” é o objetivo e aqueles que acreditam que “entregar com qualidade” é o objetivo, e que a escolha surge claramente, independente de quem escolha. Isso pode paralisar a decisão de forma completa, de modo que é necessário encontrar um objetivo de ordem maior que os contenha. Neste caso, poderia ser “entregar a tempo e com qualidade”, o que não ajuda muito; melhor seria “entregar sempre no prazo”. Este último objetivo é melhor porque nos leva a analisar o impacto, em projetos futuros, de entregar com baixa qualidade este produto (Figura 9.4). Quantos recursos estarão ausentes para realizar emendas no produto que entregamos com defeitos conhecidos? Isto leva rapidamente a outra ferramenta de pensamento, a árvore da decisão. Quando as decisões estão conectadas, a matriz de Pugh não captura isso de maneira direta. É necessária uma nova ferramenta, que já vimos no começo deste capítulo, na Figura 9.1, introduzida por Jéssica para analisar a decisão de partir ou não para a avaliação de Nível C e de fazê-la, ou não, conjuntamente.

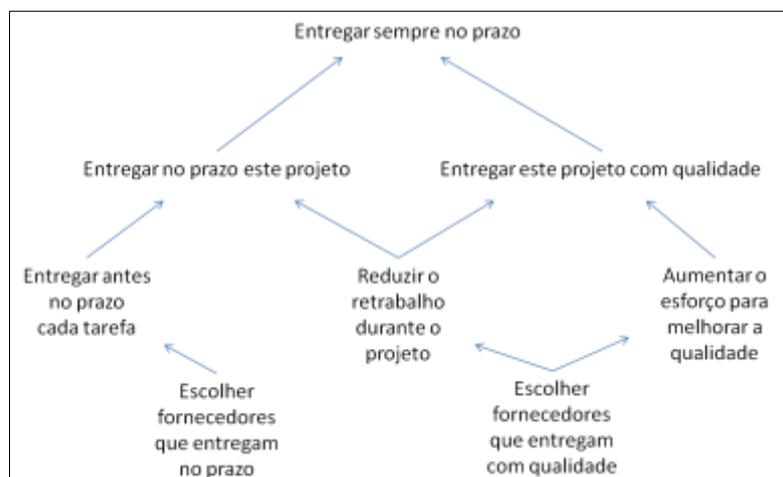


Figura 9.4: Árvore de Objetivos

Quando aparece a probabilidade na análise, e este é o caso com frequência na tomada de decisões, pois o problema está inserido em um ambiente de relativa incerteza, a ferramenta que é considerada na T<sup>2</sup> é a árvore de decisão. A árvore nasce de um nó-raiz do qual saem as primeiras decisões, porque a árvore pode ser usada para tomar decisões combinadas, sejam estas independentes ou não uma das outras. Por exemplo, pode-se usar para decidir entre fornecedores de um produto semelhante e, ao mesmo tempo, decidir se a manutenção deve ser contratada ou não (as decisões são independentes) ou pode ser usada para decidir entre dois produtos, um com adaptações e outro sem elas, e decidir conjuntamente se subcontrata-se a adaptação (as decisões não são independentes, um dos ramos ficará vazio porque a decisão de adaptação só se aplica em um caso). Esta versatilidade para se adaptar a decisões complexas é o traço mais destacado da árvore da decisão. Tampouco há restrições muito grandes sobre sua sintaxe, o importante é que as ideias sejam expressas claramente. A tabela de pagamentos (Tabela 9.1) que acompanha a árvore de decisão exemplifica mais claramente a obtenção de resultados.

Para completar o exemplo, vejamos como se pode refinar a Árvore para uma análise de diferentes possibilidades no caso de se optar pelo Nível C do MPS-SW (Figura 9.5), pensando que os benefícios podem ser diferentes com diferentes (ou as mesmas) probabilidades. Isto é fácil de expressar em uma árvore de decisão.

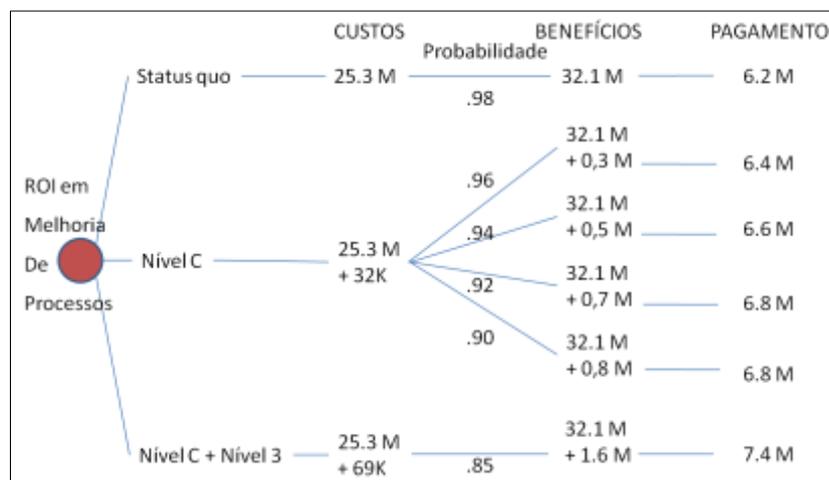


Figura 9.5: Árvore de Decisão Refinada

A tabela de pagamentos correspondente, com comentários, é mostrada na Figura 9.6.

	Estimativa experiente		Probabilidade x Entradas				fixos por Nível C	
	Status Quo		Nível C			SCAMPI		
	Entradas	Gastos	Novas Entradas		Gastos	Novas Entradas	Gastos	
	32.1	25.3	0.3	0.5	0.7	0.8	1.6	0.069
Probabilidade	0.98	1	0.96	0.94	0.92	0.9	1	0.85
Expectativa	31.458	25.3	0.288	0.47	0.64	0.72	1.36	0.069
Pago	6.158		0.256	0.438	0.612	0.688	1.291	

Expectativa - Gastos

Figura 9.6: Tabela de Pagamentos Correspondente à Árvore de Decisão Refinada

Marcela acrescenta outra técnica ao repertório, o diagrama de dependências ou diagrama de influências. Um diagrama de influência é uma representação visual simples em forma de gráfico de um problema de decisão. Oferece uma maneira intuitiva de identificar e representar elementos essenciais de um problema deste tipo, incluindo objetivos, decisões, elementos de incerteza ligados a probabilidade e às relações entre eles.

No gráfico podemos diferenciar nós e arcos. Os arcos são dirigidos e representam relações entre os nós. Os nós representam decisões (nós quadrangulares, chamados nós de decisão), variáveis aleatórias (nós ovais, chamados nós estocásticos) cujo valor é conhecido só em probabilidade e também utilidades (nós em forma romboide, chamados nós de valor).

Por exemplo, ampliando o diagrama de hierarquia de objetivos com decisões de investimento em *marketing* podemos obter um diagrama de domínio como o que é mostrado na Figura 9.7.



Figura 9.7: Diagrama de Influências com Inclusão de Outros Investimentos

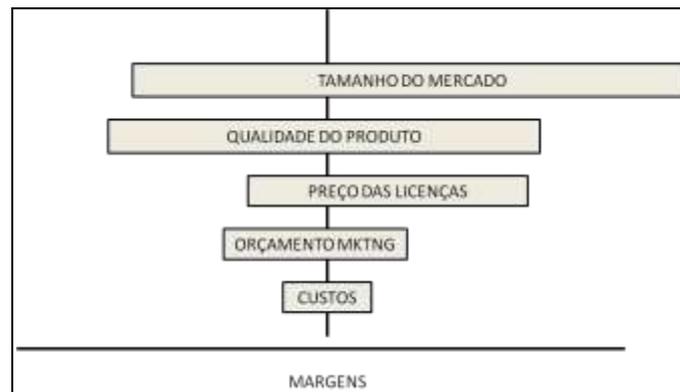
No diagrama da Figura 9.7 assumimos que o tamanho do mercado é uma variável aleatória que não pode ser influenciada pelas decisões que são tomadas. Isto poderia ser revisado, se o investimento em *marketing* pudesse alterar isto, por exemplo, incluindo clientes de outros países. Também poderiam existir outras decisões que influenciasssem nessa variável. De qualquer modo, naquelas variáveis onde opera o acaso, manteremos nós estocásticos no diagrama embora sejam dependentes, como é o caso de custos. Também é possível notar que a qualidade do produto influencia na cota do mercado e nos custos. O nó “número de licenças vendidas” é redundante, porque se desprende seu valor de “cota de mercado” e “tamanho de mercado”, mas o propósito de um diagrama de influências é mostrar as relações, pelo que sua inclusão não vulnera os objetivos. O diagrama de influências é uma ferramenta para a discussão de ideias, não um objetivo em si mesmo. Depois que se entendem as relações entre as variáveis são utilizadas outras técnicas como apoio à decisão.

Um dos problemas de estimativas é a insegurança sobre os valores estimados. Sua variação é importante em função de compreender o impacto de uma decisão. Suponhamos que o tamanho do mercado é de vários milhões de licenças possíveis. Para fixar um número, digamos que se trata de 10.000.000 de licenças. Agora, se cada licença representa uma entrada de 1.000 dólares anuais, estamos falando de um mercado de 10.000.000.000 de dólares. Uma variação de um ponto na cota de mercado representa então 100.000.000 de dólares. Parece razoável então que sejam utilizados os meios mais potentes para as estimativas caso se trate de uma cota de 1% ou de 0,002%, sobretudo se a qualidade do produto tiver uma influência decisiva neste valor. Este tipo de análise, o de entender até que ponto a incerteza associada a seus parâmetros numéricos poderia fazer variar a utilidade esperada afetando as decisões, ou em outras palavras, onde é preciso investir em conhecimento para diminuir a incerteza, denomina-se “análise de sensibilidade” e a ferramenta mais comum para realizá-la é o diagrama de tornado.

Estes diagramas mostram graficamente as mudanças que são produzidas na utilidade esperada quando varia uma quantidade ou valor específico. Se vamos mudando uma a uma as variáveis, mantendo as demais em seu valor original, obteremos uma classe de utilidades esperadas para cada uma delas. Estas classes são representadas como barras em um gráfico. Estas barras são ordenadas de cima para baixo e da mais à menos longa, de modo que o gráfico fique parecido a um tornado. Os mais longos indicam que a mudança dos valores da variável que representam implica uma maior mudança na utilidade esperada, o que é o mesmo que dizer que a importância de uma variável ao alcançar um resultado é maior quanto maior for a barra correspondente no diagrama de tornado. Geralmente, variam os valores entre dois extremos, o mínimo provável e o máximo provável, de modo que a incerteza refletida pode se refletir no impacto sobre o objetivo. Para as variáveis que não mostram sensibilidade, investir em melhorar o conhecimento de sua incerteza não é útil, assim como melhorar seu valor em função disso tampouco é.

O diagrama da Figura 9.8 foi construído sem nenhuma base de fórmulas, somente para ilustrar a forma que têm os diagramas de tornado. Se as fórmulas que produziram esse diagrama existissem, do diagrama pode-se ler que o tamanho do mercado é a variável que mais influencia no objetivo de margens. Investir para ter melhor conhecimento desse tamanho é sensato. Pelo contrário, nosso objetivo de aumentar as margens parece ser inelástico relativamente ao orçamento de *Marketing*. Seria melhor transferir esse orçamento para

melhorar a qualidade do produto, que está sob nosso controle, para aumentar assim as margens.



**Figura 9.8: Diagrama de Tornado**

Para construir o diagrama de tornado, uma ferramenta útil é conhecer a dependência estatística entre as variáveis. Este conhecimento se expressa em uma equação chamada de “regressão” que modela a relação entre uma variável chamada “dependente”, em nosso caso “Margens”, e um conjunto de variáveis “independentes” que influenciam no comportamento da variável dependente. Este modelo assume a forma de uma equação  $Y = c_0 + c_1 X_1 + c_2 X_2 + \dots + c_n X_n + \varepsilon$ . Os coeficientes  $c_j$  são os que determinam a variabilidade correspondente no diagrama de tornado. Um coeficiente muito grande em comparação aos demais permite supor que o impacto da variação da variável independente correspondente é maior que a dos demais. Isto, é claro, está limitado a que tal variação seja possível. O coeficiente  $c_0$  é o que estabelece a ordenada à origem, quer dizer, o valor de  $Y$  quando todos os valores independentes são nulos. O coeficiente  $\varepsilon$  é um artifício, chama-se termo aleatório e só existe para que se cumpra a igualdade. Se esse valor  $\varepsilon$  é conhecido, então se pode conhecer o “ajuste” da equação.

O problema da regressão consiste em escolher uns valores determinados para os parâmetros desconhecidos  $c_j$ , de modo que a equação fique completamente especificada. Para isso é necessário um conjunto de observações. Em uma observação qualquer  $i$ -ésima ( $i=1, \dots, n$ ) é registrado o comportamento simultâneo da variável dependente e as variáveis explicativas (as perturbações aleatórias supostamente não são observáveis). Mediante o uso de técnicas estatísticas são obtidos tais coeficientes. Dada a existência de múltiplas ferramentas de cálculo de tais coeficientes (começando pela Excel) e a natureza deste livro dispensaremos o leitor dos detalhes matemáticos e remetemos os interessados à literatura clássica<sup>122</sup>. O uso que se dá à equação de regressão na  $T^2$  é o de poder calcular os valores dependentes para uma árvore de decisão, mas veremos que deste modesto início surge uma aplicação muito madura, no capítulo seguinte.

#### O QUE ACONTECEU ATÉ AGORA

83. Marcela segue conselhos de seu professor e implementa um Guia de decisões que cumpre claramente com os resultados do MPS para o processo GDE.
84. É feito um piloto do processo para tomada de decisões que é aprovado para sua difusão.
85. Marcela acrescenta um Guia de Métodos para que as decisões não se limitem somente a matrizes de Pugh.

### 9.10 A Fábrica de Componentes

Mariano chega muito excitado ao escritório da  $T^2$  em uma terça-feira pela manhã. Acaba de descer do avião que o trouxe de Toronto e convoca uma reunião urgente do comitê de gestão. Sentado na sala de reuniões espera que cheguem os demais. Um a um vão se cumprimentando, servem-se do primeiro cafezinho do dia e perguntam o motivo de tanta urgência a Mariano, que sorri enigmaticamente e desvia a conversa para outros tópicos, como

<sup>122</sup> [SPIEGEL, 2011] é nosso livro de cabeceira para Estatística Elementar.

a turbulência que sempre acontece sobre o equador nos voos, ou a baixa temperatura que virou costume no interior dos aviões. Por último chegam Ana e Alfredo, que tiveram de passar antes pela creche. Mariano se levanta, respira fundo e começa sem poupar detalhes:

*“Colegas da T<sup>2</sup>, bem-vindos à grande empresa. Ontem assinamos com a TransKND um acordo de entendimento para que financiem a reengenharia de nosso produto central para hospitais e farmácias. Seu financiamento cobre a criação de uma arquitetura de linha de produto orientada a serviços, SOA. O investimento lhes dá direitos de comercialização exclusivos em mercados internacionais, mas não dá nenhum privilégio técnico nem participação alguma em decisões financeiras internas da T<sup>2</sup>. Neste momento o Board da TransKND está reunido em Toronto para ratificar ou recusar o acordo, do mesmo modo que nós estamos fazendo aqui. Revisemos o acordo”.*

Em seguida, distribui cópias a cada um, com as partes importantes previamente marcadas. O Comitê discute durante várias horas; mas, ao finalizar a discussão, o acordo é aprovado e é encomendado a Mariano e Cláudio a assinatura do contrato. Ana também vai participar na parte técnica. Alcançado o resultado, Mariano liga para sua contraparte na TransKND e os empresários se cumprimentam, chegaram ao mesmo resultado positivo no hemisfério norte, há uma aliança. A reunião se dissolve antes que os gêmeos tenham tempo de ir buscar mais Mumm.

Assim que saem no corredor, Ana gruda em Marcela e solicita sua ajuda. As duas, com apoio dos demais, devem dar forma ao processo que será utilizado para a reengenharia e sucessivamente, para que todos os projetos aproveitem a SOA.

### 9.11 *Service Oriented Architecture (SOA)* para Principiantes<sup>123</sup>

SOA é um conceito que é aplicado na construção de sistemas distribuídos de grande porte. Para compreender SOA é fundamental entender as propriedades deste tipo de sistemas. Isto significa que é preciso poder integrar código legado, porque ninguém pode parar as máquinas e voltar atrás na história quando os sistemas já produziram milhões de linhas de código. Assim, um projeto de SOA é mais parecido com um projeto de reengenharia do que com um projeto de desenvolvimento. Envolve mudar a estrutura de um sistema, mas sem reescrevê-lo. É preciso desacoplar e envolver os serviços que já estão sendo fornecidos para transformá-los em um formato que permita construir facilmente a partir das partes resultantes.

Os serviços são unidades de funcionalidade não associadas, fracamente acopladas, que não têm chamadas entre si. Cada serviço se refere a uma ação, como preencher um formulário *online* para uma conta, ou ver um extrato de conta *online*, ou a solicitação de uma reserva *online* ou comprar passagens de avião.

Os desenvolvedores que utilizam SOA associam objetos individuais (serviços) de SOA mediante o uso da instrumentação. No processo de instrumentação da funcionalidade, o desenvolvedor de *software* associa serviços em uma disposição não hierárquica a uma ferramenta de *software* que contém uma lista completa de todos os serviços disponíveis, suas características, assim como os meios para construir uma aplicação que utilize estas fontes. É necessário então que haja um sistema de comunicação heterogêneo para que se possa compatibilizar arquiteturas de décadas anteriores com os últimos avanços tecnológicos.

Vejamos o produto estrela da T<sup>2</sup>, o sistema que vincula o estado de cada paciente com seu histórico clínico, a medicação e a farmácia do hospital, assim como suas farmácias fornecedoras. Já tem três anos de desenvolvimento e cresceu mais de um milhão e meio de linhas de código. Há muitas regras de negócio embutidas que não podem ser jogadas fora ou recodificadas em novas linguagens e provadas em novos ambientes. É imprescindível preservá-las. Isto sugere que façamos as menores mudanças possíveis.

Mas agora imaginemos o novo grande produto da T<sup>2</sup>, o sistema hospitalar, hoje pendurado na nuvem, mas conectando enfermeiros, médicos, pacientes, familiares dos pacientes, sistemas de ambulância e todos os outros integrantes da grande família da indústria da saúde. É preciso desacoplar o código das interfaces e gerar um protocolo para que os telefones, os *Blackberry*, os *tablets*, os *laptops*, e toda outra futura invenção que usem correntes de *bits* possam aproveitar essas regras de negócio e aumentar a capacidade e velocidade da decisão dos

<sup>123</sup> [JOSUTTIS, 2009]

interessados. Em lugar de chamar os serviços entre si, por métodos ou rotinas de seu código fonte, utilizam protocolos definidos que descrevem como os serviços são transmitidos e analisam mensagens mediante metadados de descrição.

Atrás disto e para permitir que funcione, é necessário que estes metadados tenham detalhe suficiente para descrever não somente as características destes serviços, mas também os dados que os impulsionam a funcionar. Os programadores fizeram uso extensivo de XML em SOA para estruturar os dados descritos em um envoltório quase exaustivo da descrição do conteúdo. Analogamente, a linguagem de descrição de serviços *Web* (WSDL) no geral descreve os serviços em si, enquanto que no protocolo SOAP são descritos os protocolos de comunicação. Se estas linguagens de descrição são as melhores possíveis para o trabalho, e se serão convertidas ou continuarão sendo as favoritas no futuro, estas são perguntas abertas. A partir de 2008, a SOA depende dos dados e serviços que são descritos por metadados que devem cumprir com os dois critérios seguintes:

1. Os metadados devem vir de tal forma que os sistemas de *software* possam utilizá-los para que se configurem dinamicamente, mediante a descoberta e a incorporação de serviços definidos, mantendo coerência e integridade. Por exemplo, os metadados podem ser utilizados por outras aplicações, como um catálogo, para realizar a detecção automática dos serviços sem necessidade de modificar o contrato funcional de um serviço.
2. Os metadados devem vir em uma forma que os *designers* de sistemas possam compreender e trabalhar com um gasto razoável de custo e esforço.

SOA tem como objetivo permitir aos usuários encadear “peças”, bastante grandes de funcionalidade para formar aplicações *ad hoc*, que são construídas quase por completo a partir dos serviços de *software* existentes. Quanto maior for o bloco, menor a quantidade de pontos de interface necessários para implementar qualquer conjunto dado de funcionalidade. No entanto, se os pedaços de funcionalidade são muito grandes, podem não ser suficientemente granulares a ponto de poderem ser reutilizados facilmente. Cada interface leva consigo uma certa quantidade de “grava-me” de processo, pelo que a granularidade dos serviços é uma consideração de rendimento, tanto para o serviço quanto para seus futuros usos. Se for muito pequeno sobrecarrega as interfaces, se for muito grande reduz o reuso. Esta característica faz com que seja muito rentável o uso de SOA em domínios específicos, ao invés de domínios muito gerais.

Os serviços SOA estão menos acoplados que as funções de biblioteca conectadas para formar um executável. Os serviços SOA também são executados em *wrappers* “seguros” (como Java ou .NET) e em outras linguagens de programação que administram a designação de memória e recuperação, permitem enlace *ad hoc* e tardio, e proporcionam um certo grau indeterminado de tipo de dados (*data typing*).

SOA, como arquitetura, baseia-se na orientação a serviços como princípio fundamental de *design*. Se um serviço apresenta uma interface simples que abstrai a complexidade subjacente, os usuários podem acessar os serviços independentes sem saber como foi implementado o Santo Graal do reuso. A grande promessa da SOA é que o custo marginal da criação da enésima aplicação é baixo, já que todo o *software* necessário já existe porque foi criado para satisfazer os requisitos de outras aplicações anteriores. Idealmente, só é preciso instrumentação para produzir uma nova aplicação.

A fim de utilizar eficientemente um SOA, a arquitetura deve cumprir os seguintes requisitos:

- Deve existir interoperabilidade entre os diferentes sistemas e linguagens de programação que proporcionam a base para a integração entre aplicações em diferentes plataformas por meio de um protocolo de comunicação. Um exemplo de tal comunicação é o conceito de mensagens. O uso de mensagens por meio de canais definidos diminui a complexidade da aplicação final, permitindo deste modo ao programador da aplicação concentrar-se na funcionalidade em lugar das complexidades de um protocolo de comunicação.
- O desejo de criar uma federação de recursos. Estabelecer e manter o fluxo de dados em um sistema de base de dados distribuída. Isto permite

novas funcionalidades desenvolvidas para fazer referência a um modelo de negócio comum para cada elemento de dados.

Estes requisitos são centrais para a T<sup>2</sup>, pois são essenciais à visão de interoperabilidade e independência da configuração *hardware* e *software* do cliente. O primeiro passo é construir a arquitetura de referência SOA, que fornece um plano de *design* de serviços de uma implementação através de uma organização. Um dos aspectos relevantes em SOA é definir a Arquitetura de Referência para a Empresa, porque permite ter um marco de referência onde localizar os futuros desenvolvimentos ou ainda aqueles componentes de serviços convenientemente empacotados. No caso da T<sup>2</sup> esta arquitetura de referência deve ser documentada, ampliada e refinada, mas o conhecimento do domínio permite visualizar que esse processo não será doloroso nem longo.

A Arquitetura de Referência SOA modela os diferentes componentes de uma solução SOA, principalmente Processos de Negócios e Serviços. Além disso, mostra como interagem estes componentes com os usuários de negócio e com os sistemas existentes na Empresa (sistemas legados). Geralmente orientada a camadas pela independência que este modelo arquitetônico fornece e o pouquíssimo acoplamento que produz, há grandes semelhanças com o *design* de Sistemas Operacionais modernos<sup>124</sup>. Há definições bem documentadas<sup>125</sup> de tudo o que é necessário descrever e como criar um registro de metadados<sup>126</sup>.

SOA é menos uma arquitetura do que um paradigma para a construção e manutenção de processos de negócios que abarcam sistemas distribuídos de grande porte<sup>127</sup>. Um componente essencial de uma arquitetura SOA é o *bus*. O *bus* é o grande unificador, permite eliminar a necessidade de conhecer detalhes de interfaces entre serviços. Em troca, exige a existência de um registro dos metadados que cada um espera receber e a capacidade de cada um de interpretar metadados. Isto pode levar a um grande caos, por isso todos os autores recomendam incorporar um nível de supervisão e guia para governar o registro e a instrumentação.



Figura 9.9: Ilustração da Arquitetura SOA

### 9.12 Montando a Fábrica

Para o processo de reengenharia dentro da T<sup>2</sup>, Ana propõe contratar um ajudante que já trabalha com ela na universidade e que cumpre com as condições e tem as competências necessárias. Passadas as entrevistas e as verificações de antecedentes, o novo engenheiro de *software* passa a ser o arquiteto do projeto<sup>128</sup>. Junto com Ana e os gêmeos, elaboram a arquitetura de alto nível, aplicando JMC<sup>129</sup>. Depois desse exercício, começa a funcionar a Fábrica de *Software*: dependendo do domínio, em particular do componente a converter em

<sup>124</sup> [BORIA, 1989]

<sup>125</sup> [http://www.soa.com/solutions/metadata\\_federation/](http://www.soa.com/solutions/metadata_federation/)

<sup>126</sup> Uma busca no Google devolve mais de um milhão de sites

<sup>127</sup> [JOSUTTIS, 2009]. *SOA in Practice: The Art of Distributed System Design*.

<sup>128</sup> Uma dessas coisas que só acontecem com *softwares*!

<sup>129</sup> Ver capítulo 8, *Java Modeling in Color*.

serviço, os gêmeos recomendam um especialista que assume o papel de programador chefe da equipe<sup>130</sup>. Será ele que se encarregará dos detalhes do envoltório e da definição de metadados para o registro. Para a escrita do envoltório, montará sua equipe e a liderará. Resumindo, uma aplicação clássica de FDD e SOA.

#### O QUE ACONTECEU ATÉ AGORA

86. Uma empresa canadense estabelece uma aliança com a T<sup>2</sup> para refazer a arquitetura do produto de saúde passando-a para SOA.
87. Marcela e Ana definem um processo para fabricar os serviços a partir do código existente, aplicando FDD.

### 9.13 O nível C do MPS

Revisando com Máximo o que aconteceu, mais uma vez chamado para realizar uma análise de lacunas, a ênfase recai sobre duas coisas: O processo de Desenvolvimento para o Reuso e os resultados de atributos de processo, que marcam o grau de institucionalização dos processos.

Por causa do novo acordo com a empresa do Canadá, o resultado DRU 1: “Os domínios de aplicação nos quais serão investigadas as oportunidades de reuso de ativos ou nos quais se pretende praticar reuso são identificados, detectando os potenciais de reuso respectivos”, está claramente coberto. O segundo resultado, DRU 2: “A capacidade de reuso sistemática da organização é avaliada e as ações corretivas são tomadas, caso sejam necessárias”, fica evidenciado com a incorporação de um arquiteto para SOA e a instalação do projeto baseado em FDD. O terceiro resultado, DRU 3: “Um programa de reuso, cobrindo propósitos, escopo, metas e objetivos, é planejado a fim de atender às necessidades de reuso de domínios”, é algo também evidenciado por este projeto. O quarto resultado, DRU 4, “O programa de reuso é implantado, monitorado e avaliado” decanta dos mesmos preceitos de governança que são usados em todos os projetos da T<sup>2</sup>. DRU 5 diz: “As propostas de reuso são avaliadas de forma a garantir que o resultado do reuso seja apropriado para a aplicação objetivo”, é aplicada no mecanismo de avaliação de cada *sprint* do projeto SOA, enquanto que a DRU 6, “Formas de representação para modelos de domínio e arquiteturas de domínio são selecionadas”, fica evidenciada pelo registro e a arquitetura de negócios que constituem o nível mais alto de especificação do projeto, realizado por Ana e os gêmeos. DRU 7, “Um modelo de domínio é desenvolvido e seus limites e relações com outros domínios são estabelecidos e mantidos”, está contemplado no registro de metadados, já que o resultado é elaborado deste modo: Este modelo deve ser capaz de capturar características, capacidades, conceitos e funções comuns, variantes, opcionais e obrigatórias. Também a escolha de SOA permite facilmente evidenciar o DRU 8: “Uma arquitetura de domínio descrevendo uma família de aplicações para o domínio é desenvolvida e mantida por todo seu ciclo de vida”. Por último, DRU 9, que pede que “Os ativos de domínio sejam especificados; adquiridos ou desenvolvidos, e mantidos por todo seu ciclo de vida”, é parte do próprio contrato com a TransKND e o coração do novo projeto. Assim, o processo DRU do nível C do MPS-SW não constitui um problema.

Em relação aos resultados dos atributos de processo, o RAP, a revisão alcança aqueles aplicados até o Nível C e não foram substituídos por outros no processo de amadurecimento. Estes são: RAP 1, o mais básico, que faz referência ao “Grau em que o processo alcança seus resultados definidos”, este é o motivo para que se realize a revisão dos resultados dos processos, de modo que não é uma preocupação para Marcela e sua equipe. Tampouco surgem muitas dúvidas sobre os seguintes resultados esperados: RAP 2. “Existe uma política organizacional estabelecida e mantida para o processo”; RAP 3. “A execução do processo é planejada”; RAP 4. “As medições são planejadas e coletadas para monitorar a execução do processo e os ajustes são realizados”; RAP 5. “As informações e os recursos necessários para a execução do processo são identificados e postos à disposição dos interessados”; RAP 6. “Os papéis necessários, as responsabilidades e a autoridade para a execução do processo definido são designados e comunicados”; RAP 7. “As pessoas que executam o processo são competentes em termos de formação, treinamento e experiência”; RAP 8. “A comunicação entre as partes interessadas no processo é planejada e executada de forma a garantir seu envolvimento”; RAP 9. “Os métodos adequados para monitorar a eficácia e adequação do

<sup>130</sup> Ver capítulo 3, *Feature Driven Development*.

processo são determinados e os resultados do processo são revistos com a gerência de alto nível para dar visibilidade sobre sua situação na organização”. Todas estas formam parte do planejamento normal de projetos, programas e *sprints*, ou da estrutura de controle de qualidade e do mecanismo de governo instituído nas reuniões mensais do comitê de gestão da T<sup>2</sup>. O mesmo acontece com RAP 10, “A aderência dos processos executados a suas descrições de processo, padrões e procedimentos é avaliada objetivamente e são tratadas as não conformidades”.

A implementação da boa gerência de configuração na qual insistiram todos, em particular os gêmeos, produz as evidências necessárias para RAP 11, “Os requisitos dos produtos de trabalho do processo são identificados”; RAP 12, “Os requisitos para documentação e controle dos produtos de trabalho são estabelecidos”; RAP 13, “Os produtos de trabalho são colocados em níveis apropriados de controle”; e RAP 14, “Os produtos de trabalho são colocados objetivamente em relação aos padrões, procedimentos e requisitos aplicáveis e são tratadas as não conformidades”.

Máximo teve de comparar planos e processos para descobrir que, efetivamente, o que é dito é o que é planejado, e o que é planejado é o que é feito. Mas isto serviu para ver da mesma forma a evidência de RAP 15, “Um processo padrão é descrito, incluindo diretrizes para sua adaptação”; RAP 16, “A sequência e interação do processo padrão com outros processos são determinadas”; RAP 17, “Os papéis e competências exigidos para executar o processo são identificados como parte do processo padrão”; e finalmente de RAP 18, “A infraestrutura e o ambiente de trabalho exigidos para executar o processo são identificados como parte do processo padrão”. Tudo isto Máximo encontra na BiPro quando compara os planos dos projetos com os processos que lhes dão origem.

Nessa análise, Máximo apoia-se no processo definido que cada projeto adota, seguindo os guias de adaptação, o que lhe permite observar evidência de RAP 19, “Um processo definido é implementado baseado nos guias para seleção e/ou adaptação do processo padrão”; RAP 20, “A infraestrutura e o ambiente de trabalho exigidos para executar o processo definido são colocados à disposição, geridos e mantidos”; e finalmente RAP 21, “Os dados apropriados são coletados e analisados, constituindo uma base para a compreensão do comportamento do processo, para demonstrar a adequação e a eficácia do processo, e avaliar onde podem ser realizadas melhorias contínuas do processo”.

Máximo conclui que a T<sup>2</sup> está preparada para a avaliação do Nível C e, em consequência, pela força do modelo MPS, para realizá-la conjuntamente com um SCAMPI de Nível 3, Definido, do CMMI-DEV.

#### O QUE ACONTECEU ATÉ AGORA

88. Máximo conclui que a T<sup>2</sup> está preparada para a avaliação de Nível C e, em consequência, dada a força do modelo MR-MPS-SW, para realizá-la conjuntamente com um SCAMPI de Nível 3 do CMMI-DEV.

# Parte IV – Apogeu

## CAPÍTULO 10 - ESTABILIZAR PARA A MELHORIA CONTÍNUA

*“Não é a espécie mais forte a que sobrevive, nem a mais inteligente, mas sim aquela que melhor se adapta às mudanças”*

*Charles M. Darwin*

### 10.1 Níveis B e A do MPS

Já se foram vinte meses desde que visitamos nossos amigos da T<sup>2</sup> e já passou muita água sob a ponte. Quando nós os deixamos, eles estavam preparando a avaliação conjunta do nível C do MR-MPS-SW com o nível 3 (Definido) do CMMI-DEV, avaliação que se realizou sem sobressaltos e com sucesso poucos meses depois. O desenvolvimento da Fábrica de *Software* baseado em arquitetura SOA era incipiente e o crescimento se acentuava no acordo com a empresa do Canadá. Quase dois anos mais tarde, todos os resultados da T<sup>2</sup> se acentuaram positivamente. A empresa tem três linhas de produtos, é conhecida pelo primeiro ERP seguro na nuvem, tem ações na Bolsa e suas marcas são reconhecidas. Chegar à cidade, vindo do aeroporto, é percorrer um *outdoor* atrás do outro lembrando ao viajante que está na cidade da T<sup>2</sup>. Os fundadores são conhecidos nos programas de televisão locais e convidados frequentemente para seminários e palestras. Ainda são jovens, alguns mudaram de estado civil, Adolfo e Ana já têm três filhas, os gêmeos continuam sendo mais conhecidos nos bares que em sua casa paterna, Mariano se casou, Cláudio vive com a namorada. Marcela adotou um enorme cachorro, cruzamento de Mastim com São Bernardo, que ocupa seus dias mais que seu namorado. A vida é boa.

### 10.2 Estabilidade

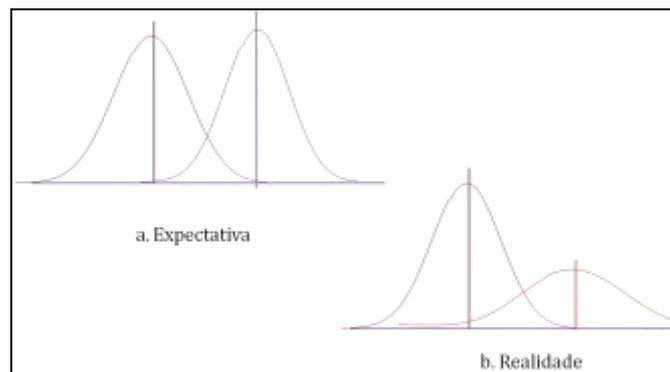
Em todos os aspectos de nossas vidas, a estabilidade nos oferece uma sensação de tranquilidade. Quando os acontecimentos são estáveis e as surpresas são mínimas, sentimos que estamos seguros. Até em esportes radicais, os seres humanos querem controle sobre o que ocorre. Os filmes de aventura nos tiram da zona de conforto e aceleram nossa adrenalina porque nos apresentam um panorama onde o futuro é incerto, não é possível fazer previsões e, em consequência, mesmos os melhores planos acabam mal.

Nossos amigos da T<sup>2</sup> deduziram o mesmo de uma experiência particular nas primeiras tentativas de planejar seus *sprints* para o projeto SOA. Durante a adoção do nível E do MPS, a estimativa de esforço e duração das tarefas foi calculada na T<sup>2</sup> a partir do tamanho dos produtos a serem desenvolvidos. No *Scrum*, no princípio, o tamanho foi calculado na forma de pontos de história. A partir do nível D, foram utilizados pontos de casos de uso, dada a introdução destes como parte da definição de escopo do projeto. No Kanban são usadas diferentes medidas de tamanho. No entanto, como o uso do Kanban é pouco frequente, voltado, a princípio, para controlar certas tarefas de duração média, não há um método tão rigoroso. A partir da estimativa de tamanho, assume-se como valor inicial do esforço o valor resultante da equação de regressão para tarefas desse tipo, tomando por base os valores históricos da base de dados de processos.

Conscientes de que estes eram processos e procedimentos novos, os gêmeos começaram a gerar os dados para que a equação pudesse ser usada com o mesmo efeito. Durante três meses, as estimativas foram feitas com base apenas na experiência e os parâmetros utilizados na equação de regressão foram ajustados até que estivessem adaptados aos coeficientes derivados dos pontos de história. Mesmo assim, os gêmeos se surpreenderam com a grande disparidade que começaram a observar entre os *sprints* da fábrica. As previsões erravam por vários dias, até por semanas em algumas ocasiões. Armados com as ferramentas de análise, começaram a trabalhar. Por que um processo, o de estimativa, que tinha sido suficientemente bom até agora, já não servia com a mesma capacidade? Obviamente que o primeiro questionamento foi a novidade do domínio e a consequência imediata foi comparar os dados

de ambas modalidades. Pegaram os dados históricos para tarefas semelhantes (análise, construção, teste) e os carregaram na ferramenta de software de análise estatística<sup>131</sup> em uso e com sua ajuda produziram curvas normais para ambas as populações. Esperavam que os valores médios fossem diferentes, mas acreditavam que no resto as distribuições não seriam muito diferentes.

As duas distribuições terminaram muito diferentes. O valor médio era, como esperado, muito maior para a nova população (à direita nos gráficos da Figura 10.1), mas a maior diferença era constituída pela dispersão<sup>132</sup>. Os gêmeos pediram a ajuda de Cláudio, para usar seus conhecimentos de estatística<sup>133</sup> recentemente adquiridos. Cláudio mostra-se um pouco surpreso pela pergunta dos gêmeos que querem saber por que as estimativas são tão imprecisas no novo projeto quando eram tão precisas nos anteriores. Dá uma resposta muito sucinta a eles, que parece tão boa que os gêmeos o convidam a realizar uma breve exposição para os líderes técnicos que produzem as estimativas.



**Figura 10.1: Distribuições de Esforço de Tarefas Semelhantes em Duas Populações**

Cláudio se dirige pela primeira vez a um auditório dentro da T<sup>2</sup>, mas não é a primeira vez que faz uma exposição em público. Ele também é “vítima” do sucesso da T<sup>2</sup> e é frequentemente convidado a fazer apresentações sobre a análise financeira aplicada à análise de portfólio, como vem fazendo na T<sup>2</sup> há anos. Portanto, está preparado, embora fazer uma apresentação sobre o tema de estatísticas seja novo para ele. Sua exposição, baseada em slides ppt, intitula-se “Estimativa e Dispersão”. Seu argumento central é que a precisão da estimativa não é um problema do método de estimativa, mas do comportamento do processo estimado. Para ilustrar isto, ele escolhe o seguinte exemplo:

*“Vamos supor que estamos tentando estimar o erro de um relógio, no qual observamos a hora que ele marca e a comparamos com a hora real, medida por um instrumento confiável, como um computador, a cada vinte minutos. No final de um dia, 24 horas, teremos 72 observações. Agora, imaginemos que observamos um cronômetro. Nesse caso, os desvios estarão, possivelmente, abaixo do limite de tolerância de nossas medições, quer dizer, se comparamos as medições em segundos, é possível que a diferença entre o computador e o cronômetro seja menor do que um segundo. Mesmo se fosse perto de um segundo, a dispersão dos dados é mínima. Por outro lado, se observamos um relógio de parede que ficou sem bateria, ele sempre marca a mesma hora. Como o relógio só marca 12 horas, cada medição que fazemos do erro aumenta em vinte minutos até alcançar ou passar das 6 horas, então diminui de vinte em vinte minutos. Se pegamos os erros com seu sinal, eles se compensam entre si, de maneira que o valor médio de ambos os relógios é zero, talvez se algum valor médio não é zero, provavelmente é o calculado com o uso do cronômetro. Claro que tomamos os desvios*

131 Há muitas ferramentas de software para análise estatística, incluindo algumas de código aberto e extensões para Excel. Muito difundido é o Mini Tab. Para as simulações de Monte Carlo, Crystal Ball. Para uma lista, ver <http://alternativeto.net/software/minitab/>.

132 As medidas de dispersão, também chamadas medidas de variabilidade, mostram a variabilidade de uma distribuição, indicando por meio de um número, se as diferentes pontuações de uma variável estão muito distantes da mediana média. Quanto maior for esse valor, maior será a variabilidade, quanto menor for, mais homogênea será à mediana média. Assim se sabe se todos os casos são parecidos ou variam muito entre eles. [http://en.wikipedia.org/wiki/Statistical\\_dispersion](http://en.wikipedia.org/wiki/Statistical_dispersion)

133 Cláudio iniciou cursos de doutorado em análise financeira para empresas. De passagem, especializou-se em Auditoria.

em seu valor absoluto; mas, de todo modo, isto é indicativo de que o valor médio não é o melhor modo de calcular estimativas, porque há uma grande parte da informação que reside na dispersão e que o valor médio não captura. Se usamos a média do erro para ajustar nossa estimativa, o relógio da parede nos trará surpresas: cada observação está muito distante da realidade, no entanto, a média compensa. O método é o mesmo, o erro, totalmente diferente. Depois, o erro não vem do método, mas da distribuição subjacente da variável que mede o comportamento do processo. Se não gostamos de nossas estimativas, é inútil punir quem está fazendo as estimativas, é preciso melhorar a estabilidade do que tentamos estimar”.

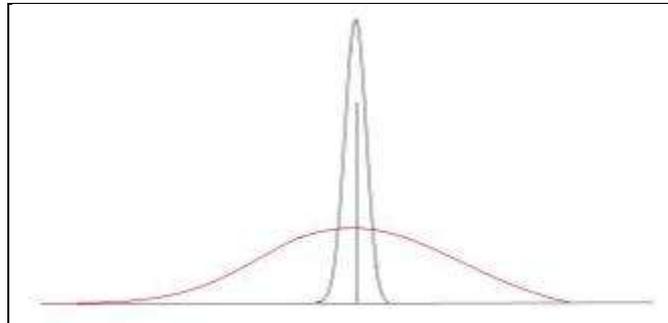


Figura 10.2: Distribuição do Erro em Dois Relógios

### 10.3 Eliminando Aleatoriedade

O resultado é que os processos de SOA não são estáveis. Procurando a causa-raiz, são encontradas várias candidatas: os procedimentos não estão definidos com suficiente precisão, há diferentes interpretações de como devem ser executados; dentro deste mesmo tema, diferentes execuções têm diferentes graus de aderência ao processo (fidelidade); os ajustes que o guia permite foram mal feitos e isto passou despercebido pela qualidade; pelo fato de ser um processo novo, as medições estão mal automatizadas e a granularidade é maior que nos casos de *Scrum*, onde as tarefas estão claramente separadas e a coleta de dados é límpida.

O primeiro passo para controlar o problema é esclarecer o processo inicial de definição do *sprint*, que antes era bem descrito pelo jogo do planejamento, mas como este não é aplicado em SOA, é substituído por um procedimento centrado na estimativa individual do tamanho, realizada pelo programador-chefe. Esta estimativa não é revisada e é parte da fonte do erro. Marcela se encarrega de tornar o procedimento de contagem preciso, construindo um método semelhante aos pontos de caso de uso que já utilizam, mas baseado na densidade e extensão dos metadados que requer o serviço ou o componente em questão. Colocados em prática, os resultados são promissores, a dispersão diminui significativamente em poucas aplicações, mas decidem continuar com o experimento até que as ferramentas estatísticas mostrem que o resultado é significativo a 10%<sup>134</sup>. Isto não os detém na melhoria contínua, já que Marcela acrescenta auditorias e revisões que na pressa para começar o projeto SOA foram deixadas de lado. Que isto seja uma lição: até as melhores intenções costumam dar oportunidade ao otimismo e o resultado pode ser pior do que o esperado. Nada substitui a vigilância contínua. De passagem, as análises estatísticas só conduzem a ideias e novas perguntas – não a soluções. Para solucionar os problemas detectados é preciso analisar as causas e possivelmente criar novas medições para investigá-las claramente.

### 10.4 O Céu Desaba

Os resultados permitem que novamente a gerência confie nas estimativas e a estabilidade resultante induz a uma feliz sensação de estar no controle. Até que um acontecimento incomum sacode todo mundo. Jéssica se lembra de suas leituras de Taleb<sup>135</sup> e os “cisnes negros”. Um projeto novo de SOA andava normalmente, quer dizer, de acordo com as expectativas, até que no *Sprint* de estabilização o céu desabou: os defeitos não só estavam fora de toda previsão, mas cada correção acrescentava novos. A conta de defeitos abertos

<sup>134</sup> Nos testes estatísticos, é considerado um resultado estatisticamente significativo se for tão extremo que tal resultado é esperado que surja por casualidade só em raras circunstâncias. A porcentagem expressa quão raro ele é: 10% significa que uma em cada nove vezes o resultado é fruto da escolha feita a partir da amostragem, 1% indica que uma em cada cem vezes é o caso. Ver [SPIEGEL, 2011].

<sup>135</sup> [TALEB, 2010] e [TALEB, 2011], op. cit., ver Capítulo 9.

conhecidos subia de ciclo de teste em ciclo de teste. Um típico cenário de projeto fora de controle<sup>136</sup>. Ao invés de diminuir, os defeitos aumentam a cada tentativa de eliminar um deles. O comitê de gestão “aperta o botão de emergência” e interrompe a linha de produção. Em uma reunião dos programadores-chefe, é realizada uma análise das causas-raiz que não chega a nenhuma conclusão. Pela primeira vez em sua história, a T<sup>2</sup> está perplexa e sem resposta. Marcela recorre, como sempre, à literatura existente e aos seus bons contatos acadêmicos. O Dr. Zito, que fez consultoria para empresas internacionais de *software* em encarnações anteriores de sua profissão, sorri beatificamente ao escutar sua narrativa, como se espera dos professores titulares que já sabem o que devem responder.

“Marcela”, diz ele, “estamos falando do problema de complexidade das interfaces. É típico que o acoplamento baixo produza, em um sistema suficientemente grande, um caos de interpretação. A parte do *software* de vocês que interpreta os metadados tornou-se mais volátil e complexa que os benefícios que obtêm do baixo acoplamento. O problema já era conhecido na época da análise estruturada, Constantine já falava disto, em 1979<sup>137</sup>. A pergunta-chave é: Quanto um módulo deve conhecer do outro para entendê-lo e se comunicar com ele? Quanto mais devemos saber do módulo B para entender do módulo A, mais acoplados estão A e B. O fato de termos que saber algo sobre o outro módulo, já é uma prova a priori de que há certo grau de interconexão, inclusive se a forma de interconexão não é conhecida. Isto é levado ao extremo no caso de SOA. Infelizmente, chega um ponto onde não ter que saber nada de nada implica poder aprender de tudo no momento de se conectar. Essa é a complexidade que estão enfrentando agora”.

“Mas”, diz Marcela, “por que funcionou antes e agora não?”.

“O que está diferente neste projeto em relação aos anteriores?”, perguntou o Doutor.

“É um projeto feito a partir do zero em um domínio que não conhecemos muito bem”, responde Marcela.

“Aí está”, diz o Doutor. “A resposta é essa. SOA não é a ferramenta para isso, estão abusando do modelo”.

Marcela agradece e vai embora pensando no que acaba de aprender. Sua perplexidade diminuiu, mas não desapareceu. Por que uma ferramenta tão útil em alguns casos é tão ruim em outros? E, sobretudo; Como se conserta isto agora? Claro que, por outro lado também, seu viés de melhoria contínua a inquieta: isso poderia ter sido prevenido?

O comitê de gestão está reunido na reunião de emergência. Marcela expõe o caso como viu com o Doutor Zito. Na ida da faculdade ao escritório teve uma “Eureka”<sup>138</sup> e a transmite ao comitê. Ela utiliza um método de Ford que é conhecido como “as oito disciplinas”<sup>139</sup>, pelo qual é definido corretamente o problema da mesma forma que no caso da análise de causas-raiz, mas é acrescentado um elemento importante: a contenção do problema antes de se preocupar em solucionar as causas-raiz.

Com isto em mente, Marcela faz sua apresentação: o problema é a necessidade de atender todos os metadados possíveis em um domínio novo, onde não há experiência necessária para entender o que é o crítico e o que é o acessório. Desse modo, os envoltórios (*wrappers*) terminam sendo extremamente complexos, quase programas de Inteligência Artificial. Isto

---

<sup>136</sup> [GLASS, 1997]: Um projeto de *software* é classificado como “fora de controle” quando excede em mais de 30% suas estimativas. No caso que citamos, o indicador de gestão que aponta para isto é o crescimento permanente dos defeitos conhecidos e abertos em cada ciclo de teste.

<sup>137</sup> [YOURDON e CONSTANTINE, 1979]. Muitos atribuem grande parte do material a Constantine, daí a citação do Doutor pela metade.

<sup>138</sup> Eureka! ou *Heureka* em grego – “Encontrei!”, é uma famosa exclamação atribuída ao matemático grego Arquimedes, que segundo a lenda pronunciou esta palavra depois de descobrir que o peso de um corpo, dividido pelo seu peso aparente ao ser submerso em água, é uma propriedade que hoje conhecemos com o nome de densidade. Isto permitiu que ele descobrisse se a coroa do Rei era feita de ouro puro. Esta descoberta teria sido feita enquanto ele se encontrava submerso na banheira. Tal foi sua alegria que saiu pelas ruas de Siracusa nu e gritando “Eureka!”. Marcela teve sua Eureka no carro e vestida, por sorte.

<sup>139</sup> [http://en.wikipedia.org/wiki/Eight\\_Disciplines\\_Problem\\_Solving](http://en.wikipedia.org/wiki/Eight_Disciplines_Problem_Solving)

ocorre porque não há um código herdado que se possa abstrair, mas desenvolvido à medida que se define o serviço.



Figura 10.3: O Método das Oito Disciplinas

Marcela agora chama um grupo de programadores-chefe que já tinha selecionado e comunicado a necessidade de estarem preparados para esta reunião, a qual chama de uma retrospectiva acelerada. Entram na sala e Ana, que já foi avisada por Marcela sobre o direcionamento que quer dar à reunião, assume o comando e conduz a discussão sobre os ajustes arquiteturais que deverão ser efetuados para contar com um produto dentro de prazos razoáveis. Os programadores se sentem muito incomodados, já que as reuniões do comitê de gestão se tornaram lendárias na T<sup>2</sup>, material de muitas histórias nascidas ao abrigo das decisões tomadas nela por aqueles que os mais jovens consideram os pais da T<sup>2</sup>.

Timidamente eles expõem suas queixas, quase todas justificadas: A direção superestimou a capacidade de adquirir conhecimentos por parte das equipes, assim como a capacidade dos clientes de se expressar com precisão e de uma maneira integral. Os métodos de Denney não serviram porque as definições iniciais não continham regras de negócio suficientemente claras nem completas. Os metadados não foram definidos de forma centralizada e única, cada um criou e adaptou às suas necessidades, à maneira que surgiam. O atraso no *sprint* de arquitetura foi tomado como um bom sintoma, como se o tempo investido na análise fosse se pagar somente no futuro, quando na realidade foi interrompido abruptamente porque todos sentiram que continuar era melhor que esperar para entender bem o problema do usuário.

O ambiente é cada vez mais sombrio dentro da sala. Só os programadores-chefe falam, na verdade, sussurram, suas sensações. Em um ponto Alberto Galarraga tenta refutar alguns comentários, mas Ana o interrompe.

“Armando”, ela diz, confundindo-o com seu irmão, “*não há nada a refutar, a percepção é a realidade neste caso. Não importa se pensamos que foi de outra maneira, para poder consertar este desastre precisamos nos colocar na visão dos que devem resolvê-lo*”.

“A armadilha de Deming: ficamos tão penderes dos recursos que nos esquecemos dos objetivos”, acrescenta Alfredo, que ficou pensando no problema.

Os programadores-chefe se sentem respaldados, mas o medo não desaparece. Marcela pede, quase implora, que sejam propostas soluções para o problema atual. Um dos mais jovens, que foi aluno de Ana, sugere começar de novo, mas com métodos diferentes e diminuir o tempo reusando o código já escrito. Tratar em parte o código como um legado, mas começar com um modelo mais preciso do negócio do cliente. Outro soma sua voz, dizendo que está de acordo, mas que precisam deixar o SOA de lado, acabar com os metadados por enquanto. As outras vezes se incorporam para pedir que os requisitos sejam refeitos. Alfredo refuta:

“*Não há tempo para refazê-los, o máximo que dispomos é de seis semanas, estourando dois meses, e, mesmo assim, fazendo concessões ao cliente*”.

“Não é completamente certo que não haja tempo”, retruca Mariano, normalmente calado nas reuniões técnicas. “Podemos aplicar JAD e salvar isto no prazo”.

### 10.5 Contenção

Agora é a vez de Mariano expor sua ideia. Sucintamente, descreve JAD para os não iniciados. Ao escutá-lo, Marcela lembra-se que Mariano era um entusiasta do método já nos anos em que os dois estudavam juntos no “Poli”.

JAD é a abreviação usada para *Joint Application Development*<sup>140</sup>, um método integral de construção de *software* baseado fortemente na participação de todos os interessados na construção do modelo do sistema antes de escrever o código. De certo modo, é um dos precursores do método ágil, porque a equipe é autônoma e escolhe seus processos até certo ponto e os períodos estão prefixados. O método baseia-se em uma investigação inicial, em essência, a coleta dos requisitos, para construir um modelo do sistema em questão. Esta etapa, segundo opina Mariano, pode ser realizada sem intervenção do cliente porque as equipes já têm o conhecimento necessário. A segunda fase constitui a identificação de uma equipe e sua convocação para revisar e atualizar o modelo. Nesta etapa, há dois resultados críticos: a escolha dos participantes e o respaldo da alta gerência. Os participantes devem ser usuários CRACK<sup>141</sup>, quer dizer, devem ser pessoas reconhecidas no ambiente do cliente, contar com capacidade de decisão, contar com conhecimento de domínio e colocar-se à disposição da equipe. O patrocínio da alta gerência é necessário por duas questões: a convocação à reunião deve vir da alta gerência, de outro modo é pouco provável que os usuários CRACK atendam; e a outra razão é que, ao fazer a convocação, o patrocinador deve garantir o comprometimento de todos com o resultado. Ninguém pode dizer “isto não é o que eu queria, mas para terminar a discussão preferi aceitá-lo”. O compromisso com o resultado obriga todos tanto a participar quanto a negociar, já que o projeto tem duração fixa. A terceira fase é a junta de análise em si, onde o modelo é proposto, analisado e melhorado. As objeções que são levantadas na reunião são incorporadas ao modelo durante a noite e apresentadas de novo no dia seguinte. Isto aliado à presença de pessoas com conhecimento e poder de decisão geram uma dinâmica muito rápida, e entre três e cinco dias é possível atender o que habitualmente leva de dois a três meses. Ficarão alguns pontos abertos, mas os responsáveis por fechá-los terão um prazo inadiável para fazer isso. Enquanto se fecham as últimas ações, o código é escrito, provado e demonstrado.

Os programadores-chefe aprovam a estratégia e Mariano se compromete a convencer o cliente se o comitê respaldá-los. Os gêmeos continuam com dúvidas, mas não têm uma opção melhor. Mariano fica encarregado de falar com o cliente e facilitar as futuras atividades de JAD.

### 10.6 Causas-Raiz

Até certo ponto a reunião atacou as causas-raiz, mas ainda não de forma sistemática. Marcela utiliza dados do fracasso para mostrar os resultados e iniciar uma discussão de análise. Jéssica, que está secretariando a reunião e recolheu os comentários dos programadores chefe, propõe fazer a análise a partir deles.

	<b>risco ou problema:</b>	<b>causa(s)-raiz:</b>	<b>mitigação:</b>
1	A capacidade de adquirir conhecimentos por parte das equipes foi superestimada	Não foi seguido o processo de riscos na parte de pré-venda	Acrescentar uma auditoria de processo à atual auditoria de produto e fazer uma revisão por colegas do produto
2	A capacidade dos clientes de se expressar com precisão e de maneira integral foi superestimada		Acrescentar aos riscos na pré-venda o conhecimento do negócio por parte do ponto de contato no cliente
3	As definições iniciais não continham regras de negócio	Os processos de negócio estavam sofrendo	Colocar uma cláusula no contrato que torna o cliente

<sup>140</sup> [WOOD e SILVER, 1995]

<sup>141</sup> Ver Capítulo 3.

	suficientemente claras nem completas	reengenharia ao mesmo tempo em que se tentava descrevê-los para construir o sistema	responsável pelas demoras relacionadas com requisitos incompletos
4	Os metadados não foram definidos de forma centralizada e única, cada um criou e adaptou às suas necessidades, à maneira que surgiam	Não há uma planilha única, não há guias de ajuste nem de uso	Construir a planilha padrão para metadados e os artefatos necessários para seu uso produtivo na BiPro
		Não foi implementado um sistema de gerência dos metadados e não há revisões das definições porque os processos de <i>Scrum</i> não são mais aplicados	Implementar um sistema para gerência dos metadados, incluir as definições de metadados entre os artefatos e documentos sujeitos a revisão por inspeção e criar a lista de itens de controle para poder realizá-las objetivamente
5	O atraso no <i>sprint</i> da arquitetura foi visto como um bom sintoma, como se o tempo investido na análise fosse se pagar somente no futuro quando, na realidade foi interrompido abruptamente porque se sentiu que continuar era melhor que esperar para entender bem o problema do usuário	?	

Tabela 10.1: Os Problemas do Projeto Fora de Controle

A reunião avança rapidamente enquanto são tratados os primeiros quatro pontos, mas chega a um impasse quando o quinto é confrontado. Há silêncios longos que são preenchidos por tosses e sugestões inconclusas, um “Vamos ver, se fizermos...” que é diluído sem terminar, ou um “E se colocássemos, por outro lado...” que tampouco prospera.

Jéssica finalmente decide intervir.

*“Isto acontece conosco porque somos bons fazendo autópsias, mas não sabemos fazer diagnósticos antecipados”*, diz, decidida.

*“Como, como, como?”*, pergunta Alfredo com uma curiosidade sincera.

*“Sim, somos médicos legistas, trabalhamos com cadáveres, mas não somos clínicos. Não sabemos medir a glicose no sangue, os triglicerídeos, a quantidade de glóbulos vermelhos, de leucócitos, de cálcio. Não sabemos e o que é pior, se tivéssemos os números não saberíamos interpretá-los”*, insiste Jéssica, que gosta de falar com parábolas e hipérboles.

Ana começa a entender:

*“Não somos clínicos, não fazemos prevenção, não fazemos análise, não medimos... é isso?”*, pergunta Ana.

*“Mas nós medimos”*, diz Alberto.

*“E também fazemos análise”*, diz Armando.

*“Mas fazemos tudo isso quando o fato já ocorreu, fazemos post mortem, é autópsia, é rigor mortis e já não há mais nada a fazer”*, exagera mais uma vez Jéssica, que tem certa afinidade com o dramaturgo dos países bálticos, Henrik Ibsen.

A discussão se anima, todos parecem querer falar ao mesmo tempo, em parte fascinados pelo que acham ser certo, em parte rejeitando as ideias porque não são descritivas da realidade.

### 10.7 A Previsão Como Ferramenta

O que Jéssica quis dizer em sua forma tão peculiar é que, apesar de serem feitas previsões, estas não são categorizadas. Quer dizer, é escolhido um ponto central e é projetado para saber quantos defeitos vamos encontrar, qual é o número de vezes que temos de executar cada caso de teste ou quando vamos entregar com a qualidade prevista, mas todas estas estimativas estão imbuídas de erro e não se insere este erro no cálculo. Lembrando a equação de regressão, devemos considerar o  $\epsilon$ , porque se este valor for muito grande, não poderemos confiar na estimativa. De certo modo, é a mesma discussão sobre a estabilidade transposta às equações de regressão. Se os processos são estáveis, deveria ser possível estabelecer a correlação entre os valores das variáveis independentes e os valores dependentes, com ajuste à aleatoriedade que elas têm, quer dizer, prever o valor médio dentro de um intervalo “de confiança”. Dessa maneira, poderíamos afirmar coisas como dizer que o projeto está dentro dos limites esperados, ao invés de colocar valores arbitrários de 15% acima e abaixo do valor médio.

Em termos das teorias de qualidade, o que não estamos fazendo é “escutar a voz do processo”. Os processos têm comportamentos derivados do grau de estabilidade que têm na empresa. Quanto menor for a dispersão, maior é a estabilidade.

### 10.8 Previsão e Análise

Nós humanos baseamos nosso conhecimento em uma lei: Como foi ontem será hoje. A repetitividade de um fenômeno é o parâmetro pelo qual medimos sua qualidade científica. As leis científicas estabelecem as relações entre eventos e objetos, medem os resultados que são enunciados como equações matemáticas. Kepler e as órbitas planetárias, ou Galileu e o plano inclinado, primeiro observaram os fenômenos e depois postularam relações matemáticas para explicá-los.

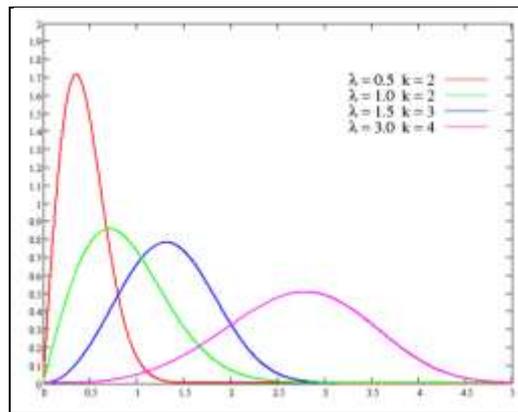
Do mesmo modo e baseados nos mesmos princípios, pensamos que se nossa fábrica se comportou de uma maneira ontem, e não houve mudanças, ela deverá se comportar de forma semelhante hoje. O que aprendemos ontem serve hoje porque a realidade é bastante parecida hoje, e podemos aproveitar esse conhecimento. Em termos dos processos, os dados que foram sendo obtidos sobre suas características podem ser utilizadas para comparar o passado com o presente. Enquanto não aparecerem “cisnes negros”, esperamos que os valores obtidos no passado sejam representativos do que está por ocorrer. Ao invés de ter presente a característica e os valores médios (mediana ou média) é mais instrutivo conhecer a distribuição dos valores históricos. O ponto de Jéssica é que, se investirmos no conhecimento da distribuição de uma variável aleatória relacionada com nossos projetos, podemos aplicar controle estatístico de processos<sup>142</sup>, abreviado SPC por suas iniciais em inglês (CEP em português).

Vamos pegar uma das variáveis associadas a determinada tarefa e observar seu comportamento. Para fixar ideias, tomamos o esforço que representa a construção de casos de teste por unidade do tamanho do caso de uso nos quais se baseiam. Derivando o histograma, vemos que toma uma forma parecida a uma distribuição normal, talvez um pouco mais voltada à direita que à esquerda. Revisando na literatura de estatística, a forma mais parecida é a  $\chi^2$ , uma distribuição derivada da curva normal de Gauss. Seguindo nossa busca, vemos que há uma família completa de curvas semelhantes, a família Weibull (ver Fig. 10.4), e que Putnam<sup>143</sup> utilizou em suas pesquisas sobre os processos de *software*.

---

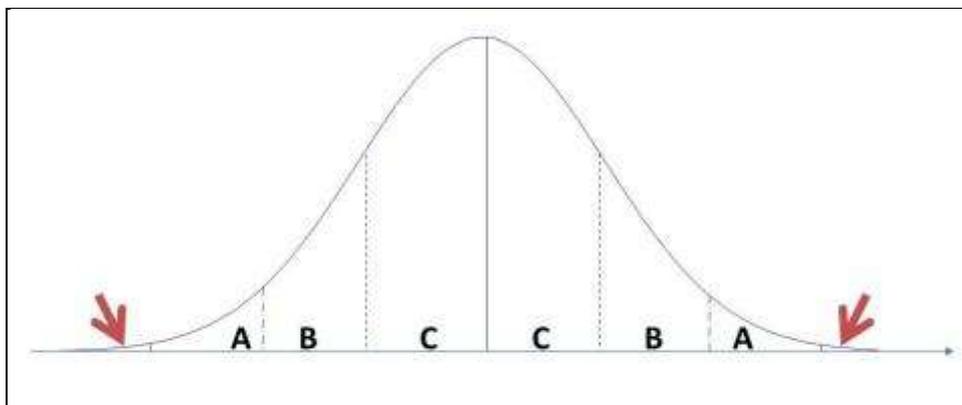
<sup>142</sup> [SHEWHART, 1939]

<sup>143</sup> [PUTNAM e MYERS, 2003]



**Figura 10.4: Curvas da Família Weibull**

Aplicar SPC teria sido útil no projeto fora de controle, porque nos teria mostrado que a duração de certas atividades estava fora de seus limites normais. Isto implica que conhecemos quais são esses limites. Digamos que entramos em um clube de *bridge* e nos aproximamos de uma mesa onde há dois casais jogando. Observando as cartas, vemos que cada um tem na mão treze cartas do mesmo naipe, o que obviamente desperta nossas suspeitas. No nosso conhecimento da distribuição que se produz ao embaralhar e distribuir as cartas de um baralho, a probabilidade de que ocorra essa exata distribuição é exatamente igual à de todas as outras mãos, mas é tão particular que nos surpreende vê-la. Como isto tem um impacto muito alto no resultado do jogo, suspeitamos que este fato não seja normal. Quer dizer, “normal” e “anormal” são julgados em termos estatísticos. Se sobre o sino de Gauss, que é a representação da curva normal, desenhamos zonas correspondentes à distância da média a um, dois e três desvios padrão<sup>144</sup>, e chamamos essas zonas de A, B e C, como na Figura 10.5, veremos que os dois extremos assinalados com setas têm muito baixa probabilidade de ser parte de uma amostra pequena da variável, já que a cada cem vezes que observamos a variável podemos encontrar um valor nessas zonas (de fato, a probabilidade de que isso ocorra naturalmente é inferior a 0,3%). Portanto, se as coisas são “normais”, não esperaríamos que isso ocorresse. Ao contrário, se vemos um valor fora das zonas A, B ou C, suspeitamos que algo anormal está ocorrendo, que o processo está nos “dizendo” algo. É por isto que a máxima é “precisamos escutar a voz do processo”.



**Figura 10.5: Zonas de SPC Sob a Distribuição Normal**

Se isto tivesse sido feito nas primeiras etapas do projeto fora de controle, teriam sido detectadas as anomalias e medidas paliativas ou corretivas poderiam ter sido tomadas.

As zonas A, B e C servem, além disso, para outros usos. Dada a característica aleatória das variáveis, os valores obtidos do mesmo processo não são sempre idênticos. A variação que consideramos aceitável é rotulada de “normal” e ignoramos o “ruído” que produz. Shewhart percebeu que o processo às vezes grita (quando os 3 desvios padrão são ultrapassados) e às vezes sussurra. As regras complementares à do ponto além da zona de controle são múltiplas e foram modificadas desde que Shewhart as gerou para a Western Electric. São as seguintes:

<sup>144</sup> Medida da dispersão de uma variável ao redor da média de sua distribuição.

- 2 de 3 pontos consecutivos na zona A, que é similar ao caso anterior, já que a probabilidade de que isto aconteça é inferior a 0,0625%.

- 6 pontos consecutivos em linha ascendente ou descendente, já que isto também tem uma pequena probabilidade (probabilidade das rajadas) é considerado que o sistema segue uma tendência não aleatória.

- 9 pontos consecutivos de um lado da linha central (seja por cima ou por baixo dela). Este caso costuma indicar um deslocamento da média, geralmente devido a uma mudança significativa no sistema. Pode ser uma boa notícia, por exemplo que aumentou a produtividade, mas, de todo modo, é preciso explicá-lo.

- 14 pontos consecutivos alternando acima ou abaixo, o que pode indicar um fenômeno cíclico ou séries temporais, ou que estamos na presença de duas populações.

- 15 pontos consecutivos na zona B ou C: isto implica na melhoria da precisão e um desvio padrão associado menor. De novo, em geral, é uma boa notícia.

- 4 de 5 pontos consecutivos na zona B ou além.

- 8 pontos consecutivos acima e abaixo da zona C indica que temos duas populações diferentes.

Definitivamente, prestar atenção ao comportamento do processo permite tomar decisões antecipadas, ao poder separar o “ruído” da “mensagem”. A tabela de riscos é completada com a última fila como mostra a Tabela 10.2: A Última Fila da Análise depois de Completa

5	O atraso no <i>sprint</i> da arquitetura foi visto como um bom sintoma, como se o tempo investido na análise fosse se pagar somente no futuro quando, na realidade foi interrompido abruptamente porque se sentiu que continuar era melhor que esperar para entender bem o problema do usuário	Não há um conhecimento profundo do comportamento dos processos e, em consequência, a superstição se impõe sobre a realidade	Aplicar SPC aos processos críticos
---	--	---	------------------------------------

Tabela 10.2: A Última Fila da Análise depois de Completa

### 10.9 Correlação e Regressão

Se a T<sup>2</sup> tivesse utilizado SPC, poderia ter detectado as anomalias no começo do projeto. Esse uso já justifica a análise, mas depois que possuímos esse conhecimento é tentador poder usá-lo mais extensamente. Considerando que a massa dos dados nos anos em que foram acumulados na T<sup>2</sup> é muito interessante, Jéssica propõe que sejam identificadas e analisadas as tendências. Cláudio se familiarizou com as técnicas de inteligência de negócios<sup>145</sup> e propõe contratar um doutorando com quem compartilha alguns cursos e sabe que é especialista no tema. É assim que, temporariamente, Damián é incorporado à equipe da T<sup>2</sup>.

Damián segmenta os dados em populações diferentes e produz uma análise detalhada de cada uma. Monta o que se chama “hipercubos” de dados, que vai submeter a métodos estatísticos. A ideia é que uma base de dados relacional em 5ª forma normal é muito lenta para realizar as milhares de operações SELECT que demanda a análise, por isso há um processo de extração que separa os dados e monta uma super tabela, o cubo em questão. Sobre esse cubo, faz diversas operações: limpa os dados, analisa correlações e fatores mediante análise da variação e análise fatorial; e constrói equações de regressão que vinculam os valores de seus cubos. O resultado desses esforços é um conjunto de modelos que predizem o

<sup>145</sup> Denomina-se inteligência empresarial, inteligência de negócios ou BI (do inglês *business intelligence*) o conjunto de estratégias e ferramentas focadas na administração e criação de conhecimento mediante a análise de dados existentes em uma organização ou empresa. [http://pt.wikipedia.org/wiki/Inteligência\\_empresarial](http://pt.wikipedia.org/wiki/Inteligência_empresarial)

comportamento de variáveis do projeto a partir de certos dados, que são obtidos em etapas iniciais.

### 10.10 Identificação de processos críticos

Por causa da grande massa de dados que possui a  $T^2$ , é possível tirar conclusões sobre eles usando *data mining*, como fez Damián; mas, nos casos em que esses dados não existam ou sejam insuficientes, a construção trabalhosa de dezenas de medições é pouco operacional. Mesmo quando os dados existem, é possível que haja um excesso de informação, que pode ser tão paralisante quanto sua falta<sup>146</sup>. Como diria Stephen Covey, o principal é se assegurar que o principal é o principal<sup>147</sup>. Mas, como?

Se lembrarmos do modelo de Kano (Figura 10.6) que vimos no Capítulo 7, a satisfação do cliente é o principal objetivo da empresa. Em primeiro lugar, deveríamos assegurar que não há requisitos obrigatórios que não tenham sido cobertos. Isto significa que devemos fazer um esforço grande para conhecer as necessidades de nosso cliente, além das que são verbalizadas nos requisitos, como já vimos no Capítulo 8 ao discutir os resultados de DRE. É necessário saber que o que o cliente pede, o que quer e o que necessita não são a mesma coisa. Faça uma lista então com as necessidades de seu cliente, identifique quando tiver satisfeito suas expectativas e onde ainda não, mas não tente medir “sucessos” e “fracassos”, mantenha seu foco nas expectativas e use o modelo de Kano para classificar os requisitos. Este conhecimento deveria nos permitir identificar os eventos “críticos”. Um evento “crítico” é um instante no qual o usuário de um produto ou serviço forma uma opinião sobre a qualidade ou o seu valor. Por exemplo, um viajante atraído pela publicidade no caminho escolhe um restaurante de *fast food* para comer, mas, ao se aproximar, o cliente passa por vários eventos críticos. Para começar, se o restaurante está do mesmo lado de seu caminho ou se precisa deixar a rodovia e cruzar para entrar no estabelecimento (do ponto de vista egocêntrico do condutor isso é um evento externo); de imediato, a acessibilidade do estacionamento; sua limpeza, já que é um indicador do que pode encontrar dentro do restaurante; sua percepção da segurança do estacionamento, já que vai deixar o carro com objetos de valor, não vai descer com a bagagem; a disponibilidade de lugares no estacionamento, já que esperar na estrada não é agradável. Tudo isto é considerado pelo cliente sem fazer uma análise de decisões formal.



Figura 10.6: Diagrama do Modelo de Kano

Em produtos e serviços de *software*, o equivalente a estes eventos críticos podem ser diferentes de cliente para cliente. Mas é preciso considerar: (a) os defeitos, como as diferenças com os requisitos, em sua severidade e número; (b) a adequação e conveniência de uso do

<sup>146</sup> <http://www.thedailybeast.com/newsweek/2011/02/27/i-can-t-think.html> na *Newsweek* faz referência a investigações a respeito disso conduzidas por Angelika Dimoka, diretora do *Centre for Neuronal Decision Making* da Universidade de Temple (Pensilvânia)

<sup>147</sup> [COVEY, 1996], *First Things First*.

sistema às necessidades; (c) sua facilidade de uso, o complicado ou não da aprendizagem; (d) o tempo que demora até sua entrega efetiva e com qualidade, talvez não só o tempo acumulado das demoras, mas também a quantidade de vezes que foi prometida a entrega e não foi cumprida; (e) a facilidade de instalação (transição da antiga versão ou do produto anterior); (f) o volume de mudança nos processos; (g) a migração de dados; (h) os custos únicos, por exemplo, de uma licença (COTS - *Commercial-Off-The-Shelf Software*), ou os custos de desenvolvimento em contratos de tempo e materiais, ou de ajustes em MOTS (*Modifiable Off-The-Shelf Software*), também com considerações específicas; (i) o suporte, como o pessoal dedicado e a curva de aprendizagem de sua própria equipe, se o suporte é transferido ao cliente; todas essas considerações podem afetar a percepção de qualidade do cliente. Isto constitui o que se denomina CTCs (*Critical to Customer*).

O próximo passo é identificar características mensuráveis de um produto, serviço ou processo que são chave, de modo que os limites de especificação ou padrões de desempenho devam ser cumpridos para obter a satisfação do cliente, que na literatura é denominada CTQs (*Critical to Quality*). Para transformar as necessidades e desejos do cliente em requisitos mensuráveis que possam ser implementados e controlados na empresa, necessitamos de uma ferramenta, e essa é a Árvore CTQ. Para construir a árvore seguimos os seguintes passos: Escutar a voz do cliente (VOC); Identificar Defeitos em Produtos e Serviços; Definir as Unidades de Produtos e Serviços; e Detalhar as Oportunidades para Produtos e Serviços. Por exemplo, vamos pegar um incidente registrado em nossa base de dados: “Cada vez que peço uma mudança no produto, tenho que esperar seis semanas para que me respondam.” O incidente passa a ser um evento crítico e queremos resolvê-lo. Devemos primeiro identificar a variável CTQ, neste caso, o tempo de resposta a pedidos de mudança. A medição que devemos usar ou criar é o tempo de resposta a um pedido de mudança, medido em dias desde que aparece em nossa base de pedidos de mudança até que o usuário receba a resposta, não antes. Falando com clientes, descobrimos que parece razoável a eles um tempo de resposta que não exceda três dias úteis. Agora dispomos de um sinal para medir quando um pedido de mudança tem um tempo de resposta que excede os três dias. Neste ponto devemos identificar o que leva os processos a demorar mais de três dias. A Figura 10.7 mostra um gráfico que exemplifica nossa análise.

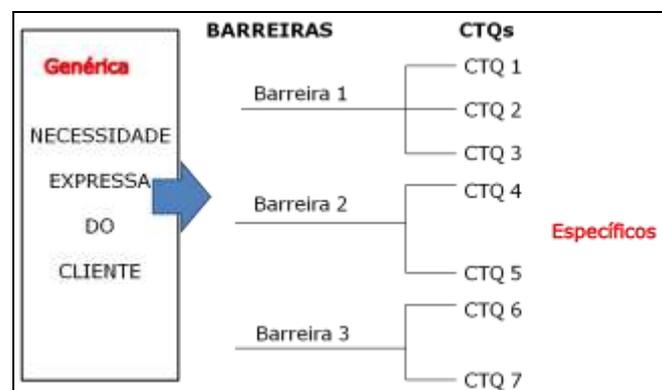


Figura 10.7: Barreiras à Qualidade

A partir de uma necessidade genérica expressa pelo cliente, identificamos os processos que estão sendo uma barreira para a satisfação do cliente e dos processos CTQs relacionados que, especificamente, precisam ser melhorados para os obstáculos serem eliminados. Um exemplo, de novo com uma cadeia de restaurantes, poderia ser que esta vem recebendo queixas do mau atendimento de seus funcionários em vários locais. Uma análise dos incidentes indica que os clientes (VOC) se queixam de ter de esperar até que sejam atendidos ao entrar no estabelecimento, e são ignorados por muito tempo; também precisam esperar muito tempo até que uma mesa fique limpa quando não há mesas vazias ao chegar; e que os atendentes não são amáveis, nem profissionais, mas muito distantes. A análise poderia dar o seguinte resultado (Figura 10.8).

Os retângulos representam processos, os objetos ovais, objetivos de negócio traduzidos a objetivos de processos, e as setas, mudanças sugeridas. Notem que “Tratamento Amável Segundo Processo” foi traduzido em uma série de objetivos que não cobrem as necessidades do cliente. Isto é típico de uma incapacidade para traduzir em objetivos mensuráveis certos comportamentos. Já recomendamos o livro de Mager em capítulos anteriores e voltamos a

fazer isso aqui. Estão faltando dois objetivos observáveis: olhar nos olhos do cliente quando se dirigir a ele e sorrir ao falar. Estes dois objetivos são mensuráveis já que são observáveis e uma lista de verificação consegue fazer essa avaliação.

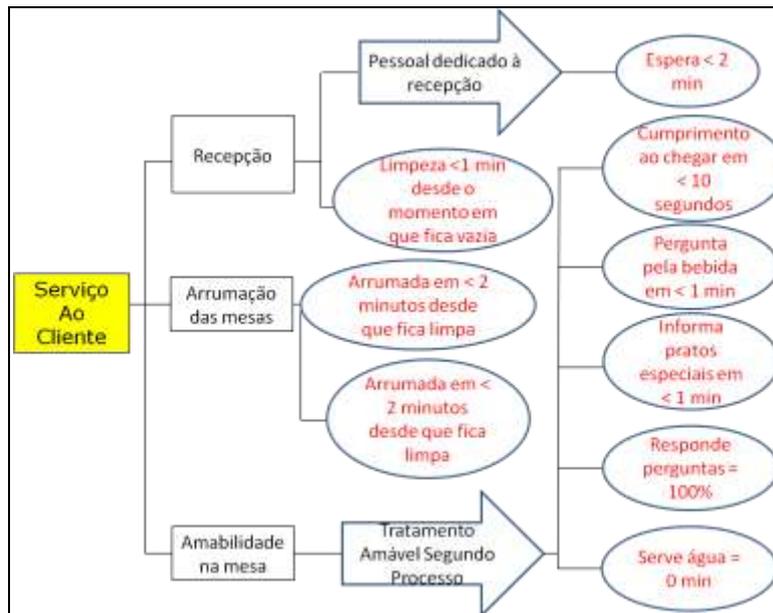


Figura 10.8: Análise de Fatores CTQ

Este processo é equivalente ao que introduziu Jéssica no comitê de gestão em uma de suas primeiras intervenções, o BSC (*Balanced Score Card*). Os passos são semelhantes, são identificados objetivos de negócio e são traduzidos a objetivos derivados até que seja obtida uma clara associação com processos. Por exemplo, partindo de reter os clientes de nosso portfólio, é possível traduzir este objetivo em vários objetivos derivados de melhorar a interface com o usuário, diminuir o número de defeitos entregues, acelerar o processamento de certos dados, melhorar o tempo de descarga e instalação, etc.

	Objetivos	Medições	Objetivos	Iniciativas	Resultados
<b>Satisfazer Acionistas</b>	Aumentar clientes	Número de clientes	% de incremento		
	Aumentar tamanho de pedidos	Venda média por cliente	% de incremento		
	Aumentar frequência de vendas	Tempo médio entre vendas a cliente	% de incremento		
	Aumentar entradas por novos produtos	Número de Produtos Novos % Entradas x Novos Produtos	% de incremento		
<b>Deleitar Clientes</b>	Aumentar a satisfação do cliente	Satisfação do cliente	% de incremento		
	Reduzir queixas de clientes	Número de Queixas	% de redução		
	Aumentar referências	Referências	% de incremento		
	Aumentar a frequência de vendas	Frequência de Vendas	% de incremento		

Figura 10.9: BSC Aplicado a Identificar Processos Críticos

Estes objetivos podem estar associados facilmente com processos, possivelmente mais de um, e, para cada um deles, podemos, por sua vez, identificar objetivos. Por exemplo, poderíamos associar a quantidade de defeitos entregues com o processo de inspeções, e colocar um objetivo de eliminar, ou diminuir significativamente a porosidade<sup>148</sup> das inspeções. Isto levaria à exploração de várias alternativas para detectar mais defeitos em cada inspeção, seja para

<sup>148</sup> A porosidade de um processo de *software* é a porcentagem de defeitos que se deixa escapar sobre o total que deveria ter sido encontrado. Obviamente, a porosidade não pode ser calculada *in processu*, é necessário contar com dados dos defeitos que escaparam e acabaram sendo encontrados em processos posteriores. Os defeitos latentes são os que nunca são encontrados.

melhorar as listas de verificação, treinar melhor o pessoal ou envolver pessoal com mais experiência. Conhecida a distribuição dos valores associados ao processo, como a densidade de defeitos encontrados, podem ser fixados objetivos que expressem as melhorias como modificações aos parâmetros da distribuição, como diminuir o valor médio, ou diminuir o valor do desvio padrão, o que significa um aumento na precisão da estimativa.

Por último, uma técnica para identificar processos críticos é construir modelos de regressão com as variáveis conhecidas e analisar a sensibilidade do modelo usando diagramas de tornado. Aqueles que demonstram maior sensibilidade são críticos e devem ser melhor controlados.

### 10.11 Processos Capazes

Depois de estabelecido o objetivo a alcançar, é útil perguntarmos se podemos modificar o processo para que este seja alcançado. Dado nosso conhecimento da variação, já não podemos enunciar objetivos em termos de um só número: devemos especificar os limites que consideramos como definidores da zona desejável para os valores da variável. Por exemplo, podemos definir um objetivo como “a densidade de defeitos detectada pelos testes de sistema deve ser de 0,1 defeito por ponto de função (PF), mais ou menos 0,001 defeito PF”. Claro, os números escolhidos são arbitrários e sumamente ambiciosos, mas de nada vale fixar objetivos que não sejam assim. Por exemplo, um objetivo de 10 defeitos por PF mais ou menos 7 defeitos por PF é tão pouco ambicioso que o efeito pode ser que as equipes diminuam sua capacidade.

Definimos um processo como “estável” se a variação é aceitável para os propósitos do negócio. Claro, somos conscientes de que esta não é uma definição operacional e que é muito subjetiva para ser útil. No entanto, como todos os processos mostram variação, qualquer definição que fixe valores é arbitrária e o único juízo que podemos propor é o da adequação. Visto da perspectiva do SPC, um processo estável é aquele observado dentro dos limites de controle. Um conceito afim, o do processo capaz, está vinculado às necessidades do negócio impostas pelo cliente. Dados os objetivos fixados a partir das necessidades do cliente, estabelecemos outros limites ao processo, não de controle, mas de especificação. Já falamos da voz do processo, estes limites agora estão associados à voz do cliente. Na realidade, há uma grande quantidade de vias para identificar os processos críticos. A Figura 10.10 ilustra alguns deles.

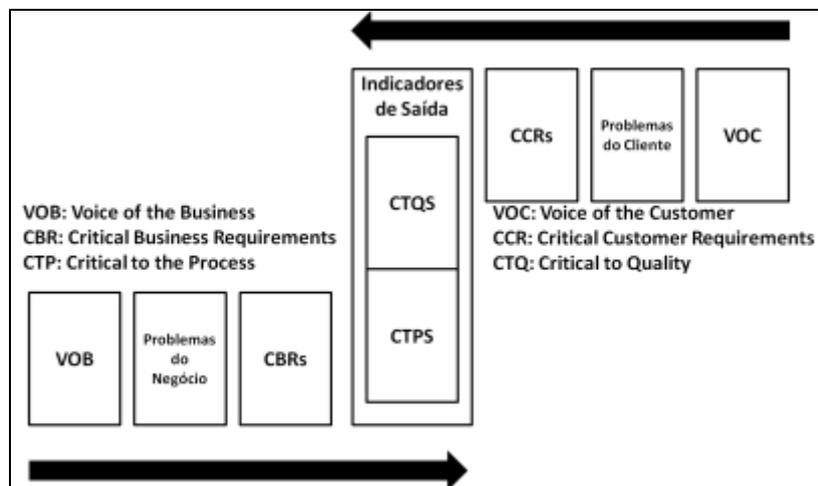


Figura 10.10: Fatores Na Escolha de Processos Críticos

Não são só as necessidades dos clientes que são importantes, há outras variáveis, como custos e utilização de recursos que podem determinar processos críticos. Desta forma é recomendável não ignorar a qualidade sob pena da empresa não sobreviver no mercado. O processo que melhor se adapta às necessidades do negócio e do cliente é aquele que é capaz e estável. De fato, podemos colocar em dúvida que um processo não estável seja capaz porque seria impossível demonstrar. A Figura 10.11 mostra a relação entre ambas as características. Na figura, UCL significa Limite de Controle Superior, do inglês *Upper Control Limit*; LCL é o Limite de Controle Inferior, por *Lower Control Limit*; USL o Limite de Especificação Superior; e LSL o Limite de Especificação Inferior. Na metade direita, é mostrado um processo capaz e estável, porque os limites de controle estão compreendidos pelos limites

de especificação, enquanto que a metade esquerda mostra um processo estável, mas não capaz. Se os requisitos do cliente não podem ser modificados, não resta alternativa a não ser modificar o processo para que seus limites de controle sejam reduzidos.

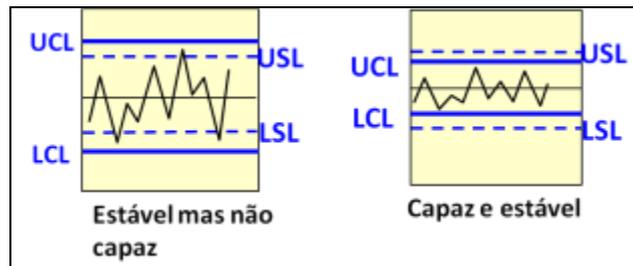


Figura 10.11: Processos Capazes e Processos Estáveis

### 10.12 Baselines

Os processos críticos devem ser controlados mais de perto que os outros. Se nos primeiros níveis do MPS é suficiente verificar o progresso de maneira global e em pontos pré-determinados do projeto, como os marcos de mudança de fase, e nos níveis G e superiores é indispensável verificar o avanço por tarefa, nos níveis B e A o controle recai no SPC. Para isso, é necessário contar com os dados necessários para entender os limites de controle naturais dos processos, em particular os críticos. Chamamos essa caracterização de “*baseline* de desempenho” e é a pedra fundamental da alta maturidade. Cada ponto ingressado no diagrama de controle de SPC permite tirar conclusões sobre o comportamento do processo. Por isso, Damián segmentou os dados em populações diferentes, estratificou-os quando diferentes classes mostravam diferentes comportamentos, depurou-os para eliminar os dados mal coletados ou que respondiam a situações excepcionais e construiu a *baseline* de desempenho de tudo o que serve para a gestão quantitativa dos processos críticos.

### 10.13 Indicadores Líderes

Por causa da relação entre os processos iniciais que são explorados com diferentes meios, Damián descobriu que vários processos servem de alerta inicial para saber se um projeto está orientado para cumprir seus objetivos. A noção de um indicador que permite antecipar resultados de um processo posterior é fundamental para a gestão quantitativa de um projeto. Digamos que decidimos que, para aumentar nossa participação no mercado, devemos reduzir os defeitos que chegam ao cliente e que, para isto, é preciso aumentar o esforço em inspeções e conseguir melhores inspetores e maior número. Com isso, pensamos que podemos diminuir a porcentagem de defeitos filtrados e, desse modo, diminuir os defeitos no cliente.



Figura 10.12: Indicadores Líderes

Os modelos de regressão que Damián construiu a partir dos dados são os que nos permitem estabelecer estas hipóteses. Um indicador é sempre um indicador de resultado<sup>149</sup>: mede o que aconteceu, nenhum pode medir o que vai acontecer. Mas alguns indicadores podem ser

<sup>149</sup> Também chamados indicadores de efeito, e em inglês, *lag indicators* ou *outcome measures*.

indicadores de causa<sup>150</sup> porque seus valores servem para prever o resultado das ações que se encontram mais adiante no fluxo do processo. Justamente esses modelos de regressão permitem estabelecer intervalos dentro dos quais se encontrará uma variável futura a partir do conhecimento do valor atual de um indicador de desempenho. Isto nos permite corrigir o rumo se for necessário. Por exemplo, Damián encontrou em seus dados uma relação que modela a duração da construção de um requisito como predictor da duração do período de testes. Com esse modelo teria sido possível prever nos primeiros momentos que o projeto estaria fora de controle.

#### 10.14 Composição do Processo Definido do Projeto

Enquanto a T<sup>2</sup> usava qualitativamente suas ferramentas da BiPro, os guias de ajuste cumpriam bem o propósito de adaptar o processo padrão às necessidades de um projeto em particular. Mas a introdução das ferramentas quantitativas, as *baselines* e os modelos, tornam necessário que seja revisado o processo de seleção dos componentes. O que antes era qualitativo agora deve ser selecionado com atenção aos objetivos da empresa. Os objetivos fixados na reunião mensal são traduzidos um a um para cada *sprint*, e Damián, a partir de seus modelos e das *baselines* do desempenho que são aplicadas, roda simulações que permitem oferecer às equipes as alternativas possíveis, uma ou mais de uma, para alcançar os objetivos. Isto implica que não há uma estratégia dominante, quer dizer que não há uma única forma de fazer as coisas.



Figura 10.13: Diferentes Possibilidades de Composição do Processo

Vamos imaginar que há três diferentes métodos para capturar requisitos, as Entrevistas Individuais, JAD e RAD, e que cada um tem sua *baseline* de desempenho em termos de requisito individual, com sua distribuição para custo, duração e qualidade. O mesmo ocorre com os dois métodos de *Design*, os quatro de Codificação e os três de Testing. Dependendo do que estivermos procurando otimizar, o caminho a seguir será diferente, por exemplo JAD é o que melhor qualidade tem, mas é mais caro que RAD ou as Entrevistas. Damián tem então de onde tirar dados para rodar suas simulações de Montecarlo e gerar resultados alternativos que permitam à equipe interessada fazer a escolha que melhor se adéqua a suas necessidades.

Se não houvesse nenhuma alternativa que permitisse alcançar os objetivos adequadamente, o comitê seria convocado para revisá-los ou para dar uma dispensa à equipe, ou para tentar, de qualquer modo, alcançá-los com um processo que não tem uma alta probabilidade de fazer isso. Neste último caso, são acrescentados aos riscos os processos que têm mais impacto no resultado e são procuradas alternativas que permitam mitigar esses riscos.

#### 10.15 Gestão Quantitativa

Equipados com um processo que deverá funcionar na maioria dos casos, as equipes simplesmente precisam executar as tarefas, medir os resultados e controlá-los mediante SPC para que a gestão seja levada adiante. Não há mais necessidade de nada e o projeto é controlado sozinho enquanto os dados não mostrarem anomalias. Se há decisões a serem tomadas frente a mudanças de qualquer porte, o *Scrum Master* ou o programador-chefe pode reusar o modelo, com ou sem a ajuda de Damián, para estimar o efeito de suas decisões. Isto

<sup>150</sup> Também chamados indicadores indutores, e em inglês, *lead indicators* ou *performance drivers*.

é assim porque os modelos possuem variáveis que estão sob controle dos que tomam as decisões, por exemplo a experiência média dos participantes em certas tarefas, a quantidade de inspeções que são introduzidas no processo, as estratégias de teste e outras que, ao poderem ser modificadas, permitem fazer ajustes no projeto durante sua execução por meio de modificações no processo.

### 10.16 A Melhoria Contínua Como Estratégia de Negócio

Para a T<sup>2</sup> o objetivo é ser cada vez melhor. Não só ser a melhor de todas as empresas em seu ramo, mas melhor que ela mesma, ontem. Desde seu humilde início, a empresa não fez mais que melhorar. A qualidade que seus produtos mostram e sua busca incessante pela superação resulta em um diferencial competitivo que é usado para ganhar e reter clientes. Cada passo em seu caminho para a excelência é difundido pelos canais tradicionais e os prêmios obtidos ajudam continuamente à conquista de mercado.

O conhecimento profundo que adquiriu de seu funcionamento não só permite prever os resultados com maior precisão que seus concorrentes, mas também permite escolher os clientes. Aqueles que não oferecem segurança nas margens são passados graciosamente aos “perseguidores de cota de mercado”, como Cláudio os chama. Este conhecimento e sua versatilidade com os métodos ágeis são o fruto direto de decisões estratégicas que permitem desenhar novas linhas de produto com alta probabilidade de sucesso. Os riscos de territórios desconhecidos e domínios sem dono são conhecidos e eles aprenderam a navegá-los. Suas análises de portfólio passaram a ser verdadeiras análises financeiras com profusão de dados de mercado, probabilidades e simulações.

### 10.17 Custo de Competir

Ainda mais importante para a T<sup>2</sup> é que seus concorrentes têm sérios problemas para poder entrar nos mercados que ela atua. Seus métodos ágeis e sua perfeita posição no mercado faz com que arrisquem margens com frequência, com as consequentes perdas ocasionais e dores de cabeça. A qualidade da T<sup>2</sup> aumentou o custo de entrada de seus concorrentes.

Além disso, a T<sup>2</sup> pode conseguir os melhores recursos humanos no mercado porque seu mito é muito firme. Trabalhar para a T<sup>2</sup> é um orgulho para os profissionais e é considerada uma excelente escola de negócios. Se alguém recebe uma oferta da T<sup>2</sup> é pouco provável que a recuse.

### 10.18 Inovação tecnológica

No entanto, a T<sup>2</sup> tem claro que não se pode dormir sobre a fama. Como parte de seu programa de introdução à empresa, os ingressantes recebem instrução no projeto de Escuta Tecnológica. Esse projeto, filho de Mariano e Alfredo, que o comparam ao SETI<sup>151</sup>, consiste na busca constante de novas ideias e tecnologias de aplicação. Cada engenheiro da T<sup>2</sup> escolhe uma publicação acadêmica, científica ou de divulgação, como *Scientific American*, *CACM*, *IEEE Computer*, *IEEE Software* ou *Discover*, e a empresa paga sua assinatura e os custos de associação, se for o caso. Em troca, o engenheiro precisa contribuir com pelo menos uma nota sobre o que leu na revista da T<sup>2</sup>, chamada *SPI*<sup>152</sup> *Glass*. No começo do programa, quando os engenheiros eram contados por unidades, publicava-se tudo que era enviado. Hoje, com os números aproximando-se dos mil nas três sedes, há um comitê de seleção de materiais, que é uma tarefa de tempo integral.

Várias ideias germinaram a partir da leitura dos materiais. Uma maneira inovadora de captar requisitos mediante protótipos em papel passou de inquietude a proposta de melhoria a ser testada em um projeto e completou seu ciclo quando foi adotada como uma técnica na BiPro. A incorporação destas mudanças aos processos sofreu transformações também. Antes, os pedidos e sugestões de mudança eram tratados em reuniões para colocar prioridades e serem aprovados, postergados, rechaçados ou dar um tratamento piloto. Hoje os dados e modelos estatísticos governam as decisões. O comitê de gestão já não pergunta “Como vai o projeto?” ou “Como podemos aproveitar isso?”, mas é muito mais preciso: “Qual é a probabilidade de que este projeto cumpra com seus compromissos?” e espera uma resposta numérica com um gráfico que o acompanhe. Para o caso dos processos, o que se pergunta é “Como são

<sup>151</sup> <http://www.seti.org/>. SETI é um projeto de busca de vida extraterrestre.

<sup>152</sup> *Software Process Improvement, Spy Glass* é o nome da luneta em inglês.

modificadas nossas margens no futuro, adotando essa mudança?” e “Qual é o valor presente desse investimento?” Marcela não apresenta nada ao comitê que não esteja acompanhado de um plano piloto e dos dados financeiros que preparou com Cláudio sobre as simulações que Damián rodou. Quando se justifica, apresenta uma árvore de decisões.

Desse modo, foram incorporadas mudanças a muitos processos, inclusive alguns que vêm da “noite dos tempos”, como os gêmeos chamam os dias nos quais os fundadores se reúnem ao redor de uma mesa de chapa de vinil para organizar suas ideias de *design* e arrematar as tarefas. A automação governa os processos, mas as ideias governam a automação. A T<sup>2</sup> aprendeu a reconhecer seu núcleo, mas também a explorar “o espaço em branco”<sup>153</sup>, aquelas oportunidades que não se encaixam com seus modelos de negócios ou com seus clientes habituais. Johnson define o “espaço em branco” como: “a gama de possíveis atividades que não estejam definidas ou dirigidas pelo modelo de negócios da empresa atual, quer dizer, as oportunidades fora de seu núcleo e além de suas adjacências que requerem um modelo de negócios diferente de exploração”.

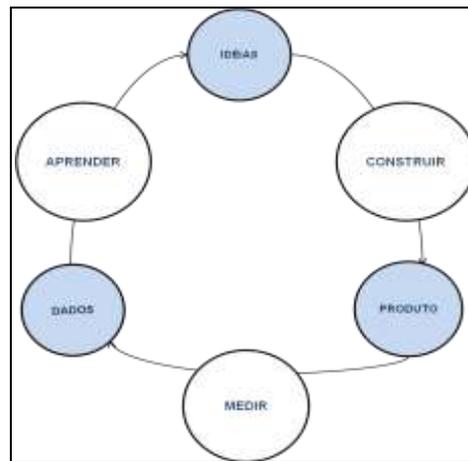


**Figura 10.14: Definição de Adjacências e Espaço em Branco Segundo Johnson**

Algumas das propostas apresentadas ao comitê são atendidas de uma maneira muito particular: cria-se um grupo de estudos que recebe orçamento do mesmo modo que um projeto, mas que não tem as mesmas obrigações. Por exemplo, não segue um ciclo de vida normal, utilizando em troca um que Jéssica adaptou do livro de Eric Ries, *The Lean Startup*<sup>154</sup>. Ries chama seu ciclo de retroalimentação Construir-Medir-Aprender (Ver Figura 10.15). O objetivo é acelerar a aprendizagem passando tão rápido quanto for possível por todos os passos do ciclo. Ao invés de construir um protótipo completo, o que se conhece como um “teste de conceito”, é construída uma maquete incompleta que é submetida de imediato ao teste de uso pelo cliente. Isto deve ser medido com base em critérios que foram fixados de antemão, como é o caso no *design* de novas funções em *software*. As reações dos usuários sugerem mudanças, ajustes, continuar pelo caminho e ampliá-lo ou simplesmente suspender o projeto. Seja qual for o resultado, a T<sup>2</sup> acha justificado o experimento porque a organização aprendeu.

<sup>153</sup> [JOHNSON, 2010], *Seizing the White Space: Business Model Innovation for Growth and Renewal*

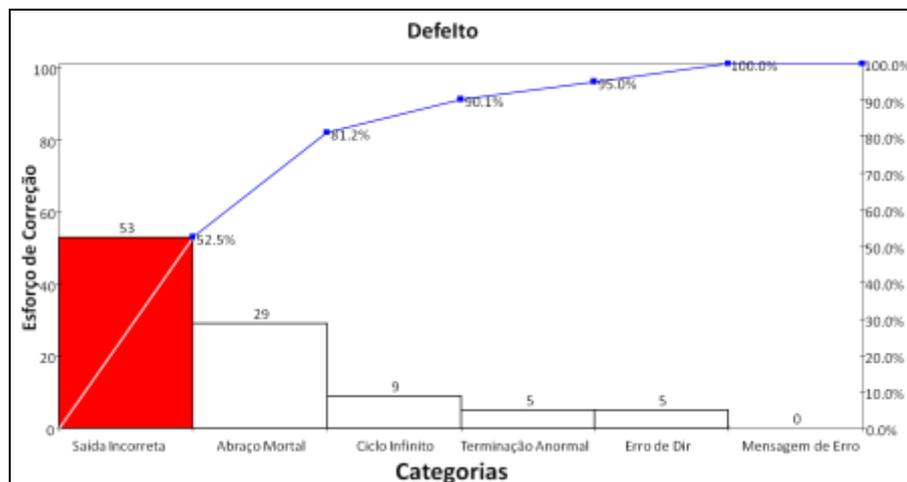
<sup>154</sup> [RIES, 2011], *The Lean Startup: How Today's Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses*



**Figura 10.15: Construir-Medir-Aprender**

Às vezes, os usuários fazem comentários que identificam defeitos grosseiros, mas não exatamente como resolvê-los. Para isso, são aplicadas as mesmas técnicas de análise de causa que usam os projetos de *software* e o grupo de qualidade de Marcela para identificar problemas, oportunidades e soluções, mas sem contar, como os anteriores, com a vantagem dos dados estatísticos. Já vimos várias técnicas de uso comum, começando na “noite dos tempos” com o diagrama de espinha de peixe e o método das oito disciplinas. Em duas oportunidades mencionamos a Lei de Pareto<sup>155</sup>, mas não mostramos uma aplicação particular que é feita na análise de causas-raiz. Quando um acontecimento merece, por exemplo um “cisne negro”, a análise de causa segue o método habitual de analisar as causas-raiz com especialistas, identificar soluções, estimar seu impacto, “vendê-las” à direção, implementá-las e medir o resultado. Mas em duas ocasiões no ano, o grupo de Marcela analisa as melhorias a implementar no semestre. As fontes de iniciativas são os grupos de estudo, as sugestões por meio do *SPI Glass* e as mudanças nos objetivos de negócios. Estes últimos sugerem melhorias em termos de margens a alcançar, para o qual um dos pontos de partida são os defeitos registrados.

Marcela e seu grupo classificam os defeitos e analisam seu impacto no retrabalho. Aqueles que exigem mais esforço são candidatos a seguir o processo de análise de causa. A Figura 10.16 mostra um diagrama usado nesta análise.



**Figura 10.16: Diagrama de Pareto de Defeitos de Código**

Os defeitos de cada categoria são contabilizados somando o esforço de correção individual em dias de trabalho. “Saída Incorreta” acumula 53 dias, que representa 52,5% do total do esforço investido em correções. Entre esta categoria e a que segue em importância acumulam 81,2% do total, um exemplo da lei do 80-20. É fácil tirar conclusões a partir do diagrama. Por exemplo, se conseguíssemos diminuir pela metade o número de defeitos da primeira categoria,

<sup>155</sup> Capítulos 2, na discussão de *Lean*, e 8, aplicando a análise de perfil operativo.

reduziríamos em média 26,5 dias de esforço de correção. Isto é traduzível em dinheiro disponível de modo imediato, pelo que se pode rapidamente ter uma ideia do volume que pode ser investido para eliminar os defeitos desse tipo. Os erros de direcionamento não têm esse mesmo peso, e, portanto, saem da análise.

Este método de experimentar novas ideias deu frutos muito interessantes. A T<sup>2</sup> gerou dois *spin-offs*: Uma linha de produto novo já é uma empresa própria, fabricando ambientes de desenvolvimento para aplicativos de telefonia inteligente; a segunda oferece uma série de serviços na nuvem para aplicativos de seguros com banda larga para fazer uso da imagem em sinistros, economizando milhares de horas de inspeção por mês. Um dos aplicativos desenvolvidos pela fábrica de *apps* é uma visão 3D de um objeto a partir de um filme de 360°. As empresas têm sinergia...

Mais interessante, no entanto, é o uso que deram a projetos que se aventuraram no “espaço em branco”. Em alguns casos, venderam as patentes, em outros iniciaram novas sociedades ou financiaram aqueles que fizeram o estudo, em troca de uma participação na nova empresa. A T<sup>2</sup> está preocupada com a doença do crescimento, a ancilose, e combate isso de todas as maneiras possíveis.

#### O QUE ACONTECEU ATÉ AGORA

89. Os valores típicos do projeto de SOA não correspondem à história dos projetos da T<sup>2</sup>.
90. Um projeto SOA novo entra em crise e sai do controle.
91. Jéssica introduz a SPC à T<sup>2</sup>.
92. Damián entra na equipe para realizar análise de dados com ferramentas de inteligência de negócios.
93. Damián segmenta os dados em populações diferentes e produz a *baseline* de desempenho de cada uma.
94. Com a ajuda de diversas técnicas são identificados os processos críticos para T<sup>2</sup>.
95. Definidos os processos críticos, a T<sup>2</sup> se assegura de que sejam estáveis e já tenham sua *baseline*.
96. Para os processos críticos, Damián gera modelos estatísticos de previsão para utilizá-los na análise de objetivos.
97. O sistema de medição é estendido para incluir indicadores líderes e intervalos de confiança.
98. A composição do processo definido do projeto passa a ser quantitativa, baseada em simulações Monte Carlo de seu desempenho.
99. A busca constante da excelência aumenta o custo de entrada para a concorrência da T<sup>2</sup>.
100. A inovação sistemática dá bons frutos.

---

## CAPÍTULO 11 - CONCLUSÕES

- **A já não tão pequena Tahini-Tahini**

Nossa história encerra aqui. Nossa amiga T<sup>2</sup> está considerando uma oferta astronômica de compra de duas de suas linhas de negócios, feita por uma dessas empresas gigantescas dos EUA. Os gerentes, ainda bem jovens, avaliam ideias sobre aposentar-se em ilhas do Pacífico Sul ou Fernando de Noronha. Considerando o caminho percorrido, merecem. Mas, por enquanto, é preciso resumir sua história para que outros possam aproveitar.

- **Lean como método de identificação da área de melhoria**

O uso de um método de identificação de oportunidades baseada na diminuição de falhas e o *shortening* dos prazos de entrega fizeram a empresa focar em negócios antes dos processos. Se bem que, enquanto melhoravam os processos, o objetivo sempre foi a melhoria da competitividade da T<sup>2</sup>. O preço nunca foi tão alto a ponto que os resultados não se justificassem. Em suma, o resultado é uma empresa de qualidade mundial que justifica sua proeminência nos fatos.

- **Métodos ágeis como ferramenta de melhoria**

Um dos fatores que aceleraram a maturação da T<sup>2</sup> como empresa, foi a escolha, no início de sua vida, de métodos ágeis. A iteração contínua acrescentou o conhecimento dos processos e sua execução. Com muitos dados nas mãos, as decisões foram mais simples e muito mais bem sucedidas. Ainda quando a crescente complexidade dos sistemas a serem desenvolvidos eliminaram o uso de *Scrum* e Kanban, a empresa continua sua adesão aos métodos ágeis, escolhendo FDD para atacar os novos desafios.

- **O MR-MPS-SW como marco do crescimento organizacional**

Uns dos aspetos mais fascinantes do MR-MPS-SW é sua funcionalidade como ferramenta de desenvolvimento organizacional. Em patamares muito acessíveis, desenha um caminho de crescimento que vai desde a desordem inicial à excelência global. Desde os primeiros passos, onde a reação é frequente e os erros são muitos, até que a qualidade impere e os clientes estão totalmente satisfeitos, o MR-MPS-SW acompanha as organizações que o adotam. O resultado nem sempre é tão bem sucedido como com a T<sup>2</sup>, mas aqueles que não tentam alcançar o topo não desfrutarão da vista.

---

---

**REFERÊNCIAS BIBLIOGRÁFICAS**

- ABNT, 2012, ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. NBR ISO/IEC 29110-4-1: Engenharia de Software - Perfis de ciclo de vida para microorganizações (VSEs) Parte 4-1: Especificações de perfil: Grupo Perfil Genérico, Rio de Janeiro: 2012.
- AMBLER, S. W., 2002, *Agile Modeling: Effective Practices for Extreme Programming and the Unified Process*, John Wiley and Sons.
- ANDERSON, D. J., 2011, *Kanban. Successful Evolutionary Change for Your Technology Business*. Blue Hole Press.
- ANDRIOLE, S., 1993, *Rapid Application Prototyping: The Storyboard Approach to User Requirements Analysis*, QED Technical Publishing Group.
- ARTHUR, J., 2004, *The Small Business Guerrilla Guide to Six Sigma*, LifeStar Publishing.
- ARTHUR, J., 2006, *Lean Simplified. The Power Laws of Speed*, LifeStar Publishing.
- ATWOOD J., 2006, *The Multitasking Myth, Coding Horror*. Disponível em: <http://www.codinghorror.com/blog/2006/09/the-multi-tasking-myth.html>.
- BECK, K., 2000, *Extreme Programming Explained*, Addison-Wesley.
- BOEHM, B., 1981, *Software Engineering Economics*, Prentice Hall.
- BOEHM, B., 1989, *Software Risk Management*, IEEE Computer Society Press.
- BOEHM, B. e TURNER, R., 2003, *Balancing Agility and Discipline: A Guide for the perplexed*, Addison-Wesley.
- BENNIS, W., 1997, *Learning to Lead: A Workbook on Becoming a Leader*, Addison Wesley.
- BORIA, J., 1987, *Ingeniería de Software*, Kapelusz (II EBAI).
- BORIA, J., 1989 *Construcción de Sistemas Operativos*, Kapeluz (IV EBAI).
- BORIA, J., 2010, *Don't Be On Time*. Disponível em: <http://www.slideshare.net/jorgeboria/dont-be-on-time>.
- BROOKS, F. P., 1995, *The Mythical Man-Month: Essays on Software Engineering*, Anniversary Edition (2nd Edition), Addison-Wesley.
- BROWN, A., 2010. Disponível em: <http://www.aaronmbrown.net/blog/2010/07/programmers-flow-and-productivity/>
- CAMERON, K., QUINN, R., 2011, *Diagnosing and Changing Organizational Culture: Based on the Competing Values Framework*, Jossey-Bass.
- CARL III, W. J., *Unpublished paper, Flow - A Theory of Optimal Experience: History and Critical Evaluation*.
- CHRISSIS, M. B.; KONRAD M. e SHRUM S., *CMMI for Development®: Guidelines for Process Integration and Product Improvement*, Addison-Wesley, 3ra edição.
- CLEMEN, R., 1997, *Making Hard Decisions: An Introduction to Decision Analysis*, Duxbury.
- COAD, P., DE LUCA, J., LEFEVRE E., 1999, *Java Modeling In Color With UML: Enterprise Components and Process*, Prentice-Hall.
- COCKBURN, A., 2000, *Writing Effective Use Cases*, Addison-Wesley Professional.
- COCKBURN, A., 2005, *Crystal Clear: A Human-Powered Methodology for Small Teams*, Addison-Wesley.
- COHN, M., 2006, *Agile Estimation and Planning*, Prentice-Hall.
- COVEY, S., 1996, *First Things First*, Free Press.
- CONNER, D., PATTERSON, R., 1982, *Building Commitment to Organizational Change*, Training and Development Journal, v36 n4 p18-26,28-30 Apr 1982.
- CULBERT, S., 2010, *Get Rid of the Performance Review!: How Companies Can Stop Intimidating, Start Managing--and Focus on What Really Matters*, Business Plus.

- CRISPIN, L. e GREGORY, J., 2009, *Agile Testing. A Practical Guide for Testers and Agile Teams*, Addison-Wesley.
- CSIKSZENTMIHALYI'S M., 1991, *Flow: The psychology of optimal experience*, Harper & Row.
- DECKER, B., RAS, E., RECH, E., KLEIN, B., HOECHT, C., 2005, *Self-organized Reuse of Software Engineering Knowledge Supported by Semantic Wikis*, Proceedings of the Workshop on Semantic Web Enabled Software Engineering.
- DE BONO, E., 1985, *Six Thinking Hats*, Little Brown and Company.
- DE MARCO, T. e LISTER, T., 1987, *Peopleware: Productive Projects and Teams*, Dorset House.
- DEMING, E. D., 2010, *Out of the Crisis*, The MIT Press.
- DENNEY, R., 2007, *Succeeding with Use Cases. Working Smart to Deliver Quality*, Addison-Wesley.
- DENNEY, R., 2012, *Use Cases Levels of Test. A Four-Step Strategy for Budgeting Time and Innovation in Software Test Design*, CreateSpace Independent Publishing Platform.
- DIAZ, M., e KING, J., 2002, *How CMM Impacts Quality, Productivity, Rework, and the Bottom Line*, Crosstalk, March 2002. Disponível em: <http://www.crosstalkonline.org/storage/issue-archives/2002/200203/200203-Diaz.pdf>
- EBENAU, R. e STRAUSS S., 1994, *Software Inspection Process*, McGraw-Hill.
- FLORAC, W. A. e CARLETON, A. D., 1999, *Measuring the Software Process*, Addison-Wesley.
- FOWLER, M., 2003, *UML Distilled: A Brief Guide to the Standard Object Modeling Language* (3rd Edition), Addison-Wesley.
- FREEDMAN, D. P. AND WEINBERG G., 1990, *Handbook of Walkthroughs, Inspections, and Technical Reviews*. Dorset House,
- FRIED, J., 2005, *An Analysis of the Correlation between System Engineering Defect Phase Containment and System Engineering Hours at General Dynamics C4 Systems*. Disponível em: <http://www.acims.arizona.edu/PUBLICATIONS/PDF/JenniferFriedMCSproject%205-21-05.pdf>.
- GAUSE, D. and WEINBERG G., 1989, *Exploring Requirements: Quality Before Design*. Dorset House.
- GILB T. e GRAHAM D., 1994, *Software Inspection*, Addison-Wesley Professional.
- GLASS R., 1997, *Software Runaways: Monumental Software Disasters*, Prentice Hall.
- GOULD S., 1996, *Dinosaur in a Haystack: Reflections in Natural History*, Three River Press.
- GRIES, D., 1987, *The Science of Programming*, Springer.
- HALL, E., 1998, *Managing Risk: Methods for Software Systems Development*, Addison-Wesley Professional.
- HIGHSMITH, J., 1999, *Adaptive Software Development: A Collaborative Approach to Managing Complex Systems*, Dorset House.
- HIGHSMITH, J., 2001, *Agile Alliance's Agile Manifesto, History and Contents*. Disponível em: <http://agilemanifesto.org/>
- HIGHSMITH, J., 2004, *Agile Project Management. Creating Innovative Products*, Addison-Wesley.
- HOFMEISTER, C., NORD, R., e SONI, D., 2000, *Applied Software Architecture*, Addison Wesley.
- HUMPHREY, W., 1989, *Managing the Software Process*, Addison Wesley.
- HUNT, A. e THOMAS, D., 1999, *The Pragmatic Programmer: From Journeyman to Master*, Addison Wesley Professional.

- ISO/IEC, 2003, ISO/IEC 15504 – *Software Engineering – Process Assessment*, The International Organization for Standardization and The International Electrotechnical Commission.
- ISO/IEC, 2008, ISO/IEC 12207 – *System and Software Engineering – Software Life Cycle Process*, The International Organization for Standardization and The International Electrotechnical Commission.
- JOHNSON, M., 2010, *Seizing the White Space: Business Model Innovation for Growth and Renewal*, Perseus Books Group.
- JONES, C., 1996 *Applied Software Measurement*, McGraw-Hill.
- JONES, C., 1994, *Assessment and Control of Software Risks*, Prentice-Hall.
- JONES, C., 1986, *Programming Productivity*, McGraw-Hill.
- JOSUTTIS, N., 2009, *SOA in Practice: The Art of Distributed System Design*, O'Reilly Media.
- KAPLAN, R., e NORTON, D., 1996, *The Balanced Scorecard: Translating Strategy into Action*, Harvard Business Review Press.
- KAN. S., 2002, *Metrics and Models in Software Quality Engineering*, 2nd Edition, Addison-Wesley Professional.
- KERNIGHAN B. W., e PLAUGER P. J., 1974, *The Elements of Programming Style*, McGraw-Hill.
- KNIBERG, H., 2007, *Scrum and XP from the Trenches. How we do Scrum*, C4Media, Publisher of InfoQ.com.
- KNIBERG, H. e SKARIN, M., 2010, *Kanban and Scrum - making the most of both*, C4Media, Publisher of InfoQ.com.
- KUBLER-ROSS, E., 1997, *On Death and Dying*. Scribner.
- KUZNETS, S., 1966, *Economic Growth and Structure. Selected Essays*, Heinemann.
- LADAS, C., 2008, *Scrumban and And Other Essays on Kanban Systems for Lean Software Development*, Modus Cooperandi Press.
- LEFFINGWELL, D., 2007, *Scaling Software Agility. Best Practices for Large Enterprises*, Addison-Wesley.
- LUCIA, A.D., LEPSINGER, R., 2007, *The Art and Science of Competency Models. Pinpointing Critical Success Factors in Organizations*, Jossey-Bass Pfeiffer.
- McFEELEY, B., 1996, *IDEAL<sup>SM</sup>: A User's Guide for Software Process Improvement*. Software Engineering Institute Handbook CMU/SEI-96-HB-001.
- McMAHON, P., 2011, *Integrating CMMI<sup>®</sup> and Agile Development*, Addison-Wesley.
- McGARRY, J.; CARD, D.; JONES, C.; LAYMAN, B.; CLARK, E.; DEAN, J. e HALL, F, 2002, *Practical Software Measurement: Objective Information for Decision Makers*, Addison Wesley.
- MEADOWS, D., 2008, *Thinking in Systems, A Primer*, Chelsea Green Publishing.
- MIRANDA, E., 2003, *Running the Successful Hi-Tech Project Office*, Artech House Publishers.
- MONDEN, Y., 2011, *Toyota Production System: An Integrated Approach to Just-In-Time*, 4th Edition, Productivity Press.
- MOREIRA, M., 2010, *Adapting Configuration Management for Agile Teams. Balancing Sustainability and Speed*, Wiley.
- MYERS, G., 1979, *The Art of Software Testing*, Wiley.
- NOLAN, R., 1973, *Managing the Computer Resource: A Stage Hypothesis*. CACM, Vol 16, No 7, July 1973, republished in *Managing The Data Resource Function*, same author, West.

- OKTABA, H. et al., 2005, *Modelo de Procesos para la Industria del Software MoProSoft, versión 1.3*.
- PALMER, S. R. e FELSING, J. M., 2002, *A Practical Guide to Feature-Driven Development*, Prentice Hall.
- PAULK, M., WEBER, C., E CURTIS, B., 1994, *The Capability Maturity Model: Guidelines for Improving the Software Process*, Addison Wesley.
- PIRSIG, R.M., 1974, *Zen and the Art of Motorcycle Maintenance, An inquiry into Values*, William Morrow and Co.
- PMI, 2008, *Project Management Institute. The Standard for Portfolio Management*. Syba: PMI Publishing Division.
- POPPENDIECK, M. e POPPENDIECK, T., 2010, *Leading Lean Software Development. Results Are Not the Point*, The Addison Wesley Signature Series.
- PUGH, S., 1991, *Total Design: Integrated Methods for Successful Product Engineering*. Addison-Wesley.
- PUGH, S. 1981, *Concept selection: a method that works*. In: Hubka, V. (ed.), *Review of design methodology*. Proceedings international conference on engineering design, March 1981, Rome. Zürich: Heurista.
- PUTNAM, L. H. e MYERS, W., 2003, *Five Core Metrics: The Intelligence Behind Successful Software Management*, Dorset House Publishing Company.
- RODIN, R. e HARTMAN, C., 2000, *Free, Perfect and Now: Connecting to the Three Insatiable Customer Demands, A CEO's true Story*, Free Press.
- RIES, E., 2011, *The Lean Startup: How Today's Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses*, Random House.
- ROYCE, W., 1970, *Managing the Development of Large Software Systems*, Proceedings of IEEE WESCON 26 (August): 1–9
- SCHWABER, K. e BEEDLE, M., 2002, *Agile Software Development with Scrum*, Prentice Hall.
- SCHWABER, K., 2004, *Agile Project Management with Scrum*, Microsoft Press.
- SCHUYLER, J., 1996, *Decision Analysis in Projects. Learn to Make Faster, More Confident Decisions*, Project Management Institute.
- SEI, 2010, *Capability Maturity Model Integration (CMMI) for Development, version 1.3*, Carnegie Mellon University, Software Engineering Institute, Technical Report CMU/SEI-2010-TR-033.
- SENGE P. M., 2006, *The Fifth Discipline: The Art & Practice of The Learning Organization*, Crown Business.
- SHEWHART, W., 1939, *Statistical Method from the Viewpoint of Quality Control*, Dover Books on Mathematics.
- SOFTEX, 2011a, ASSOCIAÇÃO PARA PROMOÇÃO DA EXCELÊNCIA DO SOFTWARE BRASILEIRO – SOFTEX, MPS.BR – Guia de Aquisição:2011, junho 2011. Disponível em: <http://www.softex.br>.
- SOFTEX, 2011b, ASSOCIAÇÃO PARA PROMOÇÃO DA EXCELÊNCIA DO SOFTWARE BRASILEIRO – SOFTEX., MPS.BR – Guia de Implementação – Parte 1: Fundamentação para Implementação do Nível G do MR-MPS:2011, junho 2011. Disponível em: <http://www.softex.br>.
- SOFTEX, 2011c, ASSOCIAÇÃO PARA PROMOÇÃO DA EXCELÊNCIA DO SOFTWARE BRASILEIRO – SOFTEX., MPS.BR – Guia de Implementação – Parte 2: Fundamentação para Implementação do Nível F do MR-MPS:2011, junho 2011. Disponível em: <http://www.softex.br>.
- SOFTEX, 2011d, ASSOCIAÇÃO PARA PROMOÇÃO DA EXCELÊNCIA DO SOFTWARE BRASILEIRO – SOFTEX., MPS.BR – Guia de Implementação – Parte 3:

- Fundamentação para Implementação do Nível E do MR-MPS:2011, junho 2011.  
Disponível em: <http://www.softex.br>.
- SOFTEX, 2011e, ASSOCIAÇÃO PARA PROMOÇÃO DA EXCELÊNCIA DO SOFTWARE BRASILEIRO – SOFTEX., MPS.BR – Guia de Implementação – Parte 4:  
Fundamentação para Implementação do Nível D do MR-MPS:2011, junho 2011.  
Disponível em: <http://www.softex.br>.
- SOFTEX, 2011f, ASSOCIAÇÃO PARA PROMOÇÃO DA EXCELÊNCIA DO SOFTWARE BRASILEIRO – SOFTEX., MPS.BR – Guia de Implementação – Parte 5:  
Fundamentação para Implementação do Nível C do MR-MPS:2011, junho 2011.  
Disponível em: <http://www.softex.br>.
- SOFTEX, 2011g, ASSOCIAÇÃO PARA PROMOÇÃO DA EXCELÊNCIA DO SOFTWARE BRASILEIRO – SOFTEX., MPS.BR – Guia de Implementação – Parte 6:  
Fundamentação para Implementação do Nível B do MR-MPS:2011, junho 2011.  
Disponível em: <http://www.softex.br>.
- SOFTEX, 2011h, ASSOCIAÇÃO PARA PROMOÇÃO DA EXCELÊNCIA DO SOFTWARE BRASILEIRO – SOFTEX., MPS.BR – Guia de Implementação – Parte 7:  
Fundamentação para Implementação do Nível A do MR-MPS:2011, junho 2011.  
Disponível em: <http://www.softex.br>.
- SOFTEX, 2011i, ASSOCIAÇÃO PARA PROMOÇÃO DA EXCELÊNCIA DO SOFTWARE BRASILEIRO – SOFTEX., MPS.BR – Guia de Implementação – Parte 8:  
Implementação do MR-MPS:2011 em organizações que adquirem *software*, junho 2011. Disponível em: <http://www.softex.br>.
- SOFTEX, 2011j, ASSOCIAÇÃO PARA PROMOÇÃO DA EXCELÊNCIA DO SOFTWARE BRASILEIRO – SOFTEX., MPS.BR – Guia de Implementação – Parte 9:  
Implementação do MR-MPS:2011 em organizações do tipo Fábrica de *Software*, junho 2011. Disponível em: <http://www.softex.br>.
- SOFTEX, 2011k, ASSOCIAÇÃO PARA PROMOÇÃO DA EXCELÊNCIA DO SOFTWARE BRASILEIRO – SOFTEX., MPS.BR – Guia de Implementação – Parte 10:  
Implementação do MR-MPS:2011 em organizações do tipo Fábrica de Teste, junho 2011. Disponível em: <http://www.softex.br>.
- SOFTEX, 2011l, ASSOCIAÇÃO PARA PROMOÇÃO DA EXCELÊNCIA DO SOFTWARE BRASILEIRO – SOFTEX., MPS.BR – Curso de Introdução ao MPS-Software (C1-MPS-SW), setembro 2011.
- SOFTEX, 2012a, ASSOCIAÇÃO PARA PROMOÇÃO DA EXCELÊNCIA DO SOFTWARE BRASILEIRO – SOFTEX., MPS.BR – Guia Geral MPS de *Software*:2012, agosto 2012.  
Disponível em: <http://www.softex.br>.
- SOFTEX, 2012b, ASSOCIAÇÃO PARA PROMOÇÃO DA EXCELÊNCIA DO SOFTWARE BRASILEIRO – SOFTEX., MPS.BR – Guia Geral MPS de Serviços:2012, agosto 2012.  
Disponível em: <http://www.softex.br>.
- SOFTEX, 2012c, ASSOCIAÇÃO PARA PROMOÇÃO DA EXCELÊNCIA DO SOFTWARE BRASILEIRO – SOFTEX., MPS.BR – Guia de Avaliação:2012, maio 2012. Disponível em: <http://www.softex.br>.
- SOFTEX, 2012d, ASSOCIAÇÃO PARA PROMOÇÃO DA EXCELÊNCIA DO SOFTWARE BRASILEIRO – SOFTEX., MPS.BR – Guia de Implementação – Parte 11:  
Implementação e Avaliação do MR-MPS-SW:2012 em Conjunto com o CMMI-DEV v1.3, agosto 2012. Disponível em: <http://www.softex.br>.
- SOFTEX, 2012e, ASSOCIAÇÃO PARA PROMOÇÃO DA EXCELÊNCIA DO SOFTWARE BRASILEIRO – SOFTEX., MPS.BR – Guia de Implementação – Parte 12: Análise da Aderência do MRMPS-SW:2012 em relação à NBR ISO/IEC 29110-4-1:2012 - Engenharia de *Software* - Perfis de ciclo de vida para microorganizações (VSEs) - Parte 4-1: Especificações de perfil: Grupo Perfil Genérico, setembro 2012. Disponível em: <http://www.softex.br>.
- SOFTEX, 2012f, ASSOCIAÇÃO PARA PROMOÇÃO DA EXCELÊNCIA DO SOFTWARE BRASILEIRO – SOFTEX., MPS.BR – Guia de Implementação – Parte 13: Mapeamento

- e Sistemas de Equivalências entre o MR-MPS-SW:2012 e o MoProSoft:2005, outubro 2012. Disponível em: <http://www.softex.br>.
- SOLINGEN, R.; BERGHOUT, E., 1999, *The Goal/Question/Metric Method: a Practical Guide for Quality Improvement of Software Development*, McGraw-Hill.
- SPIEGEL, M. e STEPHENS, L., 2011, *Schaums Outline of Statistics*, Fourth Edition (Schaum's Outline Series) Mc Graw Hill.
- STAPLETON, J. e CONSTABLE, P., 1997, *DSDM: Dynamic Systems Development Method: The Method in Practice*, Addison-Wesley Professional.
- TALEB, N., 2012, *Antifragile: Things That Gain from Disorder*, Random House.
- TALEB, N., 2010, *The Black Swan: Second Edition: The Impact of the Highly Improbable: With a new section: "On Robustness and Fragility"*, Random House Trade Paperbacks.
- TENNANT, G., 2001, *Six Sigma: SPC and TQM in Manufacturing and Services*, Gower.
- TOIGO, J., 2002, *Disaster Recovery Planning: Preparing for the Unthinkable* (3rd Edition), Prentice Hall.
- UNKNOWN AUTHOR. Disponível em: <http://c2.com/cgi/wiki?CodeSmell>.
- WARD, P., e MELLOR, S., 1986, *Structured Development for Real-Time Systems*, Volume I: Introduction and Tools, Prentice-Hall.
- WARD, P., e MELLOR, S., 1986, *Structured Development for Real-Time Systems*, Volume II: Essential Modeling Techniques, Prentice-Hall.
- WARD, P., e MELLOR, S., 1986, *Structured Development for Real-Time Systems*, Volume III: Implementation Modeling Techniques, Prentice-Hall.
- WHEELER, D., 2000, *Understanding Variation. The Key to Managing Chaos*, SPC Press.
- WEINBERG, G., 1992, *Quality Software Management, vol. 1 Systems Thinking*. Dorset.
- WEINBERG, G., 1993, *Quality Software Management, vol. 2 First-Order Measurement*. Dorset.
- WEINBERG, G., 1994, *Quality Software Management, vol. 3 Congruent Action*. Dorset.
- WEINBERG, G., 1997, *Quality Software Management, vol. 4 Anticipating Change*. Dorset.
- WOOD, S. e SILVER, D., 1995, *Joint Application Development*, Wiley.
- YOURDON, E., 1989, *Structured Walkthroughs*, Prentice-Hall.
- YOURDON, E., e CONSTANTINE, L. 1979, *Structured Design: Fundamentals of a Discipline of Computer Program and System Design*, Prentice-Hall.
- ZAHNISER, R., 1995, *System Storyboarding Techniques*, American Programmer, Sep 1993. Disponível em: <http://www.belizenorth.com/articles/SST.htm>