

UNIVERSIDADE FEDERAL DE MINAS GERAIS  
INSTITUTO DE CIÊNCIAS EXATAS  
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

UM ESTUDO E UMA FERRAMENTA DE GERÊNCIA  
DE PROJETOS COM DESENVOLVIMENTO ÁGIL DE  
SOFTWARE

LEONARDO APARECIDO CISCON

Belo Horizonte / MG

2009



**UM ESTUDO E UMA FERRAMENTA DE GERÊNCIA  
DE PROJETOS COM DESENVOLVIMENTO ÁGIL DE  
SOFTWARE**



LEONARDO APARECIDO CISCON

UM ESTUDO E UMA FERRAMENTA DE GERÊNCIA  
DE PROJETOS COM DESENVOLVIMENTO ÁGIL DE  
SOFTWARE

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação do Instituto de Ciências Exatas da Universidade Federal de Minas Gerais como requisito parcial para a obtenção do grau de Mestre em Ciência da Computação.

ORIENTADOR: PH. D. RODOLFO SÉRGIO FERREIRA RESENDE

Belo Horizonte

2009

© 2009, Leonardo Aparecido Ciscon  
Todos os direitos reservados.

Ciscon, Leonardo Aparecido

C579e

Um estudo e uma ferramenta de gerência de projetos com desenvolvimento ágil de software [manuscrito] / Leonardo Aparecido Ciscon. – Belo Horizonte, 2009.

xv, 146 f., enc. : il: 29 cm

Orientador: Rodolfo Sérgio Ferreira Resende

Dissertação (Mestrado) – Universidade Federal de Minas Gerais.  
Departamento de Ciência da Computação

1. Computação – Teses. 2. Engenharia de Software – Teses. 3. Gerência de Projetos de Software – Teses. I. Resende, Rodolfo Sérgio Ferreira. II. Universidade Federal de Minas Gerais. Departamento de Ciência da Computação. III. Título.

CDU 519.6\*32



A meus pais (José Tarcísio e Ana Maria) e irmãos (Carlos Eduardo e Giancarlo)

Por todo carinho, educação e dedicação.

A meus tios, tias, padrinhos, madrinhas, primos e primas e avó pelo apoio.

À Michelle por continuar me fazendo uma pessoa muito mais feliz.

Aos meus amigos, colegas, camaradas e conhecidos, pela presença.



Agradeço a Deus,  
Meus pais (José Tarcísio e Ana Maria) e irmãos (Carlos Eduardo e Giancarlo),  
Michelle,  
André, Emiliane e Viviane,  
Gustavo, Humberto e Mariane,  
Aos demais amigos e colegas,  
Agradeço também ao meu orientador: Professor Rodolfo,  
Demais professores do DCC / UFMG,  
Aos funcionários do DCC / UFMG.  
Aos amigos do MSN, GTalk e Orkut.

# Resumo

As práticas denominadas ágeis vêm sendo cada vez mais utilizadas na gerência de projetos de software. Neste sentido, este trabalho estuda a abordagem ágil e a abordagem tradicional de gerência de projetos. Foram analisados aspectos das organizações, em particular das pessoas e das diferentes práticas e características dos projetos. São apresentadas e discutidas as participações do gerente de projeto em cada ambiente com ênfase na abordagem ágil de gerência de projetos. Como complemento do trabalho, uma ferramenta com base na WWW para suporte à gerência de projetos, a XPlanner, foi personalizada resultando na XPlannerPlus. A XPlannerPlus permitiu a observação de ambientes de desenvolvimento com práticas ágeis e tradicionais. Outro complemento do trabalho foi a realização de um estudo de caso preliminar em uma organização de desenvolvimento de software de pequeno porte. Foi utilizado neste estudo de caso a XPlannerPlus e observamos que ela atende de forma satisfatória a várias demandas de desenvolvimento.

**Palavras-chave:** Gerência de Projetos de Software, Gerência Ágil de Projetos de Software, Ferramentas de Gerência de Projetos, Qualidade de Software, Engenharia de Software

# Abstract

The agile practices are being increasingly used in the management of software projects. This work studies the agile and traditional practices used in project management. In this study it was analyzed several organization aspects, in particular the characteristics of people and project practices. It is presented and discussed the participation of the project manager with emphasis in agile practices. As a complement of the work it was developed a tool called XPlannerPlus. XPlannerPlus was developed based on an existing web-based tool known as XPlanner. Another complement of this work is a case study conducted in a small software development organization. XPlannerPlus was used and the case study helped us to verify some of the aspects of our study as well as to test the tool itself.

**Keywords:** Software Project Management, Agile Software Project Management, Project Management Tools, Software Quality, Software Engineering

# Lista de Ilustrações

Figura 2.1 - Seqüência típica de fases no ciclo de vida de um projeto. Fonte: PMBOK14	
Figura 2.2 - A relação entre as partes interessadas e o projeto. Fonte: PMBOK	15
Figura 2.3 - Fluxo de Trabalho da Disciplina de Gerência de Projetos no RUP. Fonte: (RUP, 2003)	26
Figura 3.1 - Exemplo de Gráfico de Progresso (Gráfico de Burndown)	51
Figura 4.1 - Tarefas por Usuário	74
Figura 4.2 - Barras de Progresso da Ferramenta XPlanner	76
Figura 4.3 - Projeto de Personalização da XPlanner	78
Figura 4.4 - Página de Cadastro do Projeto	80
Figura 4.5 - Página de Cadastro de Recurso / Usuário	82
Figura 4.6 - Página de Cadastro de Riscos	84
Figura 4.7 - Página de Cadastro de Obstáculos (Impedimentos)	85
Figura 4.8 - Página de Cadastro de Informação da Avaliação Final de Projeto	87
Figura 4.9 - Relatório de Orçamento	88
Figura 4.10 - Definição do Sucesso do Projeto	90
Figura 5.1 - Projeto da XPlannerPlus: requisitos da 1a Iteração	96
Figura 5.2 - Projeto da XPlannerPlus: requisitos da 2a Iteração	97
Figura 5.3 - Projeto da XPlannerPlus: tarefas do requisito Cadastro de Riscos	98
Figura 5.4 - Projeto da XPlannerPlus: identificação de um risco do projeto	99
Figura 5.5 - Arquitetura da Ferramenta XPlanner. Fonte: (XPlanner, 2008)	102
Figura 6.1 - Projeto A: acompanhamento da primeira semana de desenvolvimento	115
Figura 6.2 - Projeto A: tarefas do requisito 17	116
Figura 6.3 - Projeto A: risco identificado	117
Figura 6.4 - Projeto A: listagem de obstáculos identificados	117
Figura 6.5 - Projeto A: informação sobre a avaliação final do projeto	118
Figura 6.6 - Projeto B: acompanhamento da primeira semana de desenvolvimento	121
Figura 6.7 - Projeto B: gráfico de progresso da iteração	122
Figura 6.8 - Projeto B: gráfico do trabalho restante	122
Figura 6.9 - Projeto B: obstáculo identificado	123
Figura 6.10 - Projeto C: gráfico do trabalho restante	125
Figura 6.11 - Projeto C: acompanhamento da segunda semana de desenvolvimento	126
Figura 6.12 - Projeto C: definição do sucesso do projeto	127

# Lista de Tabelas

Tabela 2.1 - Sucessos, sucessos parciais e falhas em Projetos de Software. Fonte: Johnson, 2001	8
Tabela 2.2 - Fatores de Sucesso em Projetos. Fonte: “The Chaos Report”	8
Tabela 2.3 - Fatores de Fracasso em Projetos. Fonte: “The Chaos Report”	9
Tabela 2.4 - Áreas de Conhecimento do PMBOK. Fonte: PMBOK	22
Tabela 2.5 - Metodologias de Gerência de Projetos de Software	28
Tabela 3.1 - Comparação entre abordagem ágil e tradicional de gerência de projetos	65
Tabela 5.1 - Funcionalidades personalizadas na Ferramenta XPlanner	103
Tabela 6.1 - Síntese das respostas do Questionário	128



# Sumário

<u>AGRADECIMENTOS.....</u>	<u>V</u>
<u>RESUMO.....</u>	<u>VI</u>
<u>ABSTRACT.....</u>	<u>VII</u>
<u>LISTA DE ILUSTRAÇÕES.....</u>	<u>VIII</u>
<u>LISTA DE TABELAS.....</u>	<u>IX</u>
<u>1. INTRODUÇÃO.....</u>	<u>1</u>
1.1. CONTEXTUALIZAÇÃO E MOTIVAÇÃO.....	1
1.2. OBJETIVOS E METODOLOGIA.....	4
1.3. ESTRUTURA DO TRABALHO.....	5
<u>2. REVISÃO BIBLIOGRÁFICA.....</u>	<u>7</u>
2.1. CONTEXTUALIZAÇÃO.....	7
2.2. GERÊNCIA DE PROJETOS DE SOFTWARE.....	10
2.2.1. CONCEITOS.....	10
2.3. CICLO DE VIDA E ORGANIZAÇÃO DE PROJETOS.....	13
2.3.1. CICLO DE VIDA DO PROJETO.....	13
2.3.2. PARTES INTERESSADAS NOS PROJETOS.....	14
2.3.3. DIFICULDADES DA GERÊNCIA DE PROJETOS DE SOFTWARE.....	16
2.4. METODOLOGIAS DE GERÊNCIA DE PROJETOS.....	18
2.5. GERÊNCIA DE PROJETOS DE SOFTWARE COM UTILIZAÇÃO DE PRÁTICAS TRADICIONAIS.....	20
2.5.1. PMBOK.....	21
2.5.2. CMMI.....	23
2.5.3. RUP.....	24
2.5.4. ISO / IEC 15504.....	27

<b>2.6.</b>	<b>GERÊNCIA DE PROJETOS DE SOFTWARE COM UTILIZAÇÃO DE PRÁTICAS ÁGEIS</b>	<b>29</b>
2.6.1.	METODOLOGIA ÁGIL DE DESENVOLVIMENTO .....	31
2.6.2.	EXTREME PROGRAMMING (XP) .....	33
2.6.3.	SCRUM.....	34
2.6.4.	DSDM.....	35
2.6.5.	CRYSTAL .....	36
2.6.6.	ASD .....	36
2.6.7.	FDD.....	37
<b>2.7.</b>	<b>FERRAMENTAS DE APOIO À GERÊNCIA DE PROJETOS .....</b>	<b>37</b>
<b>2.8.</b>	<b>CONSIDERAÇÕES FINAIS .....</b>	<b>40</b>

### **3. UM ESTUDO SOBRE A UTILIZAÇÃO DE PRÁTICAS ÁGEIS E PRÁTICAS TRADICIONAIS NA GERÊNCIA DE PROJETOS DE SOFTWARE .....**

<b>3.1.</b>	<b>CONTEXTUALIZAÇÃO .....</b>	<b>41</b>
<b>3.2.</b>	<b>ORGANIZAÇÕES .....</b>	<b>44</b>
3.2.1.	FATORES DE NEGÓCIOS .....	44
3.2.2.	PROCESSOS.....	45
3.2.3.	TECNOLOGIA.....	46
3.2.4.	TAMANHO E DISTRIBUIÇÃO.....	47
<b>3.3.</b>	<b>PROJETOS.....</b>	<b>47</b>
3.3.1.	OBJETIVO PRIMORDIAL .....	48
3.3.2.	PROCESSO DE DESENVOLVIMENTO .....	48
3.3.3.	PLANEJAMENTO E CONTROLE .....	50
3.3.4.	DOCUMENTAÇÃO.....	52
3.3.5.	REQUISITOS .....	53
3.3.6.	RISCOS .....	54
3.3.7.	CARACTERÍSTICAS DO PROJETO .....	55
<b>3.4.</b>	<b>PESSOAS .....</b>	<b>56</b>
3.4.1.	DESENVOLVEDORES.....	57
3.4.2.	TESTADORES .....	58
3.4.3.	CLIENTES .....	59
3.4.4.	ALTA DIREÇÃO.....	59
3.4.5.	EQUIPE.....	60
<b>3.5.</b>	<b>PAPÉIS DO GERENTE DE PROJETO.....</b>	<b>61</b>
<b>3.6.</b>	<b>CONSIDERAÇÕES FINAIS .....</b>	<b>64</b>

### **4. PERSONALIZAÇÃO DE UMA FERRAMENTA PARA GERÊNCIA DE PROJETOS DE SOFTWARE.....**

<b>4.1.</b>	<b>CONTEXTUALIZAÇÃO .....</b>	<b>69</b>
<b>4.2.</b>	<b>ASPECTOS CONCEITUAIS .....</b>	<b>72</b>



4.2.1.	CARACTERÍSTICAS E FUNCIONALIDADES DA FERRAMENTA XPLANNER .....	72
4.2.2.	PLANEJAMENTO, ACOMPANHAMENTO E CONTROLE DO PROJETO NA FERRAMENTA XPLANNER .....	75
4.3.	FUNCIONALIDADES PERSONALIZADAS E DESENVOLVIDAS .....	77
4.3.1.	REQUISITO 1: CADASTRAR DADOS DO PROJETO .....	78
4.3.2.	REQUISITO 2: CADASTRAR RECURSOS / USUÁRIOS .....	80
4.3.3.	REQUISITO 3: CADASTRAR, ALTERAR, REMOVER E LISTAR RISCOS DO PROJETO .....	82
4.3.4.	REQUISITO 4: CADASTRAR, ALTERAR, REMOVER E LISTAR OBSTÁCULOS (IMPEDIMENTOS) DO PROJETO .....	84
4.3.5.	REQUISITO 5: CADASTRAR INFORMAÇÃO SOBRE AVALIAÇÃO FINAL DO PROJETO ( <i>POST-MORTEM</i> ).....	86
4.3.6.	REQUISITO 6: GERAR ORÇAMENTO .....	87
4.3.7.	REQUISITO 7: DEFINIR SUCESSO DO PROJETO.....	88
4.3.8.	REQUISITO 8: AJUDA SOBRE GP ÁGIL E TRADICIONAL.....	90
4.4.	CONSIDERAÇÕES FINAIS.....	91
<b>5. ASPECTOS OPERACIONAIS E DE IMPLEMENTAÇÃO.....</b>		<b>93</b>
5.1.	ASPECTOS OPERACIONAIS .....	93
5.2.	ASPECTOS DE IMPLEMENTAÇÃO.....	100
5.2.1.	ARQUITETURA DO SISTEMA .....	101
5.2.2.	IMPLEMENTAÇÃO.....	103
5.3.	CONSIDERAÇÕES FINAIS.....	106
<b>6. ESTUDO DE CASO.....</b>		<b>107</b>
6.1.	CONTEXTUALIZAÇÃO.....	107
6.2.	METODOLOGIA.....	109
6.3.	AMBIENTE .....	112
6.4.	GERÊNCIA DO PROJETO A.....	114
6.4.1.	CARACTERIZAÇÃO.....	114
6.4.2.	USO DA FERRAMENTA .....	114
6.4.3.	RESULTADOS .....	118
6.5.	GERÊNCIA DO PROJETO B.....	119
6.5.1.	CARACTERIZAÇÃO.....	119
6.5.2.	USO DA FERRAMENTA .....	120
6.5.3.	RESULTADOS .....	123
6.6.	GERÊNCIA DO PROJETO C.....	124
6.6.1.	CARACTERIZAÇÃO.....	124
6.6.2.	USO DA FERRAMENTA .....	124
6.6.3.	RESULTADOS .....	127
6.7.	RESULTADOS DOS QUESTIONÁRIOS .....	128

6.8.	ASPECTOS OBSERVADOS.....	130
6.9.	CONSIDERAÇÕES FINAIS .....	132
<b><u>7. CONCLUSÃO .....</u></b>		<b>135</b>
7.1.	CONSIDERAÇÕES FINAIS .....	135
7.2.	TRABALHOS FUTUROS .....	137
<b><u>REFERÊNCIAS BIBLIOGRÁFICAS.....</u></b>		<b>139</b>

# Capítulo 1

## Introdução

### 1.1. Contextualização e Motivação

A história dos fracassos no desenvolvimento de projetos de software é amplamente conhecida e divulgada. Relatórios, como os realizados nos anos 90 pelo DoD (Departamento de Defesa Americano) (DOD, 1994) e o “*The Chaos Report*” (The Standish Group, 1994), apontavam de maneira geral que somente pequena parcela dos projetos de software era considerada bem sucedida, o que significa que eles foram concluídos no tempo estipulado, sob o orçamento previsto e de acordo com os requisitos dos usuários.

Ambos os estudos apontavam que o principal problema não era de caráter técnico, mas sim gerencial.

Uma das tentativas iniciais de resolver tais falhas foi incentivada e financiada pelo DoD. O *Software Engineering Institute* (SEI) da Universidade *Carnegie Mellon*, desenvolveu um modelo de maturidade de desenvolvimento de software, o CMM (*Capability Maturity Model*). O objetivo principal era estabelecer um padrão de qualidade para software desenvolvido para as forças armadas. Atualmente, o modelo de referência é o *Capability Maturity Model Integration* – CMMI v1.2 (SEI, 2002), lançado em 2002 como evolução do CMM.

Além dos modelos CMM / CMMI, outra publicação que influenciou a abordagem considerada “tradicional” de desenvolvimento de software é o Corpo de

Conhecimentos em Gerência de Projetos – PMBOK (PMI, 2004). O PMBOK descreve os conhecimentos necessários para um gerente de projetos que referimos neste trabalho como sendo “tradicional”. Contudo, segundo Nerur e outros (Nerur et. al., 2005) as organizações precisam adaptar suas estruturas, estratégias e políticas para satisfazerem os novos ambientes e a crescente demanda da sociedade contemporânea por sistemas de informação.

As organizações devem buscar alternativas sobre como gerenciar seus projetos de software, visando à diminuição dos fracassos e a melhoria na qualidade de seus produtos e serviços. Segundo Augustine e outros (Augustine et. al., 2005), os processos tradicionais de desenvolvimento e gerência de projetos (GP) têm sido caracterizados como uma descrição linear de um processo seqüencial. Estes métodos podem ser efetivos para requisitos estáveis, conhecidos e consistentes. Porém, segundo Nerur e outros (Nerur et. al., 2005), a metodologia tradicional sente falta de flexibilidade para dinamicamente ajustar-se ao processo de desenvolvimento.

Lehman (Lehman, 2001) diz que uma metodologia tem que atender a um ambiente complexo, imprevisível e de freqüente mudança nos requisitos. Já Boehm e Turner (Boehm & Turner, 2004), afirmam que as organizações precisam desenvolver-se para o melhor equilíbrio entre agilidade e processos tradicionais que melhor se adaptam a sua situação.

Neste cenário, diversos autores ((Boehm & Turner, 2004), (Augustine et. al., 2005), (Morien, 2005), (Chin, 2004) e (Anderson & Schragenheim, 2003)) evidenciam o crescimento de popularidade de novos métodos de desenvolvimento conhecidos como Métodos Ágeis, que prometem flexibilidade na entrega rápida de um produto com qualidade. Apesar de muitas discussões sobre as vantagens e desvantagens da utilização dos métodos ágeis no desenvolvimento de software (Agile Alliance, 2008), algumas questões não foram respondidas de forma definitiva: quais são as práticas mais fundamentais de um projeto? Quanto um projeto deve ter de práticas ágeis? E por fim, qual seria o papel do gerente de projetos neste novo modelo de desenvolvimento?

O PMBOK define gerência de projetos como a aplicação de conhecimentos, habilidades, ferramentas e técnicas às atividades do projeto a fim de atender aos seus requisitos. Dentre algumas das atividades de um gerente de projeto estão:

- Identificar as necessidades do projeto;
- Estabelecer objetivos claros e alcançáveis;
- Equilibrar as demandas conflitantes de qualidade, escopo, tempo e custo;
- Adaptar as especificações, planos e a abordagem às diferentes preocupações e expectativas das diversas partes interessadas; e;
- Minimizar os impactos negativos das incertezas dos projetos.

Conforme anteriormente informado, em se tratando de processos tradicionais, o PMBOK é considerado um guia completo. Já em relação aos métodos ágeis, a literatura em torno do papel do gerente de projetos ainda não foi consolidada. Contudo, diversos autores têm contribuído para este assunto.

Nerur e outros (Nerur et. al., 2005) apontam que a principal mudança está no comportamento do gerente perante o projeto e sua equipe: métodos ágeis exigem uma mudança de gerência. Esta mudança envolve uma transição vinda do que ele identifica como “comando-e-controle” para o que ele denomina um comportamento de “liderança-e-colaboração”. Anderson e outros (Anderson et. al., 2003) questionam a importância do gerente de projetos em um ambiente ágil. Poppendieck (Anderson et. al., 2003) afirma que as equipes de desenvolvimento ágil não precisam de um gerente, mas sim de um líder. Coram e Bohner (Coram & Bohner, 2005) salientam que os gerentes devem estar dispostos a partilhar a autoridade na tomada de decisões.

Além da questão de conhecimentos em gerência de projetos, a utilização de ferramentas apropriadas facilita o desenvolvimento das demandas de software da sociedade contemporânea. Neste sentido, estudamos e personalizamos a XPlanner, ferramenta de suporte à gerência de projetos em ambiente *Web*. A XPlanner atende a

uma boa gama de requisitos de gerência de projetos ágil e gerência de projetos tradicional.

## **1.2. Objetivos e Metodologia**

Esta dissertação descreve o nosso trabalho relacionado (i) ao estudo da gerência de projetos tradicional contrastada com a gerência de projetos baseados em metodologia ágil, investigando os pontos comuns, os pontos divergentes e os processos da gerência tradicional que se adaptariam aos princípios da metodologia ágil e (ii) personalização da XPlanner, resultando na XPlannerPlus. A personalização da XPlannerPlus e o seu uso em um estudo de caso serviram para dar subsídio a diversos aspectos do nosso estudo.

Em nosso estudo foi dada ênfase à gerência ágil de projetos, personalização de metodologias e qualidade de software. Também estudamos conceitos de planejamento e controle de projetos assim como os princípios e técnicas da metodologia ágil. Não foi objeto do trabalho a compilação das práticas mais utilizadas na gerência tradicional.

Identificamos os requisitos para uma ferramenta de gerência de projetos que atende aos interessados nas melhores práticas dos métodos ágeis e tradicionais. Observando, dentre outros trabalhos, o trabalho de Cheema e Shahid (Cheema & Shahid, 2005) que citam uma lista de fatores que podem influenciar na personalização da metodologia de gerência de projetos, sendo ela GP tradicional, GP Ágil ou uma junção de ambas.

Um estudo de caso preliminar que é descrito no capítulo 6 indica que a XPlannerPlus atende de forma satisfatória a várias demandas de desenvolvimento.

Acreditamos que nossa iniciativa irá contribuir na adoção de práticas de gerência de projetos por organizações que não gerenciam ou gerenciam informalmente seus

projetos, o que levaria a um aumento na qualidade dos produtos em consequência da melhoria na qualidade do processo empregado.

### **1.3. Estrutura do Trabalho**

Esta dissertação está dividida, além da introdução, em 7 capítulos.

O segundo capítulo discute alguns dos temas principais deste trabalho: gerência de projetos, abordagens tradicionais de gerência de projetos, métodos ágeis de desenvolvimento e ferramentas de apoio à gerência de projetos. O terceiro capítulo apresenta um estudo sobre a abordagem ágil e a abordagem tradicional de gerência de projetos de software com foco nas características das organizações, projetos, pessoas e os papéis do gerente de projeto.

O capítulo 4 apresenta aspectos conceituais e as funcionalidades da XPlannerPlus. São apresentadas justificativas e discussões para as funcionalidades propostas neste trabalho. O quinto capítulo discute aspectos operacionais e de implementação da personalização da ferramenta.

O capítulo 6 apresenta o estudo de caso realizado em uma organização, como forma de avaliação da personalização da ferramenta XPlanner. A XPlannerPlus tem uma contribuição por si mesma em função de ser uma extensão da XPlanner que é uma ferramenta razoavelmente difundida. Além disso, a XPlannerPlus contribuiu para uma validação das idéias que observamos no estudo.

O sétimo capítulo conclui a dissertação apresentando considerações finais ressaltando as contribuições e limitações. São discutidos ainda alguns trabalhos futuros. Após o sétimo e último capítulo são listadas as referências.





# Capítulo 2

## Revisão Bibliográfica

Este capítulo apresenta um estudo sobre os temas principais deste trabalho: gerência de projetos e suas características, metodologias de gerência de projetos, abordagem tradicional de gerência de projetos (PMBOK, CMMI, RUP, ISO/IEC 15504, dentre outras), metodologia ágil de desenvolvimento (XP, Scrum, DSDM, Crystal, dentre outros) além de uma caracterização sobre ferramentas de apoio à gerência de projetos.

Para evitar problemas de interpretação dos diversos trabalhos não reescrevemos os termos usados pelos autores, como exemplo, mantivemos o uso dos termos “metodologia ágil” ou “métodos ágeis”. Utilizamos o adjetivo “ágil”, conforme alguns autores, para denotar o uso de práticas denominadas ágeis, por exemplo, “grupo ágil” significa um grupo de desenvolvedores que utilizam práticas ágeis.

### **2.1. Contextualização**

Desde os anos 90, estudos e pesquisas demonstram que a gerência de projetos é a causa mais evidente de falhas na execução e entrega de produtos e serviços de software. O SEI – *Software Engineering Institute*, constatou, já nos anos 80, que o principal problema das organizações não é técnico, e sim gerencial.

O conjunto destas falhas ainda é realidade constante na maioria das organizações. As diversas versões do “*The Chaos Report*” (The Standish Group, 1994) (The Standish

Group, 2006) apontam uma evolução constante no número de sucesso nos projetos, porém, a porcentagem de falhas em projetos continua muito alta. Este relatório, que focou a produção de softwares comerciais, indicou que 31% dos projetos estudados foram cancelados e apenas 16% foram concluídos com sucesso, no ano de 1994. Já em 2006, o número de projetos cancelados diminuiu para 19% e o número de projetos bem-sucedidos saltou para 35%.

A Tabela 2.1 apresenta as porcentagens de sucessos, sucessos parciais e fracassos nos projetos de software no período de 1994 até 2006, segundo o “*The Standish Group*”:

**Tabela 2.1 - Sucessos, sucessos parciais e falhas em Projetos de Software. Fonte: Johnson, 2001**

	1994	1998	2000	2006
Sucesso – projetos concluídos e operacionais, dentro do escopo, prazo e orçamento.	16%	27%	28%	35%
Sucessos parciais – projetos concluídos, mas fora do escopo ou prazo ou orçamento.	53%	33%	49%	46%
Cancelados	31%	40%	23%	19%

Este relatório aponta a gerência de projetos como sendo a razão para o sucesso ou a falha de um projeto de software. Segundo os dados do ano de 2006 da Tabela 2.1, é possível perceber que 65% dos projetos não são concluídos conforme planejado ou são cancelados.

Ainda de acordo com pesquisas do “*The Standish Group*”, as duas tabelas a seguir apontam os fatores de sucesso e de fracasso dos projetos, respectivamente.

**Tabela 2.2 - Fatores de Sucesso em Projetos. Fonte: “The Chaos Report”**

Fatores de Sucesso em Projetos	%
--------------------------------	---

1. Envolvimento do Usuário	15,9%
2. Suporte da alta administração	13,9%
3. Clara definição dos requisitos	13,0%
4. Planejamento Adequado	9,6%
5. Expectativas Realísticas	8,2%
6. Pontos de verificação dos projetos menores	7,7%
7. Competência da Equipe	7,2%
8. Propriedade	5,3%
9. Clara visão e objetivos	2,9%
10. Trabalho intenso	2,4%
11. Outros	13,9%

**Tabela 2.3 - Fatores de Fracasso em Projetos. Fonte: “The Chaos Report”**

<b>Fatores de Fracasso em Projetos</b>	<b>%</b>
1. Requisitos e especificações incompletos	13,1%
2. Falta de participação dos usuários	12,4%
3. Falta de recursos	10,6%
4. Expectativas irreais	9,9%
5. Ausência de suporte da alta administração	9,3%
6. Volatilidade de requisitos e especificações	8,7%
7. Falta de planejamento	8,1%
8. Obsolescência do Projeto	7,5%
9. Ausência de gerência de Tecnologia da Informação	6,2%
10. Problemas com a tecnologia empregada	4,3%
11. Outros	9,9%

A Tabela 2.1 e a Tabela 2.2 são originárias de dados coletados em entrevistas com gerentes executivos de TI (Tecnologia da Informação) de organizações dos Estados Unidos e Canadá. Pode-se notar que dentre os diversos fatores, os 4 primeiros da Tabela 2.1 (envolvimento do usuário, suporte da alta administração, clara definição dos requisitos e planejamento adequado) têm uma importância vital para o projeto.

Todos os dados apresentados demonstram que a indústria de software ainda tem muito que evoluir neste aspecto.

## 2.2. Gerência de Projetos de Software

### 2.2.1. Conceitos

Segundo Kerzner (Kerzner, 2001) um **projeto** pode ser considerado como qualquer série de atividades e tarefas que:

- Têm um objetivo específico a ser concluído dentro de determinadas especificações;
- Têm início e término definido;
- Têm limites de financiamento (se aplicável);
- Consomem recursos humanos e não-humanos (ou seja, o dinheiro, as pessoas, equipamentos);
- Sejam multifuncionais (isto é, abarcam várias linhas funcionais).

Já segundo o PMBOK, um **projeto** é um esforço temporário empreendido para criar um produto, serviço ou resultado exclusivo. Por temporário entende-se que todos os projetos possuem um início e um final definidos. O final é alcançado quando os objetivos do projeto tiverem sido atingidos, quando se tornar claro que os objetivos do projeto não serão ou não poderão ser atingidos ou quando não existir mais a necessidade do projeto e ele for encerrado. Já por “produto, serviço ou resultado exclusivo” entende-se que um projeto visa entregar algo exclusivo, singular, mesmo possuindo elementos repetitivos no processo de construção.

Outra característica de um projeto é a sua elaboração progressiva, isto é, desenvolvimento em etapas e em contínuos incrementos.

O PMBOK define **gerência de projetos** como a aplicação de conhecimentos, habilidades, ferramentas e técnicas às atividades do projeto a fim de atender aos seus requisitos. Dentre algumas das atividades de um gerente de projeto estão:

- Identificar as necessidades do projeto;
- Estabelecer objetivos claros e alcançáveis;
- Equilibrar as demandas conflitantes de qualidade, escopo, tempo e custo;
- Adaptar as especificações, planos e a abordagem às diferentes preocupações e expectativas das diversas partes interessadas;
- Minimizar os impactos negativos das incertezas dos projetos.

Segundo a norma ISO / IEC 15504 (ISO / IEC 15504, 2006), o propósito da **gerência de projetos** é identificar, estabelecer, coordenar e monitorar as atividades, tarefas e recursos de que um projeto necessita para produzir um produto ou serviço, no contexto dos requisitos e restrições do projeto. Como atividades e resultados deste processo, a norma indica:

- Definição do escopo do trabalho;
- Avaliação da viabilidade de atingir as metas do projeto, considerando as restrições e os recursos disponíveis;
- Dimensionamento e estimativa das tarefas e recursos necessários;
- Identificação e monitoramento das interfaces entre os elementos do projeto e com outros projetos e unidades organizacionais;
- Desenvolvimento e implementação dos planos para a execução do projeto;
- Monitoramento e divulgação do progresso do projeto;
- Execuções de ações para corrigir desvios do plano e prevenção de ocorrência de problemas identificados no projeto;
- Identificação, análise, tratamento e monitoramento contínuo dos riscos do projeto.

Por outro lado, Kerzner (Kerzner, 2001) afirma que **gerência de projetos** envolve planejamento e acompanhamento de projeto e inclui itens como:

- Planejamento de Projeto
  - Definição de requisitos de trabalho;
  - Definição de quantidade e qualidade do trabalho;
  - Definição dos recursos necessários.
- Acompanhamento de Projeto
  - Acompanhamento de progresso;
  - Comparação entre resultados reais e resultados previstos;
  - Análise de impacto;
  - Realização de ajustes.

Ainda segundo Kerzner (Kerzner, 2001), uma **gerência de projetos** bem sucedida pode então ser definida como tendo atingido os objetivos do projeto:

- Dentro do tempo;
- Dentro do custo;
- No nível de desempenho e tecnologia desejados;
- Utilizando os recursos atribuídos de forma eficaz e eficiente;
- Aceito pelo cliente.

De fato, a definição do que seria o sucesso do projeto pode muitas vezes criar oposição entre o aceite do cliente e o custo, tempo, escopo e qualidade do projeto. Shenhar e Levy (Shenhar & Levy, 1997), por exemplo, dizem que a definição de sucesso de um projeto passa por quatro grandes pilares: eficiência do projeto (escopo, tempo, custo, qualidade), impacto no cliente (satisfação, resultado), sucesso nos negócios (retorno de investimento, taxa de retorno, etc.) e preparação para o futuro (suporte a infra-estrutura, evolução, etc.).

Todavia, o PMBOK afirma que o sucesso exige a participação de todos os membros da equipe, mas é sempre responsabilidade da gerência fornecer os recursos necessários para que exista sucesso.

## **2.3. Ciclo de Vida e Organização de Projetos**

Segundo o PMBOK, os projetos e a gerência de projetos são executados em um ambiente mais amplo que o do projeto propriamente dito. A equipe de gerência de projetos precisa entender esse contexto mais amplo para que possa selecionar as fases do ciclo de vida, os processos, as ferramentas e técnicas adequadas ao projeto.

Kerzner (Kerzner, 2001) afirma que uma compreensão destas fases permite aos gerentes e executivos um melhor controle total dos recursos empresariais utilizados na realização dos objetivos pretendidos.

### **2.3.1. Ciclo de vida do Projeto**

A organização ou os gerentes de projetos podem dividir projetos em fases para oferecer melhor controle gerencial com ligações adequadas com as operações em andamento da organização executora. Coletivamente, essas fases são conhecidas como o ciclo de vida do projeto (PMI, 2004).

O ciclo de vida do projeto define as fases que conectam o início de um projeto ao seu final.

Não existe uma única melhor maneira para definir um ciclo de vida ideal do projeto. Algumas organizações estabelecem políticas que padronizam todos os projetos com um único ciclo de vida, enquanto outras permitem que a equipe de gerência de projetos escolha o ciclo de vida mais adequado para seu próprio projeto.

Segundo o PMBOK, ciclos de vida do projeto geralmente definem:

- Qual trabalho técnico deve ser realizado em cada fase (por exemplo, em qual fase deve ser realizado o trabalho do analista?);
- Quando as entregas devem ser geradas em cada fase e como cada entrega é revisada, verificada e validada;
- Quem está envolvido em cada fase (por exemplo, quando os implementadores devem estar envolvidos com os requisitos e o projeto?);
- Como controlar e aprovar cada fase.

A Figura 2.1 ilustra uma seqüência típica de fases de um projeto, bem como exemplifica possíveis entradas e saídas de cada fase.

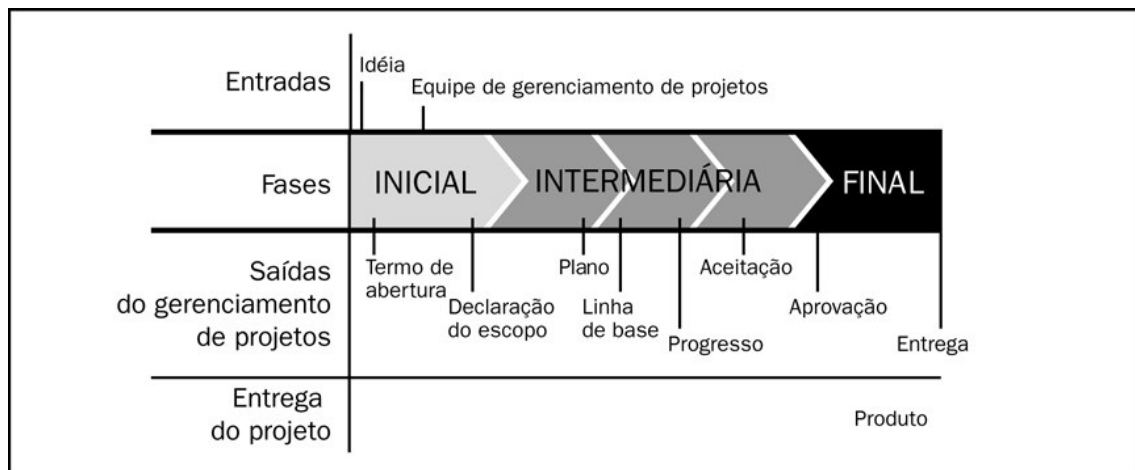


Figura 2.1 - Seqüência típica de fases no ciclo de vida de um projeto. Fonte: PMBOK

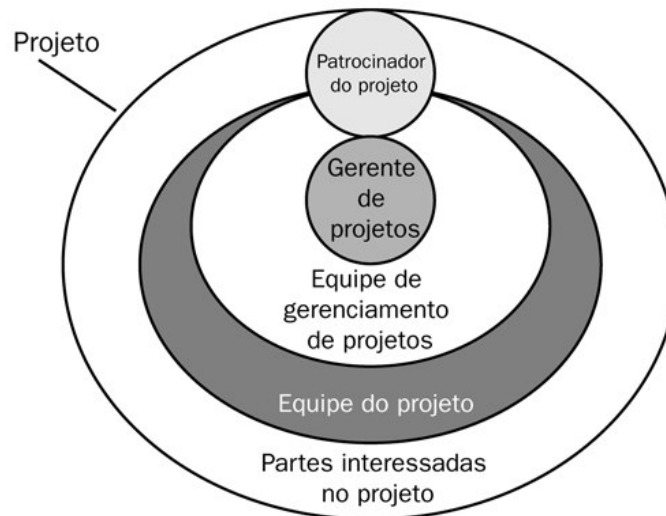
### 2.3.2. Partes Interessadas nos Projetos

Segundo o PMBOK, partes interessadas no projeto (também denominadas *stakeholders*) são pessoas e organizações ativamente envolvidas no projeto ou cujos interesses podem ser afetados como resultado da execução ou do término do projeto. Eles podem também exercer influência sobre os objetivos e resultados do projeto. A equipe de gerência de projetos precisa identificar as partes interessadas, determinar suas



necessidades e expectativas e, na medida do possível, gerenciar sua influência em relação aos requisitos para garantir um projeto bem-sucedido.

A Figura 2.2 ilustra a relação entre as partes interessadas e o projeto.



**Figura 2.2 - A relação entre as partes interessadas e o projeto. Fonte: PMBOK**

As partes interessadas possuem diversos níveis de responsabilidade e de autoridade quando participam de um projeto e estes podem mudar durante o seu ciclo de vida. Às vezes, a identificação das partes interessadas pode ser difícil. Se uma importante parte interessada não for identificada, poderá haver grandes problemas para um projeto.

As principais partes interessadas em projetos incluem, mas não se limitam a (PMI, 2004):

**Gerente de projetos:** A pessoa responsável pela gerência do projeto.

**Cliente/usuário:** A pessoa ou organização que utilizará o produto do projeto. Em algumas áreas de aplicação, os termos cliente e usuário são sinônimos, enquanto em outras, cliente se refere à entidade que adquire o produto do projeto e usuários são os que utilizarão diretamente o produto do projeto. No contexto deste trabalho, adotamos que cliente é sinônimo de usuário.

**Organização executora:** A empresa cujos funcionários estão mais diretamente envolvidos na execução do trabalho do projeto.

**Membros da equipe do projeto:** O grupo que está executando o trabalho do projeto.

**Equipe de gerência de projetos:** Os membros da equipe do projeto que estão diretamente envolvidos nas atividades de gerência de projetos.

**Patrocinador:** A pessoa ou o grupo que fornece os recursos financeiros, em dinheiro ou em recursos materiais, para o projeto.

**Influenciadores:** Pessoas ou grupos que não estão diretamente relacionados à aquisição ou ao uso do produto do projeto, mas que, devido à posição de uma pessoa na organização do cliente ou na organização executora, podem influenciar, positiva ou negativamente, no andamento do projeto.

Ainda segundo o PMBOK, os gerentes de projetos precisam gerenciar as expectativas das partes interessadas, o que pode ser difícil, pois elas em geral têm objetivos diferentes ou conflitantes.

### **2.3.3. Dificuldades da Gerência de Projetos de Software**

Muitos projetos de software são realizados sem um planejamento de como a idéia identificada pelo levantamento de requisitos e necessidades dos clientes pode ser transformada em produto. O planejamento, quando ocorre, é feito de forma superficial.

Segundo Rouiller (Rouiller, 2001), os gerentes de projetos não têm o costume de estimar. Quando estimam, costumam basear-se em estimativas passadas, mesmo sabendo que elas podem estar incorretas (não sabem também precisar o quanto elas estão incorretas). Há gerentes que se recusam a estimar somente por julgarem perda de tempo, uma vez que correm o risco de obter resultados incorretos e, portanto, estarem desperdiçando tempo.

Estimar, medir e controlar um projeto de software são tarefas difíceis, pois o desenvolvimento de software é uma atividade criativa e intelectual, com muitas variáveis envolvidas (como metodologias, modelos de ciclo de vida, técnicas, ferramentas, tecnologias, recursos e atividades diversas). Os gerentes inexperientes tentam cumprir prazos dando a máxima velocidade na fase inicial e estão despreparados para os momentos de impasse, quando as mudanças são inevitáveis (Rouiller, 2001).

A dinamicidade do processo de software dificulta também a gerência efetiva de projetos de software, devido, dentre outras, às alterações constantes nos planos de projetos, redistribuição de atividades, inclusão e exclusão de atividades, adaptação de cronogramas, realocação de recursos, novos acordos com os clientes, entregas intermediárias não previstas. Um projeto de software também deve adaptar-se ao ambiente, dependendo da disponibilidade de recursos, ferramentas e habilidades do pessoal ou equipe.

De acordo com Rouiller (Rouiller, 2001) outros fatores ainda agravam os problemas de gerência de projetos de software em empresas de pequeno e médio porte, tais como: inexistência de um processo definido, recursos pessoais e financeiros limitados, ausência ou pouca cultura em processos, pouco treinamento em engenharia de software, imaturidade metodológica, crescimento ocorrido por demanda, falta de experiência administrativa por parte dos gerentes e diretores e falta de definição das metas organizacionais.

Já Kerzner (Kerzner, 2001) enumera uma longa lista de fatores que podem comprometer o tempo em ambientes de gerenciamento de projeto, tais como:

- Um trabalho mal feito que precise ser refeito;
- Pobres canais de comunicação;
- Falha na delegação de tarefas;
- Metas e objetivos vagos;
- Restrições burocráticas;

- Gerenciamento exagerado;
- Falta de adequadas ferramentas de projeto.

Contudo, segundo Nerur e outros (Nerur et. al., 2005) as organizações precisam continuar a adaptar suas estruturas, estratégias e políticas para satisfazerem aos novos ambientes e a crescente demanda da sociedade contemporânea por sistemas de informação. As organizações devem buscar alternativas sobre como gerenciar seus projetos de software, visando à diminuição dos fracassos e a melhoria na qualidade de seus produtos e serviços.

## **2.4. Metodologias de Gerência de Projetos**

Segundo Kerzner (Kerzner, 2001) atingir a maturidade em gerência de projetos, pode não ser possível sem um processo repetitivo que possa ser usado em todos os projetos. Este processo repetitivo é referido como metodologia de gerência de projeto.

Rehman (Rehman, 2007) considera que uma metodologia provê um plano em nível estratégico para planejar e controlar projetos de software. Trata-se da combinação de processos inter-relacionados os quais nos dizem “o que deve ser feito?”, mas não “como deve ser feito?”.

Segundo Kerzner, boas metodologias integram outros processos na gerência do projeto e durante a década de 1990 diversas organizações dos mais variados ramos de atuação integraram os seguintes processos em uma única metodologia de gerência de projetos:

**Gerência de Projetos:** Os princípios básicos de planejamento, programação e controle do trabalho.

**Gerência da Qualidade Total:** O processo de garantir que o resultado final irá satisfazer as expectativas de qualidade do cliente.

**Engenharia Concorrente:** O processo de efetuar um trabalho em paralelo ao invés de em série, a fim de comprimir o calendário sem gerar riscos graves.

**Controle de Mudança de Escopo:** O processo de controle das mudanças do projeto a fim de prover um resultado final com valor agregado ao cliente.

**Gerência de Risco:** O processo de identificação, quantificação e resposta aos riscos do projeto.

Kerzner enumera as características de uma boa metodologia baseada em processos integrados:

- Um recomendado nível de detalhe;
- Utilização de modelos;
- Técnicas de planejamento, programação e controle dos custos padronizado;
- Formato de relatórios padronizado, tanto os internos quanto para os clientes;
- Flexibilidade para a aplicação em todos os projetos;
- Flexibilidade para uma rápida introdução de melhorias, conforme necessário;
- Fácil para que o cliente possa entender e seguir;
- Prontamente aceito e utilizado em toda organização;
- Uso de fases do ciclo de vida padronizado (que podem se sobrepor);
- Baseado em orientações e não em políticas e procedimentos;
- Com base em um bom trabalho ético.

Segundo Cheema e Shahid (Cheema & Shahid, 2005) não há uma única e bem definida metodologia de gerência de projeto. Existem muitos métodos para se escolher. Diferentes metodologias são necessárias, dependendo da dimensão do projeto (número de pessoas a serem coordenadas), da responsabilidade dos sistemas que estão sendo criados, as características e prioridades do projeto, bem como, o ambiente em que está envolvido.

Além disso, há um debate na indústria sobre o uso de práticas ágeis versus tradicionais de desenvolvimento de software. Cada abordagem tem vantagens e aplicabilidade no seu próprio contexto. A seção 2.5 discute a gerência de projetos com utilização de práticas tradicionais e a seção 2.6 discute a gerência de projetos com utilização de práticas ágeis. Já as demais seções do capítulo 3 apresentam um estudo sobre estas duas abordagens de gerência de projetos de software.

## **2.5. Gerência de Projetos de Software com utilização de práticas tradicionais**

Dentre as diversas abordagens tradicionais existentes para gerência de projetos, o PMBOK é o que tem maior destaque. Publicado pelo PMI (*Project Management Institute*) em 1996, atualmente o PMBOK encontra-se na terceira versão (2004).

O PMBOK aborda todas as áreas vitais de um bom planejamento e orienta os gerentes de projeto para conseguirem atingir os objetivos dos projetos que conduzem dentro do prazo, orçamento e qualidade exigidos ou, pelo menos, com o mínimo de imprevistos possíveis.

A abordagem tradicional de gerência de projetos baseia-se em processos definidos e documentados que passam por melhorias contínuas nas diversas organizações (Boehm & Turner, 2003). O planejamento detalhado e o processo disciplinado que orientam a gerência de projetos tradicional na engenharia de software permitem a medição e controle de todas as etapas do desenvolvimento de software e da equipe do projeto, onde cada membro tem o seu papel claramente definido e os artefatos gerados em cada fase são os registros da evolução do projeto (Boehm & Turner, 2003).

Segundo Boehm (Boehm, 2002), os tradicionalistas defendem o uso extenso de planejamento, processos codificados e reutilização para fazer do desenvolvimento uma atividade eficaz e previsível que amadurece lentamente em direção a perfeição.

A metodologia de desenvolvimento aplicada nesta modalidade de desenvolvimento de software geralmente são os modelos em cascata ou espiral, embora se perceba um crescimento do modelo iterativo e incremental. Já a arquitetura é definida com foco na reutilização com o objetivo de reduzir o re-trabalho e aumentar a produtividade.

Ainda segundo Boehm e Turner (Boehm & Turner, 2003), na gerência tradicional o foco está no processo, portanto o suporte gerencial, a comunicação e a infraestrutura organizacional são requisitos chaves para o sucesso do empreendimento.

O planejamento geralmente é extenso e detalhado buscando a criação de um cronograma de atividades, pontos de controle e procedimentos que direcionam a geração dos produtos previstos e a coordenação do processo. Este plano é utilizado como referência para medir o progresso durante a fase de execução do projeto e pode sofrer alterações constantes de acordo com a evolução do trabalho (Boehm & Turner, 2003).

A abordagem tradicional é aplicada em qualquer ambiente de projeto, mas nas situações críticas envolvendo prazos restritos e com volatilidade de requisitos apresenta dificuldades em responder com rapidez às mudanças impostas pelos clientes causando, às vezes, conflitos no relacionamento e comprometimento do prazo.

A seguir, são apresentados resumos de algumas das abordagens tradicionais:

### **2.5.1. PMBOK**

Segundo o próprio PMBOK, o objetivo principal do guia é identificar o subconjunto do conjunto de conhecimentos em gerência de projetos que é amplamente reconhecido como boa prática. “Boa prática” significa que existe acordo geral de que a aplicação correta dessas habilidades, ferramentas e técnicas podem aumentar as chances de sucesso em uma ampla série de projetos diferentes. Uma boa prática não significa que o conhecimento descrito deverá ser sempre aplicado uniformemente em todos os

projetos; a equipe de gerência de projetos é responsável por determinar o que é adequado para um projeto específico.

Além de guia de orientação sobre as melhores práticas de GP, o PMBOK pode ser utilizado para orientar a definição de um processo padrão para gerência de projetos de software, pois descreve “o que deve ser feito” e não “como deve ser feito”.

Os 44 processos que compõem a versão 2004 do PMBOK estão organizados em 9 áreas de conhecimento (integração, escopo, tempo, custos, qualidade, recursos humanos, comunicações, risco, aquisições) e 5 grandes grupos (iniciação, planejamento, execução, monitoramento e controle e finalização).

A Tabela 2.4 apresenta as áreas de conhecimento com um breve resumo de seu objetivo, segundo o próprio PMBOK. Para informação adicional consulte o PMBOK.

**Tabela 2.4 - Áreas de Conhecimento do PMBOK. Fonte: PMBOK**

<b>Área do Conhecimento</b>	<b>Resumo do Objetivo</b>
Integração	Identificar, definir, combinar, unificar e coordenar os diversos processos e atividades de gerência de projetos.
Escopo	Inclui os processos necessários para garantir que o projeto inclua todo o trabalho necessário, e somente ele, para terminar o projeto com sucesso.
Tempo	Inclui os processos necessários para realizar o término do projeto no prazo.
Custos	Inclui os processos envolvidos em planejamento, estimativa, orçamento e controle de custos, a fim de que seja possível terminar o projeto dentro do orçamento aprovado.
Qualidade	Inclui todas as atividades da organização executora que determinam as responsabilidades, os objetivos e as políticas de qualidade, de modo que o projeto atenda às necessidades que motivaram sua realização.
Recursos Humanos	Inclui os processos que organizam e gerenciam a equipe do projeto.
Comunicação	Emprega os processos necessários para garantir a geração, coleta, distribuição, armazenamento, recuperação e destinação final da informação sobre o projeto de forma



	oportuna e adequada.
Riscos	Inclui os processos que tratam da realização de identificação, análise, respostas, monitoramento e controle e planejamento da gerência de riscos em um projeto.
Aquisição	Inclui os processos para comprar ou adquirir os produtos, serviços ou resultados necessários de fora da equipe do projeto para realizar o trabalho.

### 2.5.2. CMMI

Conforme mencionado, o *Capability Maturity Model Integration* (CMMI) (SEI, 2002) é um modelo de maturidade de processos de desenvolvimento de produtos elaborado pelo *Software Engineering Institute* (SEI), da Universidade *Carnegie Mellon*, e que está cada vez mais sendo adotado nas empresas de software.

No modelo CMMI foram criados níveis de maturidade referentes aos estágios de maturidade que uma organização possui para desenvolver software. Apesar de abordar a questão da Gerência de Projetos nos níveis mais altos, o foco do CMMI em gerência de projetos ocorre no nível 2: Gerenciado. Este nível possui sete áreas:

- Gerência de Requisitos;
- Planejamento de Projeto;
- Monitoramento e Controle de Projeto;
- Gerência de Acordos de Fornecimento;
- Medições e Análises;
- Garantia da Qualidade de Processos e Produto;
- Gerência da Configuração.

Para cada uma destas áreas, o CMMI definiu um conjunto detalhado de metas e práticas que as organizações (e gerentes de projeto) devem executar a fim de atingirem o nível 2 de maturidade. Alguns exemplos destas metas e práticas são:

- Gerenciar as mudanças de requisitos;

- Manter a rastreabilidade bidirecional dos requisitos;
- Estabelecer estimativas;
- Definir um ciclo de vida para o projeto;
- Desenvolver o plano de projeto;
- Identificar os riscos do projeto;
- Rever os planos que afetam o projeto;
- Monitorar o projeto em relação ao plano;
- Conduzir revisões de progresso;
- Realizar ações corretivas;
- Estabelecer acordo com fornecedores;
- Estabelecer medições objetivas;
- Fornecer os resultados das medidas;
- Acompanhar e controlar mudanças;
- Executar auditorias de configuração.

No nível mais básico (nível 2), o CMMI estabelece o cumprimento de 7 áreas de processo, 15 metas e 128 práticas.

As atividades executadas para a gerência de projetos caracterizam a abordagem do CMMI como uma abordagem tradicional de gerência de projetos, pois estão aderentes ao estilo PMBOK.

### **2.5.3. RUP**

O RUP (RUP, 2003), abreviação de *Rational Unified Process* (ou Processo Unificado da Rational), é um processo proprietário de engenharia de software criado pela empresa *Rational Software Corporation* que foi posteriormente adquirida pela

IBM. Este processo prescreve técnicas a serem seguidas pelos membros da equipe de desenvolvimento de software com o objetivo de aumentar a sua produtividade.

O RUP oferece uma abordagem baseada em uma matriz de disciplinas e fases que atribui tarefas e responsabilidades para a organização de desenvolvimento de software (RUP, 2003). Sua meta é garantir a produção de software de alta qualidade que atenda às necessidades dos usuários dentro de um cronograma e de um orçamento previsíveis.

As disciplinas do RUP são:

- Modelagem de Negócios;
- Requisitos;
- Análise e Design;
- Implementação;
- Teste;
- Implantação;
- Gerência de Configuração e Mudança;
- Gerência de Projeto;
- Ambiente;

Segundo Rehman (Rehman, 2007) estas disciplinas definem gerência de projetos como a arte de equilibrar objetivos concorrentes, gerenciar risco e superar restrições para entregar de forma bem sucedida o produto que satisfaça às necessidades das partes envolvidas.

Segundo o RUP (RUP, 2003) a disciplina de gerência de projetos enfatiza principalmente os aspectos importantes de um processo de desenvolvimento iterativo:

- Gerência de risco;
- Planejamento de um projeto iterativo, por meio do ciclo de vida e de uma iteração particular;

- Monitoramento do progresso de um projeto iterativo.

A Figura 2.3 mostra através de um diagrama de atividades o fluxo de trabalho da disciplina de gerência de projetos no RUP. O perfil utilizado nesta figura é explicado na documentação do RUP.

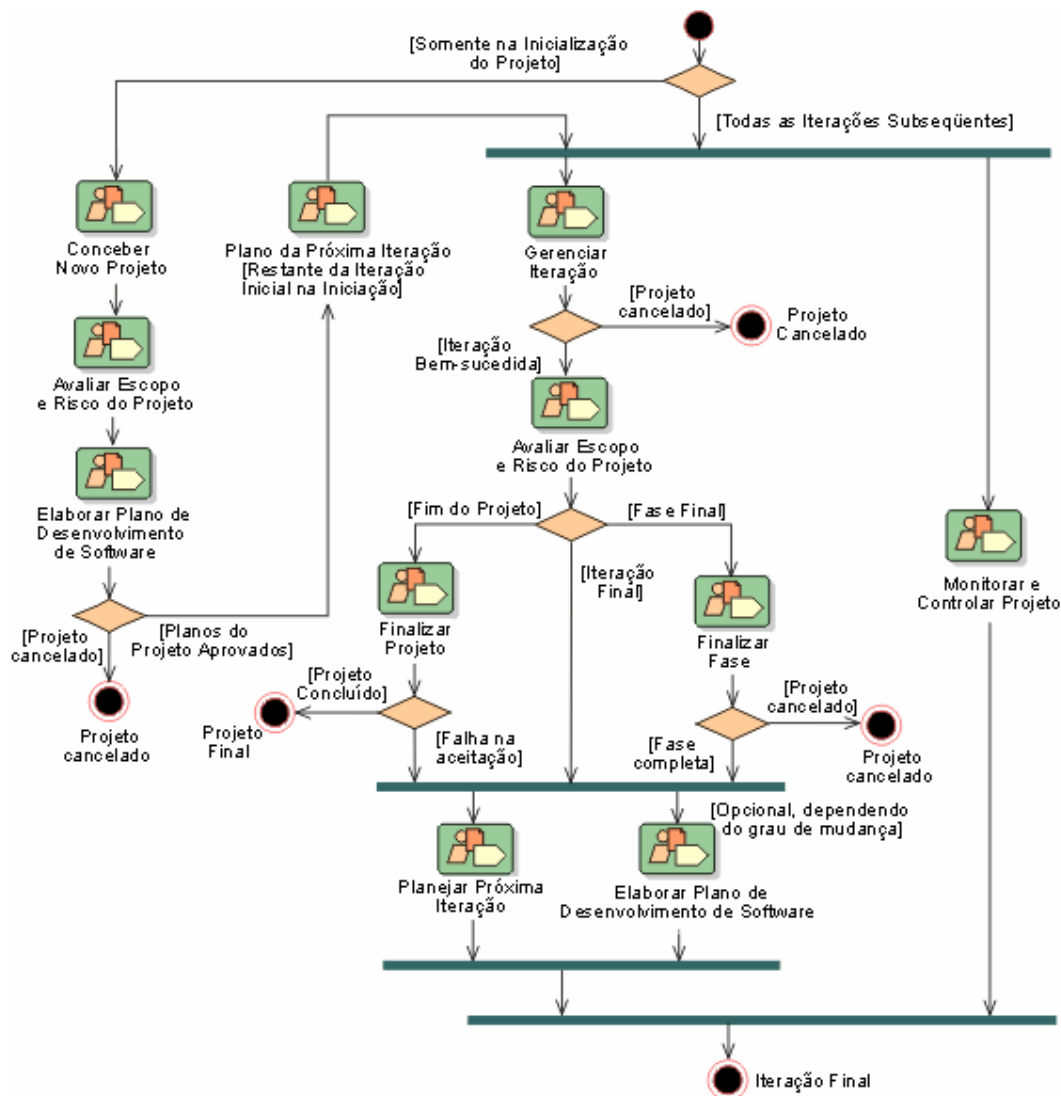


Figura 2.3 - Fluxo de Trabalho da Disciplina de Gerência de Projetos no RUP.

Fonte: (RUP, 2003)

#### **2.5.4. ISO / IEC 15504**

A ISO / IEC 15504 é uma norma que representa um padrão internacional emergente que estabelece um *framework* para construção de processos de avaliação e melhoria do processo de software. Este *framework* pode ser utilizado pelas empresas envolvidas em planejar, gerenciar, monitorar, controlar e melhorar a aquisição, fornecimento, desenvolvimento, operação, evolução e suporte do software (ISO / IEC 15504, 2006).

A ISO / IEC 15504 refere-se a um modelo de capacidade de processo para a engenharia de software, baseado nos processos definidos na nova versão da ISO / IEC 12207. Este modelo tem como usuário típico uma organização intensiva em software que queira realizar um programa de melhoria de processo de software e/ou avaliar a capacidade de seus processos de software (ISO / IEC 15504, 2006).

A norma define um conjunto de práticas básicas que podem orientar e indicar melhorias no processo de Gerência de Projetos de uma organização. Algumas destas práticas são (ISO / IEC 15504, 2006):

- **Defina o escopo do trabalho.** Identifique os objetivos do projeto, motivação e restrições;
- **Defina o ciclo de vida do projeto.** Defina um ciclo de vida e estratégias para o projeto, que sejam apropriadas ao escopo, contexto, tamanho e complexidade;
- **Avalie a viabilidade do projeto.** Avalie a viabilidade do alcance das metas do projeto com os recursos disponíveis e restrições;
- **Defina atividades e tarefas do projeto.** Identifique atividades e tarefas do projeto de acordo com o ciclo de vida definido, e defina dependências entre elas;
- **Defina necessidades de experiência, conhecimento e habilidades.** Identifique os requisitos de experiência, conhecimento e habilidades do projeto e aplique-os na seleção dos indivíduos e equipes;

- **Implemente o plano de projeto.** Implemente as atividades planejadas do projeto, grave a situação de progresso e reporte a situação atual para as partes afetadas;
- **Monitore os atributos do projeto.** Monitore o escopo, orçamento, custo, recursos e outros atributos necessários do projeto e documente desvios significativos em relação à linha de base do projeto;
- **Revise o progresso do projeto.** Regularmente, relatórios e revisões do progresso do projeto devem ser confrontados com o plano de projeto;
- **Corrija desvios.** Aja quando as metas do projeto não são alcançadas, para corrigir desvios do plano e para prevenir retorno de problemas identificados no projeto. Atualize os planos, conforme necessário;
- **Execute a revisão final do projeto.** Execute a revisão final do projeto para fornecer um histórico das experiências a fim de se facilitar a análise de viabilidade de projetos futuros e atualizar a base histórica de estimativas.

Outras abordagens tradicionais de gerência de projetos são descritas na Tabela 2.5.

**Tabela 2.5 - Metodologias de Gerência de Projetos de Software**

Metodologia	Descrição
PRINCE2 ( <i>PR</i> oject <i>C</i> ontrolled <i>E</i> nvironment) <i>IN</i>	Segundo Rehman (Rehman, 2007) a metodologia PRINCE2 é muito utilizada em organizações do Reino Unido e sua estrutura compreende 5 fases e 8 processos de alto nível, sendo 6 processos principais e 2 processos de suporte. Ainda segunda Rehman, PRINCE2 oferece benefícios à organização como eliminação da necessidade de reinventar um método de gerência a cada novo projeto. Uma das desvantagens do PRINCE2 é que ele não oferece suporte para gerência de contratos e de pessoas. Informação sobre Prince2 está disponível em < <a href="http://www.prince2.com">http://www.prince2.com</a> >.

TENSTEP	<p>O grupo TENSTEP é uma afiliação de empresas independentes, presentes na Europa, Ásia e América, focado no desenvolvimento, na distribuição e na implantação de metodologias empresariais, treinamentos e consultoria, com foco em gerenciamento de projetos.</p> <p>Os métodos incluem: processos, procedimentos, técnicas, melhores práticas e modelos.</p> <p>Informação sobre o método TENSTEP está disponível em &lt;<a href="http://www.tenstep.com.br">http://www.tenstep.com.br</a>&gt;.</p>
<i>Microsoft Solution Framework</i> (MSF)	<p>Trata-se de uma combinação de princípios, modelos e melhores práticas estabelecidas pelas Microsoft que ajuda a equipes de projetos de TI a identificar as causas mais comuns de falhas em projetos (Rehman, 2007).</p> <p>Ainda segundo Rehman, o MSF provê um guia bem sucedido para desenvolvimento de aplicações e infra-estrutura de projetos.</p> <p>Informação sobre MSF consulte: &lt;<a href="http://www.microsoft.com/technet/solutionaccelerators/msf/default.aspx">http://www.microsoft.com/technet/solutionaccelerators/msf/default.aspx</a>&gt;.</p>

## 2.6. Gerência de Projetos de Software com utilização de práticas ágeis

Segundo Boehm e Turner (Boehm & Turner, 2003) a agilidade é o contraponto da disciplina. Sempre que disciplina impregna e fortalece, a agilidade liberta e inventa. Este mecanismo permite que os atletas façam jogadas inesperadas, músicos improvisem e ornamentem, artesãos evoluam em seu estilo e engenheiros se adaptem à evolução das tecnologias e necessidades. Agilidade aplica memória e história para se ajustar a novos ambientes, reagir e adaptar, tirar vantagem das oportunidades inesperadas e atualizar a base de experiência para o futuro.

De posse deste conceito, a definição de Gerência Ágil de Projetos torna-se mais fácil. Segundo Highsmith (Highsmith, 2004), Gerência Ágil de Projetos trata-se de um conjunto de valores, princípios e práticas que auxiliam a equipe de projeto a entregar produtos ou serviços de valor em um ambiente complexo, instável e desafiador.

Já Chin (Chin, 2004) define Gerência Ágil de Projetos como uma nova plataforma de gerência de projetos, aplicável a ambientes voláteis e desafiadores, sujeitos as mudanças freqüentes, em que o processo prescritivo e padronizado não mais se adequa. Para Chin, “ágil” não é igual a “rápido”. No entanto, a velocidade é um fator multiplicador de agilidade.

A Gerência Ágil de Projetos caracteriza-se pelo planejamento rápido e não muito detalhado, participação de todos os envolvidos com o projeto, participação efetiva do cliente em todas as fases do desenvolvimento, contribuindo com a elicitação, validação e aceite dos requisitos. Além disso, o ambiente com práticas ágeis estimula a colaboração entre os membros da equipe e encoraja a rápida incorporação de alterações no decorrer do ciclo de vida do projeto.

Além disso, para Chin (Chin, 2004), o papel do Gerente de Projeto em um ambiente ágil ganha novas características: líder, facilitador, orientador e colaborador, ao contrário do perfil controlador e centralizador do ambiente tradicional.

Alguns dos conceitos mais importantes na gerência ágil são: **entregas freqüentes**, com ciclos rápidos de desenvolvimento, **aceite das alterações** e a consciência que elas sempre irão ocorrer, **participação ativa do cliente** e **perfil facilitador do gerente de projeto**.

As seções a seguir apresentam um estudo sobre as abordagens ágil e tradicional com foco no impacto de diversos fatores na gerência de projetos. O objetivo é contribuir no esclarecimento dos conceitos de gerência ágil de projetos bem como orientar os gerentes de projetos e as organizações em relação as melhores práticas que podem ser adotadas de acordo com as características do ambiente, dos projetos e das pessoas.

Além disso, Coram e Bohner (Coram & Bohner, 2005) afirmam que todas as metodologias têm riscos e a compreensão dos riscos bem como a busca por maneiras de controlar, mitigar e gerir estes riscos trata-se de um aspecto importante da gerência



de projetos de software. Segundo Cheema e Shahid (Cheema & Shahid, 2005) projetos e equipes diferem, portanto uma única metodologia não cabe a todos eles.

### **2.6.1. Metodologia Ágil de Desenvolvimento**

Segundo Highsmith e Cockburn (Highsmith & Cockburn, 2001) as abordagens tradicionais de desenvolvimento de software presumem que podemos antecipar precocemente o conjunto completo de requisitos e reduzir custos, em função da eliminação de mudanças. No entanto, a tentativa de eliminar antecipadamente a possibilidade de mudanças significa ser insensível às condições empresariais, o que em outras palavras leva ao insucesso empresarial.

Ainda segundo Highsmith e Cockburn, a abordagem de gerência tradicional – por medição contínua, identificação dos erros e refinamentos no processo – tenta conduzir as variações para fora dos processos. Esta abordagem pressupõe que as variações são os resultados de erros. Contudo, o cenário atual exige exatamente o contrário. A gerência de projetos de software deveria não somente encorajar, mas também, “abraçar” as mudanças.

Em reconhecimento a essas idéias, em fevereiro de 2001, um grupo de pessoas representando métodos alternativos de desenvolvimento de software assinou o “Manifesto Ágil do Desenvolvimento de Software” e definiram quatro itens onde foram feitas priorizações em um confronto de aspectos, valorizando:

- Indivíduos e interações sobre processos e ferramentas;
- Software funcionando sobre documentação abrangente;
- Colaboração com o cliente sobre negociação de contratos;
- Responder as mudanças sobre seguir um plano.

Para os signatários do Manifesto Ágil, embora os itens à direita sejam valorizados, os itens à esquerda agregam mais valor.

Boehm e Turner (Boehm & Turner, 2005) identificam os métodos ágeis como “processos leves” (*lightweight processes*) que empregam ciclos iterativos curtos, envolvem ativamente os usuários para estabelecer, priorizar e verificar os requisitos, e contam com uma equipe baseada em conhecimento tácito, em oposição à documentação. Um método ágil deve ser iterativo (leva vários ciclos para completar o trabalho), incremental (não entregam a totalidade do produto de uma só vez), auto-organizado (a equipe determina a melhor maneira de lidar com o trabalho) e emergente (processos, princípios e estruturas de trabalho são reconhecidos no decurso do projeto em vez de serem pré-determinados).

Abrahamsson e outros (Abrahamsson et. al., 2003) afirmam que métodos ágeis focam a simplicidade e a rapidez. Os desenvolvedores focam apenas nas funções necessárias para entregar rapidamente, coletar realimentação e reagir rapidamente às mudanças tecnológicas e de negócios.

Cockburn e Highsmith (Cockburn & Highsmith, 2001) defendem que a idéia dominante no desenvolvimento ágil é que a equipe possa ser mais eficaz na resposta a mudança:

- reduzindo o custo da troca de informação entre as pessoas, e;
- reduzindo o tempo decorrido entre a tomada de uma decisão e o aparecimento das conseqüências dessa decisão.

Apesar da crescente utilização de métodos ágeis, muitas organizações ainda evitam o seu uso, utilizando como justificativa para tal uma série de barreiras à adoção dos mesmos. Boehm e Turner (Boehm & Turner, 2005) identificaram uma lista destas barreiras, mas chegaram à conclusão que grande parte delas não são, de fato, problemas à adoção de métodos ágeis. Isto é, podem ser vencidas pelas organizações. Alguns exemplos destas barreiras são:

- Garantia de segurança nos sistemas;
- Agilidade é inadequada para gerência de defeitos;

- Efetividade de testes automáticos na aceitação e integração de sistemas;
- Percepção que agilidade é extrema ou uma moda, não responsável;
- Projetos ágeis são “ingerenciáveis”.

As subseções a seguir apresentam um resumo sobre alguns dos métodos ágeis utilizados pela indústria de software, segundo Abrahamsson e outros (Abrahamsson et. al., 2003).

### 2.6.2. Extreme Programming (XP)

Criado em 1997, por Kent Beck, o mais conhecido dos métodos ágeis, o *eXtreme Programming* possui um ciclo de vida composto por cinco fases: Exploração, Planejamento, Iterações para liberação, Produção, Manutenção e Morte.

Beck (Beck, 1999) define da seguinte forma as fases do ciclo de vida XP:

**Exploração:** tipicamente leva de algumas semanas a alguns meses para os clientes fornecerem requisitos para a primeira liberação. Ao mesmo tempo, a equipe do projeto torna-se familiarizada com a tecnologia, ferramentas e práticas que serão utilizadas no projeto.

**Planejamento:** a equipe do projeto prioriza as funcionalidades necessárias para a primeira liberação. Os desenvolvedores estimam o esforço necessário e o líder da equipe elabora um calendário de entregas de forma há não exceder dois meses.

**Iterações para liberação:** várias iterações produzem a primeira liberação. Cada iteração leva de uma a quatro semanas, e no final de cada uma, os testes funcionais são executados. A conclusão da última iteração sinaliza o início da fase de Produção.

**Produção:** a equipe do projeto realiza testes adicionais de desempenho e verificações para assegurar que a liberação satisfaça as exigências dos clientes. Novas alterações poderão ser introduzidas e a decisão deve ser tomada se os mesmos devem ser incluídos na versão atual. Esta fase termina com a liberação entregue ao cliente.

**Manutenção:** a equipe produz novas iterações do produto de software para implementar mudanças e novas funcionalidades de solicitações levantadas na fase anterior. Incluem mudanças corretivas, perfectivas e adaptativas efetuadas durante a manutenção.

**Morte:** implica concluir toda a documentação necessária e a implantação do sistema é planejada. Esta fase ocorre quando o valor proposto para evoluir o sistema se torna muito alto.

Segundo Abrahamsson e outros (Abrahamsson et. al., 2003) algumas das principais características do XP são curtas iterações com pequenas entregas e rápida realimentação, participação próxima do cliente, constante comunicação e coordenação, refatoração contínua, integração e testes contínuos, posse coletiva do código e programação aos pares.

### 2.6.3. Scrum

Scrum pode ser visto como um método incremental e iterativo, orientado para a gerência dos projetos de software. O ciclo de vida Scrum consiste em três fases: Pré-jogo, Desenvolvimento e Pós-jogo.

Segundo Schwaber (Schwaber, 2004), criador do Scrum, na fase de **pré-jogo**, há duas subfases: Planejamento e Desenho de Arquitetura. Planejamento implica a definição do sistema baseado num *Product Backlog List* (atualizado freqüentemente com funcionalidades e modificações), que contém todos os requisitos conhecidos. Estes são priorizados e o esforço necessário é estimado. Na subfase de Desenho de Arquitetura, o desenho do sistema é desenvolvido e aperfeiçoado baseado na lista de *backlog*.

Na fase de **desenvolvimento**, ciclos iterativos de desenvolvimento chamados “*Sprints*” são executados para desenvolver novas funções e para reforçar o sistema. Cada *sprint* inclui: requisitos, análise, design, evolução e entrega. Cada *sprint*

compreende desde uma semana a um mês. Três a oito *sprints* são executados no método de desenvolvimento antes do sistema estar concluído (Coram & Bohner, 2005).

A fase **pós-jogo** conclui o esforço e proporciona a liberação sem funcionalidades adicionais ou modificações. Diferentemente do XP, não existe nenhuma fase específica para distribuição do sistema (Coram & Bohner, 2005).

De acordo com Abrahamsson e outros (Abrahamsson et. al., 2003) o método Scrum deixa em aberto para os desenvolvedores a escolha de técnicas, métodos e práticas específicas para o desenvolvimento de software, a serem utilizados durante a implementação. O Scrum envolve freqüente gerência das atividades com o objetivo de identificar consistentemente algum impedimento ou deficiência no método de desenvolvimento, bem como as práticas que estão sendo utilizadas.

#### **2.6.4. DSDM**

Segundo Coram e Bohner (Coram & Bohner, 2005) um aspecto importante que distingue o método DSDM (Método Dinâmico de Desenvolvimento de Software) é que ele corrige prazos e recursos em primeiro lugar e, em seguida, ajusta a quantidade de funcionalidades a serem desenvolvidas. Este processo de recursos-primero consiste em cinco fases: estudo de viabilidade, estudo de negócios, iteração de modelo funcional, iteração de concepção e construção e implementação. As últimas três fases são iterativas e incrementais – restringindo iterações dentro de períodos pré-definidos de tempo.

Resumidamente, a fase de estudo de viabilidade é responsável pela avaliação do produto e a fase de estudo do negócio define a arquitetura do sistema a partir da avaliação das características do negócio e tecnologia. Durante a fase de iteração de modelo funcional, o projeto evolui através de iterações que envolvem algumas melhorias e os incrementos são voltados para o sistema final. Na iteração de concepção e construção o sistema é produzido baseado em observações do cliente. Por fim, na fase

de implementação o sistema é formalmente transferido para o produto atual. O sistema é entregue ao cliente e quaisquer acréscimos são planejados (Coram & Bohner, 2005).

Segundo Abrahamsson e outros (Abrahamsson et. al., 2003) DSDM pode ser visto como o primeiro método verdadeiramente ágil de desenvolvimento de software.

### **2.6.5. Crystal**

Segundo Cockburn (Cockburn, 2004), o método Crystal abrange equipes de diversas dimensões. Cockburn acredita que são necessárias abordagens diferentes quando as equipes variam em tamanho, além da criticidade dos erros em mudanças.

Apesar de suas variações, o método Crystal foca a eficiência e habilidade da equipe como componente de garantia de qualidade do projeto. Além disso, o método foca as pessoas e não as atividades e artefatos.

Algumas das características do método Crystal são (Cockburn, 2004):

- Entrega freqüente de código utilizável pelo usuário;
- Melhoria contínua refletiva;
- Foco – trabalhar sobre as prioridades;
- Fácil acesso para usuários experientes;
- Testes automatizados, gerência de configuração e integração freqüente.

### **2.6.6. ASD**

Criado por Highsmith, o ASD (*Adaptive Software Development*) propõe uma forma alternativa de se enxergar o desenvolvimento de software nas organizações (Highsmith, 2002). O ASD foi projetado para lidar com ambientes complexos e repletos de incertezas. O ASD é indicado para equipes pequenas, mas pode ser

adaptado para equipes maiores, encoraja o desenvolvimento iterativo e incremental, com liberações constantes (Highsmith, 2002).

### **2.6.7. FDD**

O *Feature-Driven Development* (FDD) criado por Peter Coad e Jeff DeLuca em 1999, é um método de desenvolvimento de software específico para aplicações críticas de negócio (Palmer & Felsing, 2001). Diferente de outros métodos ágeis, o FDD se baseia em processos bem definidos e que podem ser repetidos. Sua abordagem se concentra nas fases de projeto e construção, com maior ênfase na modelagem, em um ciclo de vida iterativo e também em atividades de gerenciamento de projetos (Palmer & Felsing, 2001).

## **2.7. Ferramentas de Apoio à Gerência de Projetos**

O gerente de projetos, além de possuir habilidades e conhecimentos, deve também utilizar ferramentas que dão apoio às suas atividades.

Pressman (Pressman, 2002) define ferramentas de gerência de projetos como as ferramentas de planejamento (estimativas de custo e esforço e planejamento do projeto) bem como as ferramentas de apoio ao acompanhamento e controle do projeto. Segundo Pressman, uma ferramenta de planejamento deve dar suporte ao cálculo do esforço estimado, a duração do projeto e o número de recursos recomendados além de permitir a definição de todas as tarefas do projeto e a interdependência entre elas.

Pressman afirma ainda que uma ferramenta de gerência de projeto deva ser utilizada para acompanhar e monitorar o cronograma do projeto e o plano de projeto em uma base contínua. Além disso, uma ferramenta de gerência de projetos pode fornecer ligações para outras ferramentas que dão suporte para outros processos de desenvolvimento de software.

Já o PMBOK define ferramenta (ou software) de gerência de projetos como um tipo de aplicativo de software especificamente projetado para auxiliar a equipe de gerenciamento de projetos no planejamento, monitoramento e controle do projeto, incluindo: estimativa de custos, elaboração de cronogramas, comunicação, colaboração, gerenciamento de configuração, controle de documentos, gerenciamento de registros e análise de risco.

Outras atividades que podem ser auxiliadas e apoiadas por ferramentas de gerência de projetos são (Kerzner, 2001):

- Gerência de calendário e agenda;
- Gerência de recursos-humanos e materiais;
- Gerência de custos;
- Monitoramento do projeto;
- Auditoria do projeto;
- Gerência de múltiplos projetos.

As ferramentas de apoio à gerência de projetos podem ser divididas em 3 categorias quanto ao seu tipo de comercialização:

- Código fechado: são os softwares distribuídos com o código inacessível ao usuário final. Podem ser pagas ou não;
- Código aberto: o código fonte está disponível e pode ser lido, estudado e modificado. Geralmente são oferecidos gratuitamente;
- Modelos científicos: são protótipos de software ainda em lançamento, propostos geralmente em estudos acadêmicos.

Atualmente, o número de ferramentas existentes para gerência de projetos é expressivo. O sítio especializado em ferramentas de gerência de projetos *Project Management Software* (Project Management Software, 2008) possui 435 ferramentas



catalogadas. Além da classificação comercial, as ferramentas também podem ser categorizadas em ferramentas *Desktops* e ferramentas *Web*.

Apesar da grande quantidade, cada ferramenta possui um direcionamento e um ambiente mais adequado para sua utilização. Uma das ferramentas comerciais mais utilizadas, por exemplo, Microsoft Project, lançado em 1987, vem evoluindo no sentido de atender às características da abordagem tradicional de gerência de projetos.

Segundo Araujo (Araujo, 2008), o ambiente atual de desenvolvimento de software – dinâmico e incerto, com escopo impreciso e em constante mudança – é diferente daquele em que as primeiras ferramentas de gerência de projetos foram criadas. Ainda segundo Araujo, as ferramentas de gerência de projetos devem atender aos seguintes desafios:

- Adequar-se as mudanças nos projetos;
- Apoiar o acompanhamento e alterações no resultado final;
- Compartilhar recursos entre projetos;
- Consumir pouco tempo com documentação;
- Empregar os princípios da gerência ágil de projetos.

White e Fortune (White & Fortune, 2002) atestam a necessidade de evolução das ferramentas de apoio à gerência de projeto em seu trabalho de pesquisa com organizações e gerentes de projeto. Neste estudo, as autoras afirmam que softwares de gerência de projetos aparecem como uma das principais limitações entre os métodos, ferramentas e técnicas de gerência de projetos, especialmente devido às inadequações as necessidades das organizações em projetos colaborativos.

Todavia, alguns autores ((Liu, 2006), (Araujo, 2008), (Cannizzo et. al., 2008) e (Vriens & Barto, 2008)) citam as ferramentas comumente utilizadas, tanto para as abordagens tradicionais quanto para as abordagens ágeis de gerência de projeto:

- Microsoft Project;
- Primavera;

- dotProject;
- GanttProject;
- OpenProject;
- VersionOne;
- ScrumWorks;
- PPTS (Project Planning and Tracking System); e
- XPlanner.

Como complementação, este trabalho apresenta uma personalização da XPlanner, ferramenta de apoio à gerência de projetos com foco em organizações pequenas e médias que trabalham com multi-projetos, equipes distribuídas e em ambiente colaborativo. Esta personalização é apresentada nos capítulos 4 e 5.

## **2.8. Considerações Finais**

Apesar dos já conhecidos benefícios que a Gerência de Projetos proporciona às organizações, ainda é elevado o número de empresas de software que simplesmente ignoram esta disciplina no seu ciclo de desenvolvimento.

Aliado a isto, há o duelo entre a abordagem tradicional e a abordagem ágil de gerência de projetos. Cada uma delas possui seu ambiente mais adequado, que deve ser bem explorado pelo gerente de projetos.

Ainda neste contexto, observamos a questão das diferentes ferramentas de suporte à gerência de projetos de software. Geralmente associadas ao processo de desenvolvimento, seu ambiente de uso mais apropriado também depende das características ágeis ou tradicionais dos projetos e das organizações.

## Capítulo 3

# Um estudo sobre a utilização de práticas ágeis e práticas tradicionais na Gerência de Projetos de Software

Este capítulo apresenta um estudo comparativo entre a abordagem ágil e a abordagem tradicional de gerência de projetos de software. Inicialmente, a seção 3.1. apresenta uma introdução e contextualização desta comparação. Já as seções seguintes descrevem a comparação através de quatro tópicos principais: organizações, projetos, pessoas e papéis do gerente de projeto.

### **3.1. Contextualização**

O desenvolvimento de software, ao longo dos anos, tem sofrido diversas transformações a fim de se adaptar a nova realidade das organizações, projetos, clientes e usuários. As organizações atuam em mercados cada vez mais competitivos, onde a qualidade dos produtos já não é somente um diferencial, e sim uma obrigação. Os projetos possuem requisitos instáveis e devem ser entregues em prazos cada vez menores. Já os clientes e usuários são mais exigentes e cientes da importância dos

projetos de TI em suas organizações. Mudanças na tecnologia também impactam diretamente no surgimento de novas metodologias de gerência de projetos.

Segundo Boehm (Boehm, 2002), de um lado os tradicionalistas defendem o uso extenso de planejamento, processos codificados e reutilização para fazer do desenvolvimento uma atividade eficaz e previsível que amadurece lentamente em direção à perfeição. Por outro lado, uma nova geração de desenvolvedores cita a burocracia corporativa, o ritmo acelerado das mudanças nas tecnologias da informação e os detalhados planos de desenvolvimento como motivo de uma revolução. Estes desenvolvedores apelam a uma abordagem revitalizada que simplifique as atividades de desenvolvimento.

Nerur e outros (Nerur et. al., 2005) afirmam que a abordagem tradicional sente falta de flexibilidade para dinamicamente ajustar-se ao processo de desenvolvimento. Ainda segundo Nerur e outros, novos métodos chamados Metodologia de Desenvolvimento Ágil ou Metodologia Leve reivindicam a possibilidade de superação nas limitações da abordagem tradicional.

Nerur e outros afirmam ainda que vários estudos evidenciam claramente a popularidade crescente da metodologia ágil. Por outro lado, segundo Coram e Bohner (Coram & Bohner, 2005) algumas organizações afirmam que métodos ágeis resolvem todos os seus problemas, porém poucas têm mostrado consistente sucesso ao longo de uma série de projetos típicos de software.

No cenário onde os gerentes de projetos devem optar pelos métodos de desenvolvimento e abordagens de gerência de projeto mais apropriadas às suas necessidades, diversas pesquisas buscam auxiliá-los na comparação entre elas.

Boehm e Turner (Boehm & Turner, 2005) apresentam os desafios gerenciais para a implementação de métodos ágeis em organizações tradicionais de desenvolvimento. Questões como variabilidade nos requisitos, sistemas legados, recursos humanos e atitudes gerenciais são apresentadas e discutidas. Sugestões para os gerentes de projetos também são citados no trabalho.

Cheema e Shahid (Cheema & Shahid, 2005) listam as etapas para a personalização de um processo de gerência de projetos para uma organização. No trabalho, 15 parâmetros de personalização foram identificados e, através da análise destes parâmetros, o gerente de projeto deve selecionar qual a metodologia mais apropriada para sua organização e seus projetos.

Já Nerur e outros (Nerur et. al., 2005) apresentam os desafios relacionados à migração do desenvolvimento tradicional para o desenvolvimento ágil. O trabalho divide estes desafios em 4 grupos principais: gerência e organização, pessoas, processos e tecnologia. De acordo com Nerur e outros, métodos ágeis são ideais para projetos que apresentam alta variabilidade em tarefas (devido à mudança de requisitos), nas capacidades das pessoas e nas tecnologias a serem utilizadas. Segundo os autores, as organizações devem avaliar sua estrutura interna antes de começar a utilizar as características de agilidade.

Outro trabalho que também apresenta uma comparação entre as abordagens ágil e tradicional foi realizado por Hirsch (Hirsch, 2005). Através da substituição de uma cultura tradicional para uma cultura ágil de desenvolvimento, o autor compara os pressupostos fundamentais, as disciplinas de desenvolvimento e os papéis das pessoas envolvidas com o projeto. Alguns diferenciais deste trabalho são a exposição das lições aprendidas durante o processo e algumas motivações para a substituição e a discussão sobre possíveis resistências internas à adoção de práticas ágeis e meios de contorná-las.

Neste sentido, o nosso trabalho apresenta um estudo sobre as abordagens ágil e tradicional de gerência de projetos de software. A fim de facilitar o estudo, optamos pela divisão em 4 sub-tópicos principais: organização, projetos, pessoas e papéis do gerente de projeto. Esta estrutura foi baseada no trabalho de Nerur e outros (Nerur et. al., 2005). Porém, realizamos adequações no sentido de expandir a comparação, agregando estudos de outros trabalhos e incluindo questões relacionadas às características dos projetos e aos papéis fundamentais do gerente de projeto.

## **3.2. Organizações**

Segundo Nerur e outros (Nerur et. al., 2005) durante décadas as organizações têm perseguido o objetivo de criar processos otimizados e repetíveis. A estabilidade para eles apresenta um dos maiores obstáculos à adoção de métodos ágeis de desenvolvimento. Ainda segundo Nerur e outros a cultura organizacional tem um impacto significativo sobre a estrutura social das organizações, que por sua vez influenciam o comportamento e as ações das pessoas. Além disso, nem a cultura nem a intenção das pessoas podem ser facilmente alteradas, o que torna a adoção dos métodos ágeis mais complicadas para muitas organizações.

Uma pesquisa realizada por Ceschi e outros (Ceschi et. al., 2005) aponta que o principal problema para as organizações que desenvolvem software é fornecer produtos com todas as funções dentro do prazo. Além disso, esta mesma pesquisa indica que o relacionamento com o cliente geralmente é um ponto crítico.

Além disso, os fatores de tamanho, processos utilizados pela organização e tecnologia têm conflitos e diferenças entre as abordagens ágeis e tradicionais.

### **3.2.1. Fatores de Negócios**

As obrigações contratuais estão entre os principais fatores que afetam as organizações com relação à adoção dos métodos ágeis. Para muitas empresas contratantes, o que está a ser efetuado pelos contratados é determinado por uma declaração do trabalho definindo os requisitos essenciais e as tarefas. A documentação é muitas vezes utilizada dentro de um relacionamento de contratação para indicar qual trabalho foi feito, além de documentar o progresso e fornecer a transição para a empresa contratante do trabalho. Contratos de governo têm frequentemente documentação significativa de requisitos tais como a observância de normas como ISO 9000 (Coram & Bohner, 2005).

Ainda segundo Coram e Bohner, se o negócio requer a capacidade de especificar datas para acomodar os clientes, isto representa um tipo de contrato. Produtos que exigem um roteiro para funcionalidades bem definidas podem ser incapazes de utilizar eficazmente métodos ágeis.

A remoção de uma atitude voltada para a produção de documentos para uma atitude mais colaborativa pode não ser possível, e trata-se de um desafio para a organização e para o gerente de projeto.

Além disso, cabe à organização e ao gerente de projeto estudar e definir regras em relação a uma possível relação com fornecedores sub-contratados. Devido à alta volatilidade dos requisitos e a documentação mínima, uma relação neste contexto entre fornecedores pode se tornar difícil.

### **3.2.2. Processos**

Processos tradicionais são dirigidos à conformidade e baseados em atividades e em medição, com o objetivo de fornecer garantias (Boehm, 2002). A metodologia ágil confia na especulação ou planejamento com a compreensão de que tudo é incerto, para orientar o rápido desenvolvimento de sistemas flexíveis e adaptáveis de alto valor (Nerur et. al., 2005). Eles salientam a importância da avaliação em oposição à medição, e são tolerantes à mudança. Um dos maiores obstáculos à migração é a mudança de um processo baseado em modelo de ciclo de vida para um que suporte desenvolvimento baseado em funcionalidades usando desenvolvimento evolutivo e iterativo. Essa mudança implica grandes alterações nos procedimentos de trabalho, ferramentas e técnicas, canais de comunicação, estratégias de resolução de problemas e papéis de pessoas (Nerur et. al., 2005).

A fim de se contornar os problemas, principalmente de adaptação de uma lista de processos a serem empregados na organização, Boehm e Turner (Boehm & Turner, 2005) sugerem a realização de uma análise dos processos existentes e propostas para identificar inadequações nas premissas e expectativas do processo. Segundo Boehm e

Turner, o ideal é construir novos processos em vez de adequá-los: “olhe para as necessidades do projeto e selecione somente os processos ativos que parecem indispensáveis”. Evidentemente, para algumas instâncias pode ser necessário mais rigor e adição de componentes necessários ao processo.

Para alguns defensores de processos, a adoção de técnicas e características ágeis pode parecer como um caminho sem controle e sem garantias que o produto final será construído. É importante que o Gerente de Projeto Ágil saiba transmitir a eficácia dos métodos ágeis, porém sem esquecer de dar atenção aos processos realmente indispensáveis e que vão contribuir para o desenvolvimento do produto.

### **3.2.3. Tecnologia**

Uma tecnologia existente na organização pode impactar os esforços para migrar para metodologias ágeis. As empresas que dependem exclusivamente de tecnologias *mainframe* podem ter dificuldade em assimilar métodos ágeis, em comparação com aqueles que usam técnicas orientadas a objetos de desenvolvimento (Nerur et. al., 2005).

Segundo Nerur e outros (Nerur et. al., 2005), ferramentas desempenham um papel crucial na implementação bem sucedida da metodologia de desenvolvimento de software. Organizações que planejam adotar métodos ágeis devem investir em ferramentas que apóiam e facilitam o desenvolvimento iterativo e rápido, gerência de versões / configuração, ferramentas de automatização de testes, refatoração e outras técnicas ágeis.

O emprego das ferramentas deve ser acompanhado com o adequado treinamento às pessoas que irão utilizá-las. Neste ponto, o papel da gerência de projetos é crucial. A organização deve estar ciente dos impactos que o uso de determinada tecnologia podem causar a partir da adoção de metodologias ágeis.



### **3.2.4. Tamanho e Distribuição**

Alguns fatores contribuem para que a Gerência Ágil de Projetos se ajuste melhor a algumas organizações que outras. Principalmente devido à documentação informal e pouco extensa, a necessidade de participação ativa dos clientes durante todo o projeto, o encorajamento à comunicação intensiva dentro da equipe e algumas práticas realizadas durante o desenvolvimento, as empresas menores e cujas equipes dos projetos preferencialmente não sejam distribuídas possuem um ambiente mais propício à gerência ágil.

Segundo Schwaber (Schwaber, 2004), em Scrum, para organizações e projetos maiores, a equipe de desenvolvimento e os projetos devem ser divididos em sub-equipes e subprojetos possuindo em torno de oito pessoas. Contudo, isto pode dificultar a gerência em casos em que esta divisão não seja viável.

Já Cohn e Ford (Cohn & Ford, 2003) afirmam que equipes utilizando métodos ágeis tendem a tomar decisões mais rapidamente do que as equipes tradicionais, baseando-se em comunicação mais freqüente (e geralmente informal) para apoiar este ritmo. Diante disto, os autores dizem que as organizações ágeis devem resistir às equipes distribuídas, contudo, quando for necessário combinar as equipes, o ideal é manter o maior número possível de pessoas em conjunto durante a primeira e segunda semana do projeto.

## **3.3. Projetos**

Cheema e Shahid (Cheema & Shahid, 2005) identificaram vários parâmetros de um projeto, que devem ser levados em conta antes de escolher uma metodologia de desenvolvimento e gerência. Dentre estes, cabe citar: natureza do projeto, tamanho do projeto e da equipe, número e localização dos participantes, flexibilidade dos requisitos,

disponibilidade do cliente, iteratividade do processo de desenvolvimento e nível de habilidade da equipe.

Além disso, as abordagens ágil e tradicional possuem objetivos primordiais contrastantes, bem como, questões de documentação e planejamento e controle também possuem características diferentes.

### **3.3.1. Objetivo Primordial**

Segundo Boehm (Boehm, 2002) o primeiro princípio do manifesto ágil afirma que “A nossa maior prioridade é satisfazer o cliente através do fornecimento mais cedo e contínuo de software de valor”. “Mais cedo” e “continuamente” são metas razoavelmente compatíveis para os pequenos sistemas desenvolvidos por pessoas de grande competência. Mas o foco sobre resultados precoces em grandes sistemas pode levar a maior re-trabalho quando a arquitetura não está escalável.

Já os processos tradicionais são mais precisos em alta garantia. Ainda de acordo com Boehm, outro conjunto de objetivos para os processos tradicionais são a previsibilidade, repetibilidade e otimização. Contudo, em um mundo envolvendo mudanças freqüentes e radicais, processos repetitivos e otimizados podem não satisfazer bem os objetivos.

Apesar de terem conceitos e objetivos primordiais diferentes, o resultado final a ser esperado é (e tem que ser) o mesmo: a execução do projeto da maneira mais eficiente possível a fim de se desenvolver um produto de valor para o cliente e dentro de limites aceitáveis de tempo e custo.

### **3.3.2. Processo de Desenvolvimento**

De acordo com Boehm e Turner (Boehm & Turner, 2005) esta é a primeira e talvez mais óbvia área de dificuldade. Como fundir métodos ágeis com processos

tradicionais, sem um ou outro matar a agilidade ou prejudicar os anos que foram gastos em definição e aperfeiçoamento dos sistemas e processos existentes?

Métodos ágeis freqüentemente incentivam princípios que mudam radicalmente o processo utilizado em abordagens tradicionais. Segundo Coram e Bohner (Coram & Bohner, 2005) atividades chaves de desenvolvimento ágil, que podem impactar o processo de desenvolvimento da organização “tradicional” são refatoração, revisões de código, integração contínua e a abordagem de “testes primeiro”.

De acordo com Coram e Bohner (Coram & Bohner, 2005) refatoração é a técnica de melhorar o código, sem perder nenhuma funcionalidade. O código poderia ser melhorado para legibilidade, manutenibilidade, desempenho.

Outra questão importante para os gerentes de projetos é sobre a abordagem conhecida como “testes primeiro”. Segundo Nerur e outros (Nerur et. al., 2005) a noção de desenvolvimento guiado pelo teste, muito utilizado em métodos ágeis, é motivada pelo fato de que pensar e escrever testes antes da codificação poderá tornar o código mais compreensível e manutenível.

Coram e Bohner (Coram & Bohner, 2005) definem revisões de código como a atividade pelo qual um ou mais desenvolvedores analisam o código escrito por outro. Isto poderia ser contínuo, como no aspecto da programação aos pares do XP ou ser periódico como as revisões aos pares incorporadas na DSDM. A principal vantagem das revisões de código neste contexto é que eles servem como um método de comunicação.

Além dos aspectos de implementação, a comunicação durante o desenvolvimento do projeto é diferente entre as abordagens ágil e tradicional. Enquanto que no ambiente ágil a comunicação é informal, porém intensa, promovendo constante realimentação e transparência, no ambiente tradicional a comunicação é formal e documentada, com fluxos bem definidos de aprovação, acompanhamento e divulgação da informação do trabalho. Todavia, o gerente de projeto deve cuidar para que o excesso de informalidade em um projeto ágil não cause conflitos entre os envolvidos.

Coram e Bohner (Coram & Bohner, 2005) sugerem que pode ser necessário que o gerente de projeto incorpore estas técnicas lentamente e com incentivos a fim de aumentar as chances de sua aceitação. Além disso, é responsabilidade do gerente de projeto o importante papel de institucionalizar as práticas no desenvolvimento dos processos tradicionais e dos métodos ágeis, facilitando a adoção por parte da equipe.

### **3.3.3. Planejamento e Controle**

Segundo Coram e Bohner (Coram & Bohner, 2005) métodos ágeis são caracterizados pela menor ênfase no planejamento formal. Isto não quer dizer que o planejamento não ocorra. Por exemplo, a decisão sobre o que será entregue em cada iteração é tomada através da reunião diária Scrum, que debate os problemas pendentes, priorizando o trabalho e atribuindo recursos para os problemas.

Contudo, de acordo com Boehm (Boehm, 2002), comparados com codificação não-planejada e não-disciplinada, métodos ágeis enfatizam uma quantidade considerável de planejamento. Como regra, porém, coloca-se mais valor sobre o processo de planejamento do que sobre a documentação resultante, assim estes métodos muitas vezes parecem menos orientados a plano do que eles realmente são. Boehm critica o excesso de detalhes no planejamento, pois planos excessivamente pré-especificados sobre-forçam a equipe de desenvolvimento, mesmo em menores níveis de mudança de equipe, tecnologia ou de mercado.

Boehm (Boehm, 2002) critica ainda o planejamento excessivo quando uma organização pretende lançar um novo produto no mercado. Segundo Boehm, um longo planejamento irá causar prejuízos para a organização, pois os atrasos irão deixar os concorrentes mais fortes, pois estes podem capturar uma fatia maior do mercado.

À primeira vista, pode parecer que o planejamento ágil não define ou atende aos prazos definidos pelo cliente, contudo, como é recomendável que o cliente trabalhe junto com a equipe, definindo as prioridades e acompanhando de perto o desenvolvimento dos requisitos, o atrito causado por algum possível atraso é reduzido.

Em relação ao controle do projeto, Boehm e Turner (Boehm & Turner, 2005) afirmam que nos contratos tradicionais, marcos importantes e técnicas de medição de progressos poderiam ser insuficientes para suportar o ritmo rápido de métodos ágeis. Contratos e pagamentos que tendem a serem baseados em marcos com revisões preliminares quase não tem significado em ambiente ágil. Pois, enquanto no ambiente tradicional o controle tem alto grau de formalização e todas as ocorrências são documentadas como informação histórica, na metodologia ágil a presença do cliente entre a equipe favorece a informalidade e o conhecimento implícito. Boehm e Turner sugerem, então, que outras técnicas sejam utilizadas em métodos ágeis, tal como o Gráfico de Progresso (Gráfico de *Burndown*) que apresenta o progresso de implementação de uma iteração ou *release*. Sendo que o eixo X representa os dias e o eixo Y o trabalho restante (em horas, por exemplo).

A Figura 3.1 ilustra um exemplo de Gráfico de Progresso.

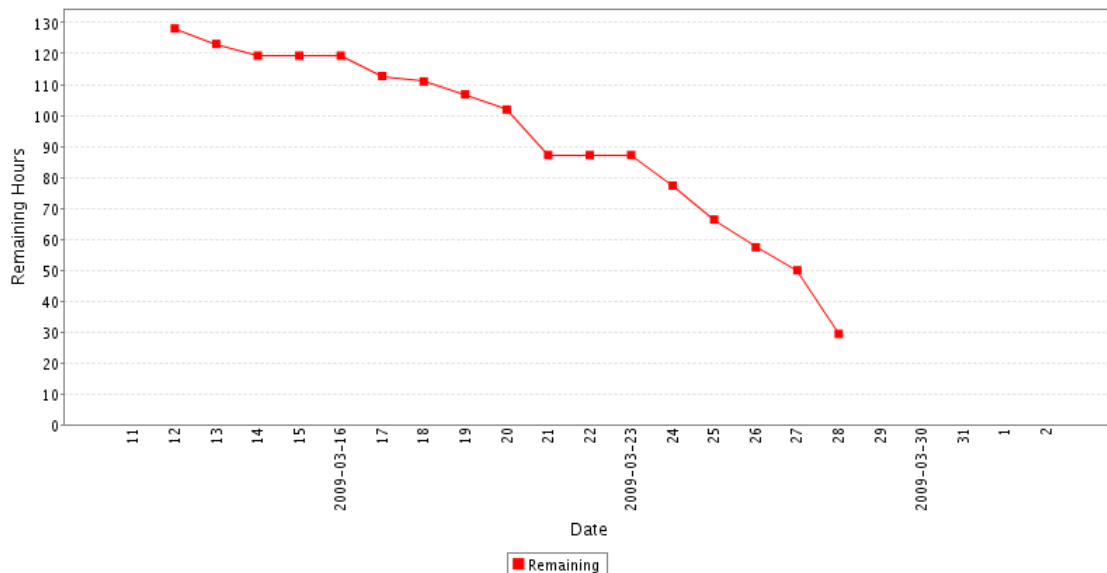


Figura 3.1 - Exemplo de Gráfico de Progresso (Gráfico de Burndown)

Além disso, Hartmann e Dymond (Hartmann & Dymond, 2006) afirmam que em nível de projeto, a medida de velocidade (*velocity*), utilizada em métodos ágeis, permite que a equipe possa estimar o trabalho a ser realizado com base em esforços anteriores. Segundo Hartmann e Dymond a velocidade significa quanto de software

(em pontos de história de usuário ou horas de desenvolvimento) a equipe pode entregar por iteração. Esta medida visa auxiliar o gerente de projeto, principalmente, no planejamento de iterações futuras e acompanhamento e controle do projeto.

Muitos clientes podem ser avessos à utilização de métodos ágeis, pois pensam que não serão informados formalmente sobre o andamento das atividades. Mas o cenário é totalmente o contrário. O gerente de projeto deve deixar claro ao cliente que seu papel deve ser atuante, participando e acompanhando o desenvolvimento o mais próximo possível.

### **3.3.4. Documentação**

Abordagens de desenvolvimento tradicional criam muita documentação. Esses registros servem como úteis artefatos de comunicação e rastreabilidade de desenho. Metodologia ágil, por outro lado, incentiva o pensamento leve e reduz as despesas gerais, em especial com a documentação. Grande parte do conhecimento em desenvolvimento ágil é tácita e reside na cabeça dos membros da equipe de desenvolvimento (Boehm, 2002). Isso pode tornar a organização dependente das equipes de desenvolvimento e podem transferir o equilíbrio de poder entre as equipes de gerência para o desenvolvimento. Tal situação pode não ser aceitável para muitas organizações. Este impasse pode ser resolvido determinando qual o conhecimento deve ser codificado e qual pode permanecer tácito (Nerur et. al., 2005).

Segundo Coram e Bohner (Coram & Bohner, 2005), métodos ágeis geralmente não aplicam extensa documentação, pois estas precisam ser continuamente atualizadas. O processo de “documentação-leve” evita esforço inútil, uma vez que os documentos são escritos e, em seguida, tornam-se obsoletos, pois não são atualizados de forma a refletir as alterações.

Com métodos ágeis, a informação é comunicada informalmente e é simplesmente mantida como parte do conhecimento coletivo da organização. Embora a redução do montante da documentação possa aumentar a produtividade, ela vem com alguns riscos

e custos. Documentação serve como uma forma de trazer novos membros até a produtividade e velocidade do restante da equipe. É útil quando o projeto está em transição para uma equipe de manutenção. De uma perspectiva comercial, documentos constituem a base para as auditorias garantindo boa qualidade nos procedimentos que são seguidos. Documentação serve como um repositório no domínio de conhecimento. Se a organização muda dramaticamente, este conhecimento pode ser perdido (Coram & Bohner, 2005).

O gerente de projeto deve manter o equilíbrio em relação à documentação, não documentando em excesso tão pouco de maneira insuficiente. O cliente e a equipe de desenvolvimento devem ajudar a definir qual o nível ideal de documentação do projeto.

### **3.3.5. Requisitos**

As diferenças entre como as abordagens tradicionais e ágeis executam as atividades de requisitos podem causar problemas e dificuldades para o gerente de projetos.

Highsmith e Cockburn (Highsmith & Cockburn, 2001) salientam que abordagens ágeis “são mais aplicáveis para ambientes turbulentos, com altas taxas de mudanças”. De acordo com a visão do mundo dos autores, as organizações são complexos sistemas adaptativos em que condições são emergentes, em vez de pré-especificadas. No entanto, Boehm (Boehm, 2002) afirma que embora fundamentos do manifesto ágil, como o segundo princípio – mudanças em requisitos são bem aceitas, mesmo tardias no desenvolvimento – oferecem um grande potencial de sucesso, os desenvolvedores podem fazer mau uso deles, quando incluem funcionalidades não solicitadas, por exemplo.

Segundo Boehm (Boehm, 2002) processos tradicionais funcionam melhor quando desenvolvedores podem determinar os requisitos com antecedência – inclusive através de protótipos e quando os requisitos permanecem relativamente estáveis. Nas situações e domínios em que são cada vez mais freqüentes os requisitos alterarem a

uma taxa muito elevada, a ênfase tradicional para ter requisitos completos, coerentes, precisos, testáveis e rastreáveis irá encontrar dificuldades e problemas para a atualização dos requisitos.

Por outro lado, Boehm e Turner (Boehm & Turner, 2005) afirmam que requisitos ágeis tendem a ser essencialmente informais e razoavelmente funcionais, o que pode dificultar nas abordagens de verificação e validação dos mesmos. Segundo Boehm e Turner, uma maneira de contornar este problema seria o fornecimento de informação adicional aos requisitos, principalmente aos requisitos não-funcionais, tais como confiabilidade e segurança.

Apesar de diferentes, a engenharia e gerência de requisitos nas duas abordagens devem caminhar para um senso comum. Os “tradicionalistas” não podem esperar somente requisitos definidos e estáveis para poderem desenvolver seus projetos e nem os “agilistas” devem desenvolver sem nenhum tipo de definição, pois correm o risco de não produzirem nada. Mais uma vez, o equilíbrio deve prevalecer.

### **3.3.6. Riscos**

A identificação, análise, monitoramento e respostas aos riscos do projeto são atividades comuns às duas abordagens de gerência de projetos.

No entanto, enquanto na abordagem tradicional é feito um plano formal de gerência de riscos, com indicações de planejamento e análise quantitativa e qualitativa dos riscos, na abordagem ágil a identificação, análise, monitoramento e respostas aos eventos e sinais de risco ocorrem de maneira informal e são realizados de forma contínua nas discussões de planejamento das iterações.

Segundo Marçal (Marçal, 2008), em Scrum, por exemplo, os riscos são identificados nas reuniões diárias e registrados em quadro branco ou listas de obstáculos, porém, informação como categoria, fonte e parâmetros de análise do risco não costuma ser identificada. Ainda segundo Marçal, o monitoramento e controle dos



riscos são realizados de maneira informal pelo gerente do projeto. Além disso, Scrum não apresenta estratégias de mitigação ou plano de respostas para os riscos.

Boehm (Boehm, 2002) afirma que a análise de risco das características do projeto versus determinadas características básicas dos métodos ágeis ou processos tradicionais podem ajudar a determinar o melhor equilíbrio de ambas as abordagens e contribuir para o sucesso do projeto.

### **3.3.7. Características do Projeto**

Para Coram e Bohner (Coram & Bohner, 2005), os métodos ágeis são mais aplicáveis aos projetos em que requisitos estão mal definidos e flexíveis uma vez que visam acomodar facilmente alterações. Projetos que são sem precedentes dentro de uma organização ou usam a vanguarda tecnológica (ou são eles próprios a vanguarda tecnológica), são exemplos de projetos em que a mudança é susceptível de ter um impacto significativo sobre o projeto. Em contrapartida, abordagens tradicionais são mais utilizadas em projetos cujo escopo é facilmente definido, detalhado e aprovado pelo cliente.

Projetos de longa execução também são um desafio, uma vez que tendem a ser maiores na natureza, com um elevado número de funcionalidades e capacidades. Isto pode gerar dificuldades de priorizar o trabalho. Um único representante cliente pode não ser suficiente e o gerente do projeto pode ser obrigado a tomar decisões sobre as prioridades.

Outro aspecto importante de projetos de longa vida é que eles tendem a ter longa vida de manutenção também. Manutenção pode ser um problema para desenvolvimento ágil uma vez que a quantidade de documentação que pode ser usada por desenvolvedores de manutenção é geralmente muito pequena. É também provável que os desenvolvedores originais tenham trocado de projeto sem nem sequer lembrar as decisões que foram feitas informalmente anos antes. Apoiar um produto que se espera que seja em serviço durante uma série de anos provavelmente irá exigir um grau

de documentação significativamente além daquele empregado por métodos ágeis (Coram & Bohner, 2005).

A segmentação no desenvolvimento de software deve ser constantemente observada. O gerente de projeto e a organização devem avaliar as características do projeto bem como os demais fatores informados neste trabalho a fim de se decidirem por determinada metodologia de desenvolvimento e gerência dos projetos. Cada abordagem, sendo ela tradicional ou ágil, possui o seu ambiente mais propício e produzirá os resultados esperados com mais facilidade.

### **3.4. Pessoas**

Segundo Boehm e Turner (Boehm & Turner, 2005), as questões de pessoas são as mais críticas na melhoria da gerência de engenharia e desenvolvimento pessoal.

Além das questões de capacidade individual, o trabalho em equipe e a predisposição em se utilizar ou não métodos diferentes de desenvolvimento e gerência podem se tornar um desafio para a organização. Por possuir conceitos diferentes em relação ao tratamento e responsabilidades das pessoas em um projeto, os métodos ágeis podem causar impactos.

Além da equipe de desenvolvimento (desenvolvedores e testadores), os clientes, a direção da organização, e claro, os gerentes de projetos são os mais afetados em relação à utilização de abordagens diferentes de gerência de projetos.

Além disso, Chin (Chin 2004) considera que os gerentes de projeto devem dedicar tempo considerável com todos os envolvidos quando o projeto está indo bem, de modo que, caso aconteça alguma crise no projeto, o gerente já tenha uma sólida relação de confiança, o que pode facilitar na tomada de decisões mais corretas.

### 3.4.1. Desenvolvedores

Segundo Coram e Bohner (Coram & Bohner, 2005), talvez o maior impacto dos métodos ágeis esteja nos desenvolvedores. Highsmith e Cockburn (Highsmith & Cockburn, 2001) afirmam que Métodos Ágeis dependem consideravelmente dos desenvolvedores – e estes devem ser “amigáveis, talentosos, qualificados e capazes de se comunicar bem”. Coram e Bohner afirmam ainda que os desenvolvedores devam estar dispostos a trabalhar como uma equipe, capaz de lidar com constantes mudanças, e com recursos (experiência) suficientes para resolver os problemas.

De acordo com Nerur e outros (Nerur et. al., 2005) para programadores “tradicionais”, acostumados a atividades solitárias ou que trabalham com grupos relativamente homogêneos de analistas e designers, as idéias de aprendizagem compartilhada, reuniões de reflexão, programação aos pares e processo decisório colaborativo podem representar dificuldades.

De acordo com Coram e Bohner, métodos ágeis não acomodam bem desenvolvedores iniciantes. No entanto, os trabalhadores qualificados em tecnologia são muitas vezes uma rara *commodity*. Este é um risco de gerência, além disso, alguns desenvolvedores podem não se encaixar neste ambiente ágil.

Isto não quer dizer que métodos ágeis exigem uniformemente pessoas de alta capacidade. Muitos projetos ágeis foram bem sucedidos com misturas de pessoas experientes e juniores, assim como em projetos tradicionais. A principal diferença é que métodos ágeis derivam sua agilidade baseando-se no conhecimento tácito incorporado na equipe, em vez da construção de conhecimento a partir de planos (Boehm, 2002).

Segundo Ceschi e outros (Ceschi et. al., 2005) tanto organizações ágeis quanto tradicionais têm respostas similares sobre quais as boas qualidades que os desenvolvedores devem ter. Em ambos os casos, eles destacam a capacidade de trabalhar em equipe e desenvolvedores com alta capacidade individual ou que estão constantemente motivados.

A análise da organização deve sobressair sobre os conhecimentos e habilidades dos desenvolvedores. De acordo com a literatura, uma equipe totalmente inexperiente em métodos ágeis tem maiores chances de fracasso em um projeto nesta metodologia. Conforme mencionado, o ideal seria uma mistura entre desenvolvedores experientes com outros inexperientes.

### **3.4.2. Testadores**

Coram e Bohner (Coram & Bohner, 2005) consideram que o impacto do uso de um método ágil nos testes (ou de garantia de qualidade) da organização depende dos ciclos de desenvolvimento curtos onde testes ocorrem durante todo o desenvolvimento. Testadores devem trabalhar em estreita colaboração com os desenvolvedores à medida que o código está sendo escrito.

Coram e Bohner afirmam ainda que testadores em ambiente ágil possam precisar ser mais capazes, tais como programadores, para automatizar os seus testes funcionais e de sistemas e incorporá-las ao *framework* de testes automatizados. Isso pode representar um conjunto de habilidades diferentes.

Outra mudança importante para os testadores é em relação ao foco que o teste recebe dentro do projeto. Segundo Cohn e Ford (Cohn & Ford, 2003), as atividades de teste recebem mais atenção em métodos ágeis, ao contrário das abordagens tradicionais, as quais consideram as atividades de teste como atividades pontuais.

Neste sentido, o desafio da gerência do projeto consiste em re-alocar os testadores que já não se inserem no grupo ágil e encontrar testadores com habilidades apropriadas de desenvolvimento e testes. Isso representa uma oportunidade para os desenvolvedores iniciantes, para começar e ganhar experiência no sistema e nos métodos ágeis.

### **3.4.3. Clientes**

Nerur e outros (Nerur et. al., 2005) apontam que o êxito do desenvolvimento ágil depende de encontrar clientes que irão participar ativamente no processo de desenvolvimento. Além disso, segundo Boehm (Boehm, 2002) espera-se que os clientes possam ser “colaborativos, representativos, com poderes, comprometidos e entendidos”. Contudo, não é uma tarefa fácil encontrar essas pessoas, especialmente para os sistemas complexos.

Clientes em métodos mais tradicionais podem estar envolvidos no lançamento do projeto – ajudando a definir os requisitos e obrigações contratuais – e no final do projeto com testes alfa, beta e de aceitação. Clientes em métodos ágeis estão envolvidos com maior frequência e com mais influência (Coram & Bohner, 2005).

Diversos autores ((Boehm, 2002), (Coram & Bohner, 2005) e (Highsmith & Cockburn, 2001)) afirmam ainda que métodos ágeis funcionam melhor quando os clientes operam em modo dedicado com a equipe de desenvolvimento (*on site*), e quando o seu conhecimento tácito é suficiente para toda a amplitude da aplicação. De acordo com Nerur e outros, em um ambiente ágil, a equipe de desenvolvimento, composta por programadores e o cliente, toma a maioria das decisões. Isso cria um ambiente pluralista de tomada de decisão devido a diversas formações, atitudes, metas e disposições cognitivas dos participantes.

Caso o gerente de projeto opte pelo uso de métodos ágeis é seu papel essencial analisar a real disponibilidade de representantes dos clientes.

### **3.4.4. Alta Direção**

Tal como acontece com a seleção de algum novo processo organizacional, o apoio da alta direção é essencial na adoção de uma abordagem ágil de gerência de projetos (Cockburn & Highsmith, 2001).

Segundo Coram e Bohner (Coram & Bohner, 2005) a alta direção quer o comprometimento com datas de entrega para funcionalidades específicas, o progresso da tarefa, bem como cronogramas e planos detalhados. Métodos ágeis representam uma importante mudança cultural para eles.

Além disso, a função de estimativa do custo de um projeto específico é difícil definir sob métodos ágeis. Visto que requisitos não são fixos, não há como saber a priori qual será o produto acabado e, por isso, quando a estimativa poderá ser finalizada. A alta direção vê-se confrontada com a incapacidade de garantir a entrega nas datas, custo e até mesmo a funcionalidade – uma situação que é considerada antiética para a maioria das abordagens de gerência. Esta pode ser uma situação insustentável para a gerência do projeto e geralmente é um dos fatores que contribuem para a não adoção de um método ágil na organização (Coram & Bohner, 2005).

A chave para um gerente de projeto é o de convencer os executivos que métodos ágeis possuem diversos benefícios. A escolha de um projeto apropriado é o primeiro passo para a conquista de confiança da alta direção em prol da adoção de métodos ágeis na organização.

### **3.4.5. Equipe**

Nerur e outros (Nerur et. al., 2005) afirmam que o desenvolvimento ágil apóia-se na equipe, em oposição à atribuição de papel individual que caracteriza o desenvolvimento tradicional. Coram e Bohner (Coram & Bohner, 2005) dizem que os métodos ágeis confiam substancialmente na colaboração e comunicação, sendo a equipe fundamental para o sucesso. A interação da equipe representa um risco significativo para um projeto ágil (Coram & Bohner, 2005).

A rotatividade é outro fator pessoal importante a ser considerado com uma equipe ágil. Sem documentação formal, a alta rotatividade em um projeto pode levar à perda de conhecimento crítico. Mesmo isto sendo atenuado por revisões de código e desenvolvedores alternando em diferentes áreas funcionais de trabalho, a perda de um

importante membro de uma equipe ainda pode ser catastrófica (Coram & Bohner, 2005). Já na abordagem tradicional, a documentação pode servir como auxílio no caso de uma alta rotatividade na equipe (Coram & Bohner, 2005).

Hirsch (Hirsch, 2005) aponta outra diferença: na abordagem ágil, a equipe conhece o estado atual do trabalho de cada membro, já na abordagem tradicional, quem costuma ter este conhecimento é somente o gerente do projeto.

O gerente do projeto deve considerar esse risco quando examinar se a equipe e a organização irão se adaptar melhor aos métodos ágeis ou processos tradicionais.

### **3.5. Papéis do Gerente de Projeto**

Segundo Highsmith e Cockburn (Highsmith & Cockburn, 2001) um gerente de projetos deve estar disposto a permitir que os membros da equipe possam tomar iniciativa. Isso pode representar uma mudança cultural para aqueles que devem estar dispostos a partilhar a autoridade na tomada de decisões.

Em ambientes ágeis, gerentes de projetos são responsáveis pelo acompanhamento de progresso e tomada de decisões empresariais. Além disso, a ênfase é colocada em resposta à mudança, em vez de se seguir um plano específico (Coram & Bohner, 2005).

Conforme mencionado, métodos ágeis pregam uma mudança de gerência de “comando-e-controle” para “liderança-e-colaboração”. A forma organizacional que facilita essa mudança precisa exatamente da mistura de autonomia e cooperação para alcançar as vantagens da sinergia ao mesmo tempo em que fornece flexibilidade e capacidade de resposta. Os papéis do gerente de projetos tradicional de planejador e controlador são confrontados com o papel de facilitador que dirige e coordena os esforços de colaboração de todos os envolvidos no desenvolvimento, garantindo assim que as idéias criativas de todos os participantes estejam refletidas na decisão final. No

entanto, o maior desafio aqui é se o gerente de projeto vai abrir mão exatamente da autoridade que gozava anteriormente (Nerur et. al., 2005).

Além disso, os gerentes de projeto têm um papel mais envolvido em um ambiente ágil. Em Scrum, por exemplo, o gerente de projeto encontra-se com a equipe diariamente e lidera a reunião diária Scrum (Highsmith & Cockburn, 2001). Gerentes de Projeto também estão mais envolvidos com a colaboração dos clientes, em vez do habitual foco na definição de entregas e contratos (Coram & Bohner, 2005).

Coram e Bohner (Coram & Bohner, 2005) afirmam que a principal responsabilidade do gerente de projeto é o de assegurar um ambiente altamente colaborativo. Segundo Chin (Chin, 2004), o gerente de projeto deve se transformar naquele que ouve novas idéias e as avalia para segunda opinião. Quando isso acontece, a equipe irá discutir os obstáculos técnicos com o gerente de projetos em vez de ocultá-las àquele que “somente dá tarefas”.

Esta mudança de perfil do gerente de projeto pode ser crítica. Conforme a literatura, alguns deles podem simplesmente se negar a ceder (compartilhar) sua autoridade à equipe do projeto. Cabe a organização conscientizar e envolver o gerente de projeto na gerência mais colaborativa do que controladora.

Segundo Chin (Chin, 2004), caso o Gerente de Projetos não seja um técnico especializado, será importante seu papel de observador externo, visando detectar informação valiosa que possa afetar o projeto. Em um ambiente ágil, o gerente de projetos raramente recebe a autoridade formal sobre a equipe. Ele deve estabelecer a sua própria credibilidade para conduzir a equipe com sucesso. E uma forma do gerente de projeto conseguir tal credibilidade e confiança dos membros da equipe é trazer informação valiosa de fora do projeto a fim de complementar com a informação já existente no ambiente do projeto.

Para Chin, o papel de um gerente de projetos combina muitas funções e competências:



- Reconhece que projetos ágeis irão mudar de direção muitas vezes através do decurso do projeto;
- É mais um facilitador do que um gerente;
- Atua como um coletor inteligente de informação;
- Trabalha proativamente para construir relacionamentos com as partes interessadas durante o tempo de sossego do projeto;
- Mantém um ponto de vista de “ampla visão” para a equipe do projeto;
- Combina sólidas competências inter-pessoais com uma abordagem gentil.

Por outro lado, segundo Anderson e outros (Anderson et. al., 2003) a presença do gerente de projetos em um projeto ágil é questionável. O trabalho apresenta um debate entre diversos especialistas das áreas de gerência de projetos e métodos ágeis. De acordo com os autores, um dos mantras dos métodos ágeis é a equipe auto-dirigida. Auto-direção significa que o gerente da equipe é relegado para um papel de facilitador com pouca ou nenhuma influência sobre o dia-a-dia das atividades. De um lado, Kent Beck (Anderson et. al., 2003) aponta que o gerente de um projeto XP pode fazer quatro coisas: pedir estimativas em termos de custos e resultados, mover pessoas entre os projetos, pedir relatório de progresso e cancelar o projeto. Já Glen Alleman indica que num contexto de auto-direção a equipe pode operar dentro do seu domínio, enquanto que as forças externas que afetam o seu êxito se tornam o domínio do gerente. Por outro lado, Joe Blotner afirma que as melhores práticas de gerência, quando aplicadas adequadamente e com perspicácia, não limitam, mas sim aumentam a produtividade e satisfação profissional de cada um dos membros da equipe, bem como melhora o produto final no processo. Um bom gerente pode e deve ajudar a equipe a ser mais produtiva, ter um melhor conhecimento a fim de se encaixar na organização como um todo e desenvolver os membros da equipe por ser ativo e envolvido com a equipe e o resto da organização (Anderson et. al., 2003).

Apesar de alguns autores afirmarem que o gerente de projetos em um ambiente ágil possa ser um mero coletor de estimativas e facilitador do trabalho dos

desenvolvedores, conforme afirmado por Blotner e outros autores, o gerente de projeto pode e deve assumir diversos papéis no projeto: líder, colaborador, facilitador, coletor e difusor de informação, elo entre o cliente e a equipe de desenvolvimento, tomador de decisões junto com a equipe, o cliente e a organização, entre outros.

### **3.6. Considerações Finais**

As abordagens tradicionais de gerência de projeto estão consolidadas no mercado e os resultados de sua aplicação são evidentes nos diversos ambientes de projeto. Já os métodos ágeis apresentam-se como uma opção às organizações que buscam alternativas de melhoria em seu processo de desenvolvimento, embora a literatura e as evidências empíricas dos seus resultados sejam reduzidas se comparadas à abordagem tradicional (Nerur et. al., 2005).

No entanto, as organizações podem optar pela utilização de práticas e características de ambas as abordagens. As características da organização, dos projetos e das pessoas, conforme destacado neste capítulo, tem papel essencial sobre as decisões do gerente de projeto sobre qual a melhor abordagem de gerência de projeto a ser utilizada. Em resumo, projetos que buscam a inovação e possuem natureza mais dinâmica dificultam o uso de gerência tradicional. Por outro lado, planejamento estratégico e projetos críticos que envolvam risco de vida não serão bem aceitos com gerência ágil.

Neste sentido, a Tabela 3.1 apresenta um quadro comparativo entre as duas abordagens. O objetivo é apresentar, de forma resumida, as características das organizações, projetos e pessoas e os papéis do gerente de projeto em cada ambiente.

**Tabela 3.1 - Comparação entre abordagem ágil e tradicional de gestão de projetos**

Organizações		
	ÁGIL	TRADICIONAL
Fatores de Negócios	Relação entre organização e empresa cliente é colaborativa.	Obrigação contratual, especificação de datas e funcionalidades bem definidas.
Processos	Práticas e valores toleráveis à mudança; Baseados em funcionalidades usando desenvolvimento evolutivo e iterativo.	Processos dirigidos à conformidade; Baseados em atividades e medições; Baseados em um modelo de ciclo de vida.
Tecnologia	Preferência por tecnologia Orientada a Objetos; Utilização de ferramentas específicas para automatização de testes e refatoração.	Sem restrições.
Tamanho e Distribuição	Preferência por organizações menores. Preferência por equipes não distribuídas.	Sem restrição de tamanho, porém indicada para organizações maiores. Sem restrição quanto à distribuição da equipe.
Projetos		
	ÁGIL	TRADICIONAL
Foco Principal	Execução.	Planejamento.
Objetivo Primordial	Satisfação do cliente através do fornecimento mais cedo e contínuo de software de valor; Rápido Valor.	Alta garantia, previsibilidade, repetibilidade e otimização;  Alta segurança.
Processo de Desenvolvimento	Refatoração; Revisões de código; Integração contínua; “Testes Primeiro” e constantes; Incorporação de novos requisitos é bem aceita; Procedimento de mudança é dinâmico e com rapidez na incorporação nas iterações.	Testes após a implementação; Incorporação de novos requisitos pode ser lenta e cara; Processo de mudança formal com identificação e aprovação entre os envolvidos.
Planejamento e Controle	Menor ênfase no planejamento formal; Planejamento mínimo; Planejamento iterativo; Confiança do cliente em relação ao	Planejamento formal;  Planejamento detalhado; Planejamento de todo o projeto; Contratos, marcos e medições de

	trabalho da equipe; Envolvidos no planejamento tem participação na elaboração.	progresso; Envolvidos no planejamento têm papel apenas de validação.
Documentação	Documentação mínima; Utilizada quando é extremamente necessária; Conhecimento tácito.	Documentação abrangente; Utilizada para comunicação e rastreabilidade; Conhecimento explícito.
Requisitos	Alta variabilidade; Ênfase em requisitos que atendam as necessidades dos usuários / clientes.	Variabilidade limitada; Ênfase em definir requisitos completos, coerentes, precisos, testáveis e rastreáveis.
Riscos	Identificados, monitorados e controlados durante as reuniões de iteração, de entrega e de revisão do projeto.	Construção de um plano formal de gerência de riscos.
Características do Projeto	Aplicáveis a projetos cujos requisitos estejam mal definidos e flexíveis;  Projetos que utilizam vanguarda tecnológica; Preferência por projetos pequenos e curtos; A arquitetura do projeto é desenhada para requisitos atuais.	Aplicáveis a projetos cujos requisitos estejam, preferencialmente, definidos e “congelados”; Projetos tradicionais;  Mais utilizada em projetos maiores e de longa duração; A arquitetura do projeto é desenhada para requisitos atuais e previstos.
<b>Pessoas</b>		
	<b>ÁGIL</b>	<b>TRADICIONAL</b>
Desenvolvedores	Agilidade; Podem assumir vários papéis;  Alta rotatividade nas equipes de desenvolvedores; Conhecimento tácito sobre o domínio do projeto; Preferência por desenvolvedores mais capacitados.	Orientados pelo plano; Geralmente assumem um único papel dentro do projeto; Baixa rotatividade dentro da equipe;  Conhecimento formal e documentado;  Sem preferência.
Testadores	Testadores trabalham com estreita relação com os desenvolvedores; Necessidade de conhecimento em programação, a fim de automatizarem os testes; Abordagem do Teste primeiro.	Geralmente os testadores trabalham separados dos desenvolvedores; Conhecimento restrito a área de testes;  Testes alfa, beta e de homologação.

Clientes	Clientes <i>on site</i> ; Clientes participam de todo o desenvolvimento; Cliente contribui na tomada de decisões.	Clientes não muito presentes; Clientes geralmente não participam de todo o desenvolvimento Cliente nem sempre contribui nas decisões tomadas.
Direção Executiva	Confia no trabalho do gerente de projeto e na equipe;  Pouco apegada às estimativas.	Comprometida com datas de entrega, cronogramas e planos detalhados;  Apegada às estimativas.
Equipe	Projeto apoiado na equipe;  Atuação colaborativa em todas as atividades do projeto; Alta rotatividade dentro da equipe; Equipe conhece o estado do trabalho de cada membro; Cliente faz parte da equipe.	Projeto apoiado no gerente de projeto;  Atuação da equipe com papéis claros e bem definidos; Baixa rotatividade dentro da equipe; Equipe não conhece o estado do trabalho de cada membro; Cliente não sabe quem é a equipe.
<b>Gerente de Projeto</b>		
	<b>ÁGIL</b>	<b>TRADICIONAL</b>
Papéis do Gerente de Projeto	Tendência para: Lidera e Colabora; Facilitação; Habilidades de influência; Compartilha a tomada de decisões no projeto; Contato direto e freqüente com o cliente; Foco no ambiente interno e externo;  Geralmente preocupado em coletar e canalizar informação;	Tendência para: Comando e Controle; Planejamento; Autoridade formal; Normalmente centraliza a tomada de decisões no projeto; Foco na definição de entregas e contratos com o cliente; Normalmente foca o ambiente interno; Geralmente preocupado em planejar e controlar;



## Capítulo 4

# Personalização de uma Ferramenta para Gerência de Projetos de Software

Este capítulo apresenta aspectos conceituais da XPlanner, dando destaque para suas características e funcionalidades. É apresentado também como ocorre o planejamento, acompanhamento e controle de um projeto na XPlanner. Por fim são descritas as funcionalidades personalizadas para a XPlannerPlus.

### **4.1. Contextualização**

Boehm e Turner (Boehm & Turner, 2003) afirmam que o ambiente no qual o software é imaginado, especificado e criado está mudando. Sistemas de software têm se tornado maiores e mais complexos e o ritmo das mudanças nos requisitos está se acelerando. Software está onipresente, tornando sua qualidade e usabilidade aspectos críticos.

Neste novo cenário de desenvolvimento de software, Boehm e Turner afirmam que o resultado de forte disciplina sem agilidade é sinônimo de burocracia e estagnação; e que agilidade sem disciplina significa entusiasmo sem responsabilidade.

Diversos autores criticam a rigidez dos processos nos desenvolvimentos tradicionais e afirmam que o conjunto de regras, práticas e valores dos métodos ágeis é

uma nova e revolucionária abordagem para resolução dos principais problemas da criação de software, contudo, a crença de que é necessário um processo (mesmo que simplificado) para o desenvolvimento de software não desapareceu.

Segundo Sommerville (Sommerville, 2007), independente da abordagem de desenvolvimento de software, algumas atividades são comuns a todas elas:

- **Especificação do software:** definição das funcionalidades (requisitos ou histórias de usuário) e das restrições do software;
- **Desenho e Implementação:** envolve a produção do software de acordo com as especificações;
- **Validação do Software:** as funcionalidades do software são validadas (ou homologadas / aceitas);
- **Evolução do Desenvolvimento:** o software evolui para continuar atendendo ao cliente;
- **Gerência do Desenvolvimento:** o desenvolvimento é planejado e acompanhado;

O que muda de uma abordagem para outra, de um método ou de um processo para outro é a quantidade de detalhamento e complexidade associados à execução das atividades.

Neste contexto, a ferramenta de gerência de projetos pode ser considerada a base para a aplicação do processo de desenvolvimento de software utilizado pela organização. Cabe à ferramenta, em maior ou menor grau, disponibilizar e direcionar a execução das atividades que envolvem o desenvolvimento do software.

Conforme mencionado na seção 2.7, uma ferramenta de gerência de projetos é uma ferramenta que apóia o planejamento, acompanhamento e controle de um projeto. Em uma base comum, o gerente do projeto e a equipe de desenvolvimento podem visualizar o progresso do projeto através de gráficos e medidas coletadas pela ferramenta.



E é justamente de acordo com a seqüência de utilização da ferramenta de gerência de projetos é que o processo de desenvolvimento a ser utilizado está implícito. Neste sentido, as organizações buscam selecionar e adaptar as ferramentas de acordo com seu processo de desenvolvimento. Por exemplo, uma ferramenta que exige um planejamento detalhado será mais adequada em ambientes tradicionais, já uma ferramenta focada na definição direta das iterações, requisitos e tarefas será mais apropriada em ambientes ágeis.

Em função do estudo sobre as abordagens ágeis e tradicionais de gerência de projeto apresentado no capítulo 3, percebemos que ampliaríamos nossa formação e aprendizado se utilizássemos uma ferramenta de gerência de projetos que pudesse agregar funcionalidades ligadas às melhores práticas de ambas as abordagens, mas mantendo o foco no desenvolvimento “desburocratizado” dos ambientes ágeis.

Dentre as diversas ferramentas existentes, a XPlanner (XPlanner, 2008) foi a selecionada para ser a base de nossos estudos. A XPlanner permitiu também sua extensão para dar suporte a algumas características adicionais.

A XPlanner é uma ferramenta *Web*, de código livre, voltada para o desenvolvimento ágil. Trata-se de uma ferramenta simples, porém com funcionalidades importantes, especialmente para uso no desenvolvimento com o método ágil XP.

Além de ser uma ferramenta voltada para ambiente ágil de desenvolvimento e gerência de projetos, a ferramenta XPlanner foi selecionada devido aos seguintes fatores:

- É uma ferramenta de código livre, que conseqüentemente, permite sua alteração e adaptação de novas funcionalidades;
- É uma ferramenta que tem a capacidade para suportar a gerência de multi-projetos e equipes distribuídas;
- É uma ferramenta já utilizada por diversas organizações (XPlanner, 2008);

- Foi desenvolvida em tecnologia *Java* (Java, 2008) e utilizando *Hibernate* (Bauer & King, 2004), (*Hibernate*, 2008), *Struts* (Struts, 2008) e outros modernos *frameworks*;
- O sítio dos desenvolvedores disponibiliza diversos canais para o auxílio da personalização da ferramenta;
- Possui diversas funcionalidades necessárias à gerência da maioria dos projetos de software;
- Pode ser facilmente disponibilizada em diversas línguas, tais como: Inglês, Português, Espanhol, Italiano, Chinês, entre outras;
- Possibilidade de se conectar a diversos sistemas gerenciadores de banco de dados, dentre eles, MySQL (MySQL, 2008), PostgreSQL e SQLServer.

As seções a seguir apresentam e discutem aspectos conceituais relacionados às funcionalidades já existentes na ferramenta, bem como das funcionalidades que foram criadas ou personalizadas em função do estudo apresentado no capítulo 3.

## **4.2. Aspectos Conceituais**

### **4.2.1. Características e Funcionalidades da Ferramenta XPlanner**

Segundo o sítio dos desenvolvedores (XPlanner, 2008), a XPlanner é uma ferramenta de planejamento e monitoramento de projetos para equipes *eXtreme Programming* (XP). Contudo, eles afirmam que a ferramenta pode atender e facilitar o trabalho em outros métodos ágeis.

Algumas das principais características da ferramenta, de acordo com o sítio oficial são:

- Modelo simples de planejamento;

- Suporte para armazenamento e monitoramento de projetos, iterações, histórias de usuário e tarefas;
- Controle de histórias não concluídas;
- Envio de notificações a respeito de alocação de histórias e tarefas;
- Monitoramento *on line* e geração de folha de registro de horas (*time sheet*) individual e da equipe;
- Geração de medidas (velocidade da equipe, tempo individual, entre outras);
- Geração de gráficos de velocidade da iteração, de progresso (*burndown*), distribuição das tarefas, entre outras;
- Disponibilidade para acrescentar notas para as histórias e tarefas (com arquivos anexos de diversos formatos);
- Visão precisa das estimativas da iteração;
- Exibição da situação das histórias para desenvolvedores e clientes;
- Exportação dos dados do projeto e da iteração para os formatos XML, MPX (Microsoft Project) e PDF;
- Integração e suporte autenticado para múltiplos projetos com autorizações específicas;
- Suporte a diversas línguas como Inglês, Espanhol, Francês, Alemão, Italiano, Português Brasileiro, Russo, Chinês e Japonês. Este suporte é realizado via um arquivo de internacionalização que armazena todos os textos estáticos utilizados pela ferramenta.

A ferramenta XPlanner possui funcionalidades ligadas à gerência de escopo, gerência de tempo, acompanhamento do cronograma, produtividade da equipe, geração de medidas, dentre outras. Contudo, não possui funcionalidades relacionadas à gerência de riscos, gerência de custos, gerência de recursos humanos, dentre outras. Evidencia-se, com isto, o foco da ferramenta na simplicidade e agilidade na gerência dos projetos.

A gerência do escopo do projeto na XPlanner ocorre através do cadastro e acompanhamento das iterações, histórias do usuário e tarefas. O gerente de projeto deve alocar as histórias de usuário e tarefas à sua equipe, que no caso da XPlanner, seriam os usuários previamente cadastrados na ferramenta. Assim que um usuário ativa a ferramenta, tem acesso às atividades que deve desempenhar. Estas atividades ficam separadas em 3 grupos: tarefas em execução, tarefas planejadas e tarefas finalizadas, conforme pode ser visualizado na Figura 4.1.

## XPlanner

[Top](#) | [Back](#) Content:  Search | ID:  Find | [Me](#)

**Name:** Leonardo Aparecido Ciscon

**Contact Info:**

Email: [leonardo@swfactory.com.br](mailto:leonardo@swfactory.com.br)  
Phone:

**Tasks in progress:**

Iteration	Story	Task	Acceptor?	Remaining	Active?	Actions
2a Iteração	<a href="#">Definição do Sucesso do projeto através de Régua(s) de Sucesso</a>	<a href="#">Implementação - JSP</a>	Yes	0,0		

**Planned Tasks:**

Iteration	Story	Task	Acceptor?	Remaining	Active?	Actions
2a Iteração	<a href="#">Definição do Sucesso do projeto através de Régua(s) de Sucesso</a>	<a href="#">Teste Unitário e Integrado</a>	Yes	7,5		
2a Iteração	<a href="#">Alteração nos termos da ferramenta</a>	<a href="#">Análise</a>	Yes	2,0		
2a Iteração	<a href="#">Alteração nos termos da ferramenta</a>	<a href="#">Teste Unitário e Integrado</a>	Yes	1,5		
2a Iteração	<a href="#">Ajuda sobre Gerência Ágil e Tradicional de Projetos</a>	<a href="#">Preparação</a>	Yes	3,0		
2a Iteração	<a href="#">Ajuda sobre Gerência Ágil e Tradicional de Projetos</a>	<a href="#">Implementação</a>	Yes	3,0		
2a Iteração	<a href="#">Ajuda sobre Gerência Ágil e Tradicional de Projetos</a>	<a href="#">Revisão</a>	Yes	2,0		

**Closed Tasks:**

Iteration	Story	Task	Acceptor?	Remaining	Active?	Actions
2a Iteração	<a href="#">Geração de Relatório de Orçamento</a>	<a href="#">Análise</a>	Yes	0,0		
2a Iteração	<a href="#">Geração de Relatório de Orçamento</a>	<a href="#">Implementação - BD, Mapeamento e Modelo</a>	Yes	0,0		
2a Iteração	<a href="#">Geração de Relatório de Orçamento</a>	<a href="#">Implementação - Demais classes</a>	Yes	0,0		
2a Iteração	<a href="#">Geração de Relatório de Orçamento</a>	<a href="#">Implementação - JSP</a>	Yes	0,0		
2a Iteração	<a href="#">Geração de Relatório de Orçamento</a>	<a href="#">Teste Unitário e Integrado</a>	Yes	0,0		
2a Iteração	<a href="#">Definição do Sucesso do projeto através de Régua(s) de Sucesso</a>	<a href="#">Análise</a>	Yes	0,0		

No stories where you are the customer.

No stories where you are the tracker.

No future tasks.

[Edit](#) | [People](#) | [Timesheet](#) | [Export](#)

User: [leonardo](#) [Logout](#) Version 0.7b7 built 05/24/2008 (rev 1149)

Contact your [Administrator](#) for production help. Any enhancement or bug report should be created following [these instructions](#). Click [here](#) to get your system information.

Figura 4.1 - Tarefas por Usuário

Em relação à gerência de tempo e acompanhamento do cronograma, a XPlanner disponibiliza diversas funcionalidades. O gerente de projeto pode estimar o tempo necessário para a realização de cada história de usuário e cada tarefa, e os usuários devem preencher a folha de registro de horas diariamente, isto é, o tempo efetivamente

gasto na realização da tarefa. A fim de facilitar o acompanhamento do progresso do projeto, a XPlanner disponibiliza relatórios de conclusão das tarefas, gráficos de velocidade da equipe e medidas relacionadas ao esforço planejado, esforço real e esforço remanescente. Os relatórios representam a divisão do tempo já gasto sobre o tempo estimado pelo gerente de projeto para cada atividade.

Outras funcionalidades disponibilizadas pela XPlanner são:

- Inserção de observações nos requisitos. Estas observações podem ser inseridas pelo gerente de projeto, pelos desenvolvedores e até pelos clientes;
- Exibição de painel de avisos. Com esta funcionalidade a XPlanner atende a um dos conceitos centrais de métodos ágeis: facilidade de comunicação entre a equipe do projeto;
- Possibilidade de se trabalhar em vários projetos ao mesmo tempo;
- Importação de dados para o cadastro dos requisitos; entre outras.

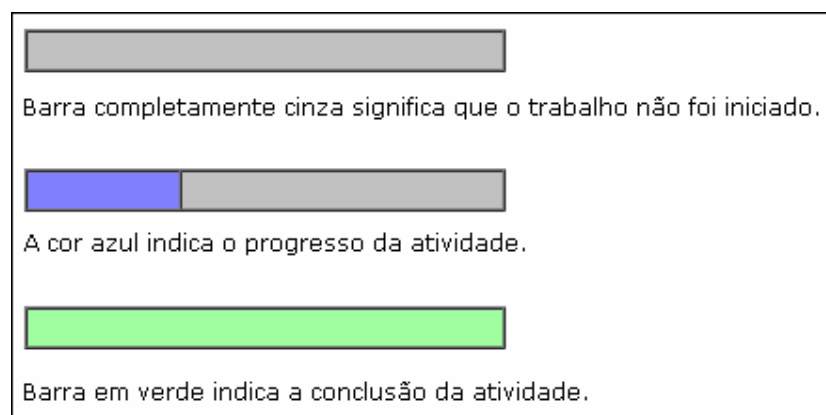
#### **4.2.2. Planejamento, Acompanhamento e Controle do Projeto na Ferramenta XPlanner**

O planejamento do projeto na ferramenta XPlanner inicia-se com a definição do plano da iteração pelos desenvolvedores e clientes. A ferramenta permite o cadastro das iterações e a atribuição das histórias de usuários. Uma história pode ser movida para outra iteração e sua prioridade pode ser definida pelo cliente (XPlanner, 2008).

Após esta etapa, o desenvolvedor deve estimar o esforço necessário para cada história e esta estimativa inicial pode determinar se é possível a implementação de um conjunto de histórias solicitadas pelo cliente. Após a definição das histórias, as tarefas devem ser cadastradas e estas são atribuídas a um desenvolvedor. À medida que o trabalho é realizado, o desenvolvedor deve informar o tempo efetivamente gasto com cada tarefa (XPlanner, 2008).

O acompanhamento do progresso do projeto pode ser feito em três níveis: iteração, história de usuário e tarefa através de barras de progresso. A intenção destas barras de progresso é prover uma rápida visão do progresso da equipe. As barras são normalizadas para reduzir “ruídos” visuais, porém, o tamanho relativo da história não é exibido (XPlanner, 2008).

A Figura 4.2 ilustra as barras de progresso da ferramenta XPlanner.



**Figura 4.2 - Barras de Progresso da Ferramenta XPlanner**

O acompanhamento da execução de um projeto utilizando as práticas ágeis do XP pode ser feito utilizando a “velocidade” do projeto (*velocity*). Por uma questão de tempo e escopo não investimos esforços no contraste de técnicas para acompanhamento de projetos ágeis e tradicionais, entretanto sabemos de alguns trabalhos onde é discutido o uso de técnicas de gestão de projetos tradicionais (p.ex. Valor Adquirido) no lugar de velocidade (Alleman & Henderson, 2003) e tomamos contato com esta medida através da XPlanner.

Além disso, a ferramenta disponibiliza uma série de recursos para facilitar o acompanhamento do progresso do projeto, tais como: listagem de todas as histórias cadastradas, relatório de medidas do projeto, gráficos de progresso e dados de acurácia do projeto.

### **4.3. Funcionalidades personalizadas e desenvolvidas**

A partir do estudo e utilização das funcionalidades existentes na ferramenta XPlanner, verificou-se a necessidade de personalizações e acréscimos de novas funcionalidades.

A elicitação de novas funcionalidades foi feita a partir de revisão da literatura e estudos de alguns métodos ágeis, tais como: XP, Scrum, Crystal, XPM (*Extreme Project Management*) e APM (*Agile Project Management*) e de trabalhos sobre ferramentas ágeis de gerência de projetos ((Liu, 2006), (Araujo, 2008), (Trapa & Rao, 2006), (Morgan & Maurer, 2006), entre outros). Além disso, a análise de outras ferramentas de gerência de projetos existentes, a experiência em gerência de projetos do autor do trabalho e finalmente o estudo sobre a abordagem ágil e tradicional de gerência de projetos, apresentado no capítulo 3, também serviram de base para a proposição das funcionalidades.

O objetivo principal é permitir que a ferramenta XPlanner seja utilizada em um maior número de organizações, e não somente àquelas que adotam métodos ágeis em seu desenvolvimento. Contudo, não é objetivo transformar a ferramenta em um novo Microsoft Project, por exemplo. As funcionalidades devem permanecer simples e ágeis em seu modo de execução. Além disso, a personalização foi realizada mantendo a língua inglesa nos termos utilizados nas funcionalidades. Optou-se pela língua inglesa a fim de manter a língua de origem da ferramenta, bem como, facilitar sua utilização para um número maior de organizações. Contudo, como a ferramenta utiliza um arquivo de internacionalização, é fácil a mudança para as outras línguas.

A partir da lista de funcionalidades, desenvolvemos um documento de requisitos, com o detalhamento dos requisitos funcionais e uma especificação de possíveis requisitos não-funcionais. Após priorização dos requisitos a construção foi dividida em duas iterações. Durante o desenvolvimento, buscamos a realização de atividades de controle de qualidade, como revisões e testes.

A primeira iniciativa adotada no projeto foi a utilização da própria ferramenta como instrumento de planejamento, acompanhamento e controle do projeto.

A Figura 4.3 ilustra primeira atividade realizada em relação à gerência do projeto: a divisão do projeto em duas iterações.

The screenshot shows the XPlanner web interface. At the top, there is a search bar with 'Content:' and 'ID:' labels, and buttons for 'Search' and 'Find'. Below this, the project name 'Project: PCX [id=562]' is displayed. A navigation breadcrumb shows '« PCX »'. The main content area features a yellow box with the project title 'Projeto de Customização da Ferramenta XPlanner'. Below this is a table with columns: Actions, ID, Iteration, Start Date, End Date, Days Wrk., and Stories. Two rows are visible, representing two iterations. Below the table, it says '2 items found, displaying all items. 1'. There is a 'Current Iteration' dropdown menu. At the bottom, there are links for 'Edit', 'Delete', 'Create Iteration', 'People', 'Export', 'History', 'Risks', 'Obstacles', and 'Print'. A 'Notes' section is also present with an 'Add Note/Attachment' link. The footer includes user information 'user: leonardo [Logout]' and version information 'Version 0.7b7 built 05/24/2006 (rev 1149)'. A small disclaimer is also visible: 'Contact your Administrator for production help. Any enhancement or bug report should be created following these instructions. Click here to get your system information.'

Actions	ID	Iteration	Start Date	End Date	Days Wrk.	Stories
	584	1a Iteração	2008-11-25	2008-12-05	0,0	4
	585	2a Iteração	2008-12-08	2008-12-19	0,0	4

**Figura 4.3 - Projeto de Personalização da XPlanner**

Os requisitos podem ser divididos em duas categorias:

- Requisito de personalização; e
- Requisito de construção.

As sub-seções a seguir apresentam informação a respeito dos requisitos personalizados e desenvolvidos para a ferramenta XPlanner, dando origem à XPlannerPlus, contudo, o logotipo original da ferramenta foi mantido.

Além do nome dos requisitos e detalhes sobre seu funcionamento, são apresentadas também justificativas sobre os mesmos. Para os requisitos considerados complexos, foram utilizadas imagens para facilitar o entendimento.

### 4.3.1. Requisito 1: Cadastrar Dados do Projeto

**Categoria:** Requisito de Personalização



**Descrição da personalização:** O cadastro do projeto consistia apenas dos seguintes campos:

- Nome do Projeto;
- Visível / Invisível;
- Elo de Navegação (*link*) para *Wiki* (Leuf & Cunningham, 2001) do Projeto.

Foram acrescentados os seguintes campos:

- Nome do Gerente do Projeto;
- Nome do Cliente do Projeto;
- Situação;
- Descrição do Projeto (Informação de visão geral do projeto e um resumo do escopo).

**Justificativa:** A informação incluída pode auxiliar o gerente de projetos e o responsável pela organização na gerência de múltiplos projetos na organização. Uma descrição do projeto, que inclui informação da visão geral do projeto e seu escopo macro, pode direcionar e contribuir para uma definição mais clara dos objetivos e metas do projeto. Já através do campo “Situação”, o gerente do projeto pode definir e acompanhar a situação atual do projeto: “Prospecção”, “Negociação”, “Desenvolvimento”, “Aceitação”, “Concluído” ou “Suspenso”.

**Imagem:**

**XPlanner**

[Top](#) | [Back](#)

**Create Project:**

**Name:**

**Manager's Name:**

**Customer's Name:**

**Status:**

**Description (Overview and Scope):** [Formatting Help](#)

Hidden  Escape < >

Remind developers to enter time (reminder will be sent via email)

**Project Wiki link:**

Figura 4.4 - Página de Cadastro do Projeto

### 4.3.2. Requisito 2: Cadastrar Recursos / Usuários

**Categoria:** Requisito de Personalização

**Descrição da personalização:** O cadastro dos recursos humanos do projeto consistia dos seguintes campos:

- Nome do Recurso / Usuário da Ferramenta;
- Login do Usuário;
- Iniciais do Nome do Recurso / Usuário;
- E-mail;
- Telefone;

- Senha.

Foram acrescentados os seguintes campos:

- Custo (Homem / Hora);
- Número de horas (semanal);
- Observação.

**Justificativa:** O cadastro dos recursos humanos do projeto estava relacionado aos dados de usuário para acesso à ferramenta. A informação de caráter gerencial incluída é utilizada para a geração do relatório de orçamento do projeto, além de contribuir para uma melhor alocação dos recursos dentro do projeto ou paralelamente em outros projetos da organização.

O campo de observação pode ser utilizado para o armazenamento de informação sobre o perfil profissional e pessoal de cada recurso do projeto.

**Imagem:**

**XPlanner**

[Top](#) | [Back](#)

**Create Profile:**

**Name:**

**User Id:**

**Initials:**

**Email:**

**Phone:**

**Cost Man-Hour**

**Number of Hours (Week)**

**Observation**

**Hide?:**

**Password**

**Confirm Password**

**System Administrator?**

Figura 4.5 - Página de Cadastro de Recurso / Usuário

### 4.3.3. Requisito 3: Cadastrar, Alterar, Remover e Listar Riscos do Projeto

**Categoria:** Requisito de Construção

**Descrição da personalização:** Cadastro e manutenção dos seguintes dados relacionados aos possíveis riscos do projeto:

- Descrição do Risco;
- Responsáveis;
- Informação Detalhada;
- Situação;
- Data de Identificação;

- Impacto;
- Probabilidade;
- Prioridade.

**Justificativa:** A ferramenta XPlanner não considerava nenhuma informação a respeito dos riscos que envolvem o projeto. Segundo Boehm (Boehm, 2002), a análise dos riscos do projeto pode ajudar no equilíbrio entre os métodos ágeis e tradicionais, o que pode direcionar o sucesso do projeto.

A partir da informação disponibilizada, o gerente de projeto pode armazenar, acompanhar e controlar de maneira simples os riscos que envolvem o projeto. A ferramenta permite a listagem ordenada dos riscos, o que facilita o tratamento do risco. Através do campo “Situação”, o gerente do projeto pode definir se o risco é “Novo”, se está “Em Revisão”, “Controlado”, “Eliminado” ou se o risco “Ocorreu”.

Além disso, o armazenamento desta informação formará uma base de dados históricos que será utilizada por outros projetos da organização.

**Imagem:**

**XPlanner**

[Top](#) | [Back](#)

Create Risk:

Description: [Formatting Help](#)

Responsibles: [Formatting Help](#)

Details: [Formatting Help](#)

Status: New

Date:

Impact (1 to 10):

Probability (1 to 10):

Priority (10 to 1):

Figura 4.6 - Página de Cadastro de Riscos

#### 4.3.4. Requisito 4: Cadastrar, Alterar, Remover e Listar Obstáculos (Impedimentos) do Projeto

**Categoria:** Requisito de Construção

**Descrição da personalização:** Cadastro e manutenção dos seguintes dados relacionados aos possíveis obstáculos (impedimentos) do projeto:

- Descrição do Obstáculo;
- Relator;
- Observações;
- Data de Identificação;

- Situação;
- Categoria.

**Justificativa:** A ferramenta XPlanner não fazia nenhum tipo de controle sobre os obstáculos (impedimentos) do projeto. Em Scrum (Schwaber, 2004), a lista de obstáculos é frequentemente acompanhada pelo gerente de projeto (o *Scrum Master*) a fim de resolver o mais rápido possível os obstáculos do projeto.

A inclusão desta funcionalidade permitirá ao gerente de projeto o cadastro e acompanhamento dos obstáculos, classificando-os em “Técnicos”, “Requisitos” e “Hardware” além da definição de sua situação: “Novo”, “Em Revisão”, “Eliminado”.

Assim como os riscos, os obstáculos armazenados para os projetos formarão uma base histórica para futuros projetos da organização.

**Imagem:**

The image shows a web application interface for XPlanner. At the top, the title 'XPlanner' is displayed in a large blue font. Below the title, there are two links: 'Top' and 'Back'. The main heading is 'Create Obstacle:'. The form contains three text input fields, each with a 'Formatting Help' link to its right. The fields are labeled 'Description:', 'Relator:', and 'Observation:'. Below these fields, there are three dropdown menus: 'Date' (with a calendar icon), 'Status' (set to 'New'), and 'Category' (set to 'Technic'). At the bottom of the form, there are two buttons: 'Create' and 'Reset'.

Figura 4.7 - Página de Cadastro de Obstáculos (Impedimentos)

#### **4.3.5. Requisito 5: Cadastrar Informação sobre Avaliação Final do Projeto (*Post-Mortem*)**

**Categoria:** Requisito de Construção

**Descrição da personalização:** Cadastros e manutenção dos seguintes dados relacionados à informação da avaliação final do projeto:

- Melhores Práticas;
- Piores Práticas;
- Lições Aprendidas;
- Futuro do Projeto.

**Justificativa:** O armazenamento das “Melhores Práticas”, “Piores Práticas” e “Lições Aprendidas” podem contribuir para a melhoria do processo de gerência de projetos da organização. Atividades que não geraram os resultados esperados, soluções aplicadas para tratamento dos riscos, resoluções de obstáculos técnicos, bem como definições a respeito dos projetos podem servir de guia para futuros projetos da organização.

Segundo o PMBOK, a informação histórica e demais informações sobre o projeto devem ser transferidas para a base de conhecimento de lições aprendidas para serem usadas por futuros projetos da organização.

Além disso, o “Futuro do Projeto” também é uma informação importante, visto que, através disso será possível afirmar se o projeto atingiu os objetivos e metas definidas pelas partes interessadas do projeto.

**Imagem:**



**XPlanner**

[Top](#) | [Back](#)

**Create Post-Mortem:**

**Best Practices:** [Formatting Help](#)

**Worst Practices:** [Formatting Help](#)

**Lessons Learned:** [Formatting Help](#)

**Future Of Project:** [Formatting Help](#)

Figura 4.8 - Página de Cadastro de Informação da Avaliação Final de Projeto

#### 4.3.6. Requisito 6: Gerar Orçamento

**Categoria:** Requisito de Construção

**Descrição da personalização:** Geração de um orçamento simplificado com os custos do projeto. O custo será calculado somente a partir do valor homem / hora do recurso multiplicado pelos seguintes esforços de cada recurso:

- Planejado;
- Trabalhado (real);
- Restante.

O total do custo do projeto calculado pela ferramenta é a soma de cada um destes custos dos recursos.

**Justificativa:** A ferramenta XPlanner não possui nenhuma funcionalidade relacionada ao acompanhamento de custos do projeto. O PMBOK, por exemplo,

afirma que conhecer e influenciar os fatores que criam as variações e controlar as mudanças no orçamento do projeto é uma das atividades que o gerente de projetos deve desempenhar.

Contudo, o orçamento do projeto é composto por diversas variáveis, e nesta personalização, apenas o custo da mão-de-obra foi incluído.

**Imagem:**

## XPlanner

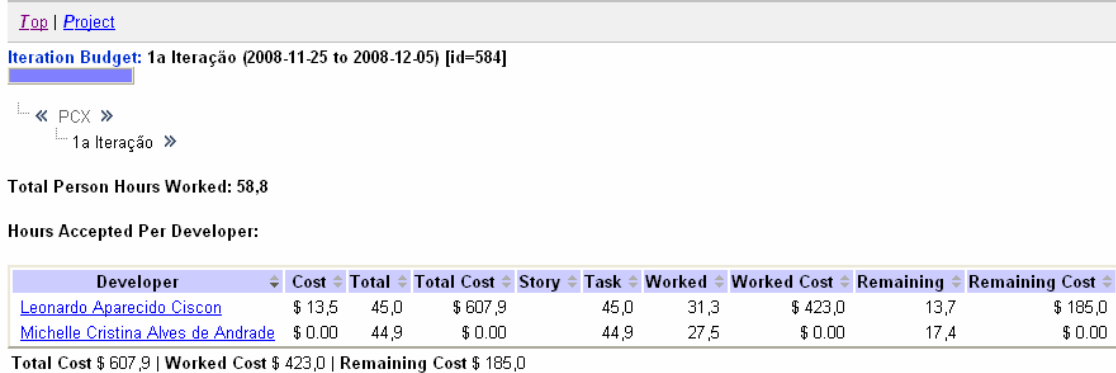


Figura 4.9 - Relatório de Orçamento

### 4.3.7. Requisito 7: Definir Sucesso do Projeto

**Categoria:** Requisito de Construção

**Descrição da personalização:** Definição do que seria o “sucesso” do projeto para cada um dos envolvidos: desenvolvedores, gerente do projeto, cliente, usuário, patrocinador, direção da organização, entre outros. Individualmente, cada usuário deve indicar quais itens são prioritários dentro do projeto:

- Satisfação das partes interessadas;
- Atendimento aos requisitos;
- Dentro do orçamento;
- Dentro do prazo;

- Valor agregado;
- Dentro do padrão de qualidade;
- Satisfação da equipe.

**Justificativa:** Um grande problema do gerente de projeto que deve ser minimizado logo no início e acompanhado durante todo o projeto é o alinhamento das expectativas do cliente, patrocinador e outras partes interessadas importantes do projeto. Thomsett (Thomsett, 2002) desenvolveu uma ferramenta para fazer este alinhamento, chamada “Régua de Sucesso”. São cinco opções na escala da régua: Desligado, 25%, 50%, 75% e Ligado. Obviamente deixar todos ligados não é realista, pois se for necessário decidir entre um critério ou outro, normalmente um se destaca mais.

Este será o indicador para o gerente de projetos de onde ele não pode falhar se tiver que privilegiar algum critério. Serão também as áreas mais bem acompanhadas nos relatórios e nas quais haverá mais controle.

**Imagem:**

# XPlanner


[Top](#) | [Back](#)

Create Success:

User: leonardo

[Update Sliders](#)

Off 25% 50% 75% On



Stakeholders's Satisfaction:

Customers's Requirements:

Within the budget:

Within the deadline:

Value added:

Within the Standard of quality:

Team's satisfaction:

Figura 4.10 - Definição do Sucesso do Projeto

### 4.3.8. Requisito 8: Ajuda sobre GP Ágil e Tradicional

**Categoria:** Requisito de Construção

**Descrição da personalização:** Construção de ajuda sobre gerência de projetos ágil e tradicional, focando o planejamento, monitoramento e controle de projetos na XPlannerPlus.

**Justificativa:** Conforme mencionado, um dos objetivos do desenvolvimento da XPlannerPlus é a sua utilização em organizações que não possuem a cultura de gerência de projetos. Neste cenário, uma contribuição seria a inclusão de conceitos e práticas de gerência de projetos ágil e tradicional. Esta ajuda não possui caráter final e definitivo sobre gerência de projetos, mas sim, servir de introdução e motivação para o uso de

gerência de projetos em todas as organizações. Além disso, cabe mencionar que o conteúdo desta ajuda é editável, de forma que possa permitir o acréscimo de nova informação e que esta possa ser compartilhada.

#### **4.4. Considerações Finais**

Além dos processos e das pessoas, as ferramentas desempenham papel fundamental no desenvolvimento de software. Muitas organizações priorizam o uso de ferramentas de análise e codificação e não dão o devido valor para as ferramentas de suporte à gerência de projetos.

A XPlannerPlus é mais uma alternativa que visa simplificar e agilizar a gerência de projetos, especialmente para as organizações ágeis e organizações que estão iniciando na cultura de gerência de projetos. Funcionalidades como cadastro de iterações e histórias de usuários / requisitos, geração de gráficos para acompanhamento do progresso do projeto, cadastro de riscos e obstáculos do projeto, dentre outras, podem facilitar a gerência do projeto e conseqüentemente promover o seu sucesso.

Além de sua utilidade intrínseca, a XPlannerPlus serviu para que nosso estudo tivesse uma dimensão prática que aperfeiçoou nossa visão dos diferentes aspectos de um projeto onde são utilizadas práticas dos métodos de desenvolvimento ágil de software.



## Capítulo 5

# Aspectos Operacionais e de Implementação

Este capítulo apresenta alguns aspectos de operação da ferramenta XPlanner detalhando como foi o processo de criação do ambiente de utilização e implementação, destacando as dificuldades encontradas e as soluções adotadas. Além disso, apresentamos informação sobre o cadastro dos dados básicos da ferramenta (projetos, iterações, histórias de usuário e tarefas) e descrevemos como foi a utilização da XPlanner para a gerência do projeto da XPlannerPlus.

São discutidos aspectos de implementação da XPlannerPlus, tais como arquitetura da XPlanner, tecnologias e ferramentas utilizadas na personalização das funcionalidades apresentadas no capítulo 4, informação sobre suporte para a personalização e novamente as dificuldades e soluções adotadas durante a implementação.

### **5.1. Aspectos Operacionais**

Após pesquisas em mecanismos de busca, o *download* do código fonte da ferramenta foi realizado no sítio *SourceForge.net* (<http://sourceforge.net>), acessado a partir do sítio dos desenvolvedores da ferramenta (<http://www.xplanner.org/>) (XPlanner, 2008). A versão disponibilizada no arquivo *xplanner-0.7b7b-war.zip* era a 0.7b7 (0.7 beta 7).

Após a descompactação do arquivo utilizando a ferramenta WinRAR, foi utilizada a ferramenta NetBeans 6.1 (NetBeans, 2008) para a criação do projeto de desenvolvimento da ferramenta. Não existe no sítio da ferramenta qualquer tipo de indicação sobre qual ferramenta de implementação deve-se utilizar para personalização da XPlanner. A escolha do NetBeans está relacionada ao conhecimento prévio do autor em relação a esta ferramenta.

Durante as tentativas iniciais de compilação foram identificadas bibliotecas ausentes no projeto. Após pesquisas em mecanismos de buscas, foi localizado o sítio *findJAR.com* (<http://www.findjar.com>). Com auxílio deste sítio, foi possível encontrar todas as bibliotecas necessárias.

Na primeira execução, a XPlanner acusou a ausência do banco de dados que seria utilizado pela ferramenta. De acordo com o manual de instalação (disponível no sítio dos desenvolvedores), o sistema de gerenciamento de banco de dados padrão da ferramenta é o MySQL (MySQL, 2008). Foi necessário a criação de um banco de dados denominado “xplanner”, cujo usuário deveria ser “xplanner” e senha “xp”. De fato, esta informação está configurada em um arquivo de propriedades da XPlanner denominado *xplanner.properties*. Para a criação do banco de dados foi utilizada a ferramenta MySQL Administrator.

Após a criação do banco de dados, a primeira execução da ferramenta detectou a ausência das tabelas, e de acordo com as configurações da XPlanner, foram criadas as tabelas com seus campos e relacionamentos.

Assim que a ferramenta foi executada com sucesso, o primeiro passo foi o cadastro de projetos e de pessoas. O cadastro do projeto exigia apenas o preenchimento do nome do projeto, de uma breve descrição e de um elo de navegação para algum *wiki* referente ao projeto. Já o cadastro de pessoas requeria somente o nome, identificador, e-mail, iniciais, telefone, senha e o papel do usuário nos projetos já cadastrados na ferramenta. A ferramenta disponibiliza os seguintes papéis:

- Nenhum;



- Visualizador;
- Editor;
- Administrador.

Depois de selecionar um dos projetos de teste cadastrados, foi necessária a inclusão de iterações, histórias de usuário e tarefas, necessariamente nesta ordem. O cadastro de iteração exige apenas o preenchimento do nome, data de início, data de fim e descrição da iteração. Já o cadastro de uma história de usuário necessita do preenchimento dos seguintes campos:

- Nome;
- Disposição (se a história foi planejada, transportada de outra iteração ou adicionada após o início da iteração);
- Usuário Cliente;
- Usuário Rastreador (responsável pelo acompanhamento do desenvolvimento da história);
- Situação (Rascunho, Definido, Estimado, Planejado, Implementado, Verificado e Aceito);
- Prioridade (de 1 a 10);
- Ordem (de 1 a 10);
- Esforço Estimado (em horas); e
- Descrição.

Por fim, a inclusão das tarefas para as histórias de usuário exige os seguintes campos:

- Nome;
- Tipo (Funcionalidade, Dívida, Defeito, Teste Funcional, Teste de Aceitação e Despesas Gerais);

- Disposição;
- Usuário Aceitador;
- Esforço Estimado (em horas); e
- Descrição.

A interface dos cadastros da ferramenta é muito simples, intuitiva e padronizada. Geralmente os campos são caixas de texto ou lista de opções (*combo*). Sempre depois que uma informação é cadastrada, a ferramenta retorna a sua exibição na listagem de informação do mesmo tipo, exibindo a informação recém cadastrada. A partir desta página, o usuário tem a opção de editar ou excluir alguma informação.

Após a realização dos cadastros básicos, o planejamento, acompanhamento e controle do projeto da XPlannerPlus ocorreram conforme descrito nas seções 4.2.2. A divisão dos requisitos em duas iterações foi a primeira atividade, conforme Figura 5.1 e Figura 5.2. Depois disso, ocorreu a definição e cadastro das tarefas para cada um dos requisitos.

**XPlanner**

Top | Project

Content:  Search | ID:  Find | Integrations | Me

Iteration 1a Iteração (2008-11-25 to 2008-12-05) [id=584]

« PCX »  
1a Iteração »

Cadastro, Listagem, Alteração e Remoção de Riscos, Impedimentos e Avaliação Final do Projeto. Alteração no cadastro de recursos do projeto.

Hours: Estimate 90,0, Actual 0,0, Remaining 90,0

Save order

Actions	ID	Order	Requirement / User Story	Cust.	Progress	Act.	Rem.	Cur. Est.	Orig. Est.	Tasks	Tracker	Disp.	Status
<a href="#">595</a>	1		Cadastro, Listagem, Alteração e Remoção de Riscos	1 <a href="#">LAC</a>	<div style="width: 100%;"></div>	0,0	30,0	30,0	30,0	5	<a href="#">LAC</a>	Planned	Draft
<a href="#">596</a>	2		Cadastro, Listagem, Alteração e Remoção de Impedimentos	1 <a href="#">LAC</a>	<div style="width: 100%;"></div>	0,0	25,0	25,0	0,0	0	<a href="#">MCAA</a>	Planned	Draft
<a href="#">597</a>	3		Cadastro, Listagem, Alteração e Remoção de Informações sobre Avaliação Final do Projeto	1 <a href="#">LAC</a>	<div style="width: 100%;"></div>	0,0	25,0	25,0	0,0	0	<a href="#">MCAA</a>	Planned	Draft
<a href="#">598</a>	4		Alteração no cadastro de recursos do projeto	2 <a href="#">LAC</a>	<div style="width: 100%;"></div>	0,0	10,0	10,0	0,0	0	<a href="#">LAC</a>	Planned	Draft

Save order

[Edit](#) | [Delete](#) | [Create Requirement / User Story](#) | [Start](#) | [Import](#) | [Export](#) | [Stories](#) | [All Tasks](#) | [Metrics](#) | [Charts](#) | [Accuracy](#) | [History](#) | [Print](#)

Notes: [Add Note/Attachment](#)

user: leonardo [Logout] Version 0.7b7 built 05/24/2006 (rev 1149)

Contact your Administrator for production help. Any enhancement or bug report should be created following [these instructions](#). Click [here](#) to get your system information.

Figura 5.1 - Projeto da XPlannerPlus: requisitos da 1a Iteração

The screenshot displays the XPlanner web application interface. At the top, there is a navigation bar with the XPlanner logo and a search bar. Below this, the current iteration is identified as "Iteration 2a Iteração (2008-12-08 to 2008-12-19) [id=585]". A breadcrumb trail shows the user's location: "PCX" > "2a Iteração". A summary box indicates the iteration's goal: "Geração de Relatório de Orçamento, Definição do Sucesso do projeto através de Réguas (sliders) de Sucesso, Alteração nos termos da ferramenta e Ajuda sobre Gerência Ágil e Tradicional de Projetos".

Key statistics are shown: "Hours: Estimate 75.0, Actual 0.0, Remaining 75.0". Below this is a table of requirements for the iteration. The table has columns for Actions, ID, Order, Requirement / User Story, Cust., Progress, Act., Rem., Cur. Est., Orig. Est., Tasks, Tracker, Disp., and Status. Four requirements are listed, each with a progress bar and associated metrics.

Actions	ID	Order	Requirement / User Story	Cust.	Progress	Act.	Rem.	Cur. Est.	Orig. Est.	Tasks	Tracker	Disp.	Status
	599	1	Geração de Relatório de Orçamento	1	LAC	0,0	25,0	25,0	0,0	0	LAC	Planned	Draft
	600	2	Definição do Sucesso do projeto através de Réguas (sliders) de Sucesso	2	LAC	0,0	30,0	30,0	0,0	0	LAC	Planned	Draft
	601	3	Alteração nos Dados de Cadastro do Projeto	3	LAC	0,0	10,0	10,0	0,0	0	MCAA	Planned	Draft
	602	4	Ajuda sobre Gerência Ágil e Tradicional de Projetos	4	LAC	0,0	10,0	10,0	0,0	0	LAC	Planned	Draft

At the bottom of the interface, there are links for "Edit", "Delete", "Create Requirement / User Story", "Start", "Import", "Export", "Stories", "All Tasks", "Metrics", "Charts", "Accuracy", "History", and "Print". A "Notes" section is also present with an "Add Note/Attachment" link. The footer includes the user name "user: leonardo (Logout)" and the version information "Version 0.7b7 built 05/24/2006 (rev:1149)".

Figura 5.2 - Projeto da XPlannerPlus: requisitos da 2a Iteração

A interface intuitiva da ferramenta facilitou o cadastro das tarefas, e por serem requisitos semelhantes, as tarefas foram quase sempre as mesmas: análise, implementação do BD (Banco de Dados), classes e interface e testes de unidade e de integração. No cadastro de cada tarefa, o usuário pode inserir sua estimativa de tempo para execução. Esta estimativa, comparada com o lançamento diário da realização das atividades por parte dos membros da equipe é utilizada pela ferramenta para permitir o monitoramento do progresso do projeto.

A Figura 5.3 exibe as tarefas do requisito “Cadastro, Listagem, Alteração e Remoção de Riscos”. Além disso, exibe também as estimativas de tempo originais, o tempo já gasto com cada tarefa e o tempo restante para cada tarefa. Neste caso, todas as tarefas já tinham sido completadas.

The screenshot displays the XPlanner interface for a requirement titled "Cadastro, Listagem, Alteração e Remoção de Riscos [id=595]". The requirement is implemented, with a priority of 1, estimated hours of 22.8, and actual hours of 22.8. The customer is Leonardo Aparecido Ciscon, and the tracker is also Leonardo Aparecido Ciscon. The last update was on 2008-12-05 at 18:47. The status is implemented.

Actions	ID	Task Name	Type	Progress	Acc.	Ori.	Est.	Act.	Rem.	Disp.	Type
	661	Análise	Feature	<div style="width: 100%;"></div>	LAC	3,0	3,0	2,0	0,0	Planned	Feature
	662	Implementação - BD, Mapeamento e Modelo	Feature	<div style="width: 100%;"></div>	LAC	3,0	3,0	2,3	0,0	Planned	Feature
	663	Implementação - Demais classes	Feature	<div style="width: 100%;"></div>	LAC	9,0	9,0	7,0	0,0	Planned	Feature
	664	Implementação - JSP	Feature	<div style="width: 100%;"></div>	LAC	7,5	7,5	6,5	0,0	Planned	Feature
	665	Teste de Unidade e de Integração	FTest	<div style="width: 100%;"></div>	LAC	7,5	7,5	5,0	0,0	Planned	FTest

At the bottom of the page, there are navigation links: [Edit](#) | [Delete](#) | [Move/Continue](#) | [Create Task](#) | [Export](#) | [History](#) | [Print](#). There is also a [Notes](#) section with an [Add Note/Attachment](#) link. The footer shows the user "leonardo [Lacocul]" and the version "Version: 0.7h7 built 05/24/2006 (rev: 1149)".

Figura 5.3 - Projeto da XPlannerPlus: tarefas do requisito Cadastro de Riscos

Durante o desenvolvimento do projeto, os riscos e obstáculos identificados foram cadastrados na ferramenta. A Figura 5.4 ilustra a página de cadastro de riscos da XPlannerPlus. No exemplo, o cadastro de um risco referente ao requisito: “Definir Sucesso do Projeto”.

The screenshot displays the XPlannerPlus interface for creating a risk. At the top, the title 'XPlanner' is shown in blue. Below it are links for 'Top' and 'Back', and a 'Create Risk:' button. The main form is divided into several sections:

- Description:** A text area containing 'Não localização / existência de componente JQuery para WEB.' with a 'Formatting Help' link.
- Responsibles:** A text area containing 'Leonardo' with a 'Formatting Help' link.
- Details:** A text area containing 'Caso não exista este componente, a interface para o requisito de sucesso de projeto será modificada.' with a 'Formatting Help' link.
- Status:** A dropdown menu set to 'New'.
- Date:** A date field set to '2008-12-11' with a calendar icon.
- Impact (1 to 10):** A text input field with the value '6'.
- Probability (1 to 10):** A text input field with the value '7'.
- Priority (10 to 1):** A text input field with the value '7'.

At the bottom of the form are 'Create' and 'Reset' buttons.

Figura 5.4 - Projeto da XPlannerPlus: identificação de um risco do projeto

Já o acompanhamento do projeto é viabilizado através das medidas, gráficos e dados de acurácia que a ferramenta disponibiliza. O gerente de projeto pode facilmente acompanhar:

- A quantidade de horas planejadas, trabalhadas e restantes agrupadas para cada desenvolvedor;
- Apresentação gráfica do progresso (horas trabalhadas) do projeto;
- Apresentação gráfica das horas restantes do projeto;
- Tarefas agrupadas por Tipo e Disposição;
- Dados de horas trabalhadas, restantes, super-estimadas e sub-estimadas;
- Comparação entre as horas estimadas e as horas trabalhadas;

Outra característica da ferramenta é que todas as ações realizadas (bem como usuário, data e hora) são armazenadas em um histórico, dividido em:

- Iterações;
- Histórias de Usuário; e
- Tarefas.

Além da facilidade e usabilidade na navegação da ferramenta, o desempenho na execução das ações durante todo o desenvolvimento do projeto foi outro ponto positivo. Todas as ações são executadas em frações de segundo, mesmo na geração de gráficos e de relatórios de acompanhamento do projeto.

Apesar de ter sido um projeto com uma equipe de apenas um integrante, o uso da ferramenta permitiu o planejamento, organização e acompanhamento dos requisitos a serem desenvolvidos. O lançamento das estimativas para cada tarefa a ser desempenhada permitiu uma planejamento em relação ao período destinado a implementação de cada requisito.

## 5.2. Aspectos de Implementação

Conforme já mencionado no capítulo 4, a ferramenta XPlanner foi desenvolvida na linguagem Java (Java, 2008) e é uma aplicação cliente / servidor. Ela atua como um servidor para agentes *Web* (navegadores *Web*, robôs de busca), atendendo, simultaneamente, várias requisições de vários clientes. A XPlanner também age como um cliente, necessitando de um servidor de banco de dados.

A ferramenta foi desenvolvida utilizando apenas tecnologias e projetos *open-source*, sendo os principais (XPlanner, 2008):

- Hibernate;
- MySQL;
- Struts;
- JasperReports;

- JFreeChart;
- JUnit; entre outros.

O código fonte da ferramenta está disponível no sítio dos desenvolvedores sob licença LGPL (*Lesser General Public Licence*). Segundo o sítio oficial da licença (GNU, 2009), o sistema pode ser personalizado, porém, permanecendo nesta licença, seu código fonte deve ser também disponibilizado.

### 5.2.1. Arquitetura do Sistema

A arquitetura da XPlanner corresponde a três camadas: apresentação, negócios e persistência. A camada de apresentação é constituída basicamente por páginas JSP (*Java Server Pages*) e páginas HTML (*HyperText Markup Language*). É através desta camada que o usuário interage com a ferramenta e visualiza os resultados de suas ações. Todos os textos estáticos das páginas são armazenados em um único arquivo denominado *ResourceBundle.properties*, a fim de facilitar a internacionalização.

Para a camada de negócios foi utilizado o *framework* Struts (Struts, 2008). A camada de negócios (as ações) manipula os objetos do domínio do planejamento e os provê à camada de apresentação. Além disso, possui todas as regras de negócios da ferramenta e realiza a ligação entre as demais camadas.

A comunicação entre a camada de negócios e de apresentação é feita de acordo com o modelo MVC (*Model, View e Controller*). O modelo corresponde às classes das entidades do modelo de negócios. A visão corresponde às interfaces gráficas apresentadas ao usuário. E a camada de controle corresponde às classes que fazem a mediação entre as visões (usuário) e o modelo (ferramenta). O controlador, na XPlanner é implementado com o *framework* Struts (Struts, 2008). Já as visões são páginas JSP (*Java Server Pages*), conforme mencionado.

A camada de persistência é responsável pela gravação e recuperação dos dados no banco de dados. A ferramenta pode se conectar a diversos sistemas gerenciadores de

banco de dados, contudo, a versão padrão utiliza o banco de dados livre MySQL (MySQL, 2008). Os dados não são acessados diretamente, mas sim carregados como objetos Java utilizando-se o mapeamento objeto relacional proporcionado pelo *framework* Hibernate (Bauer e King, 2004) e (Hibernate, 2008). A conexão com o banco de dados ocorre via JDBC (*Java DataBase Connectivity*) (JDBC, 2008).

A Figura 5.5 ilustra a arquitetura da ferramenta XPlanner.

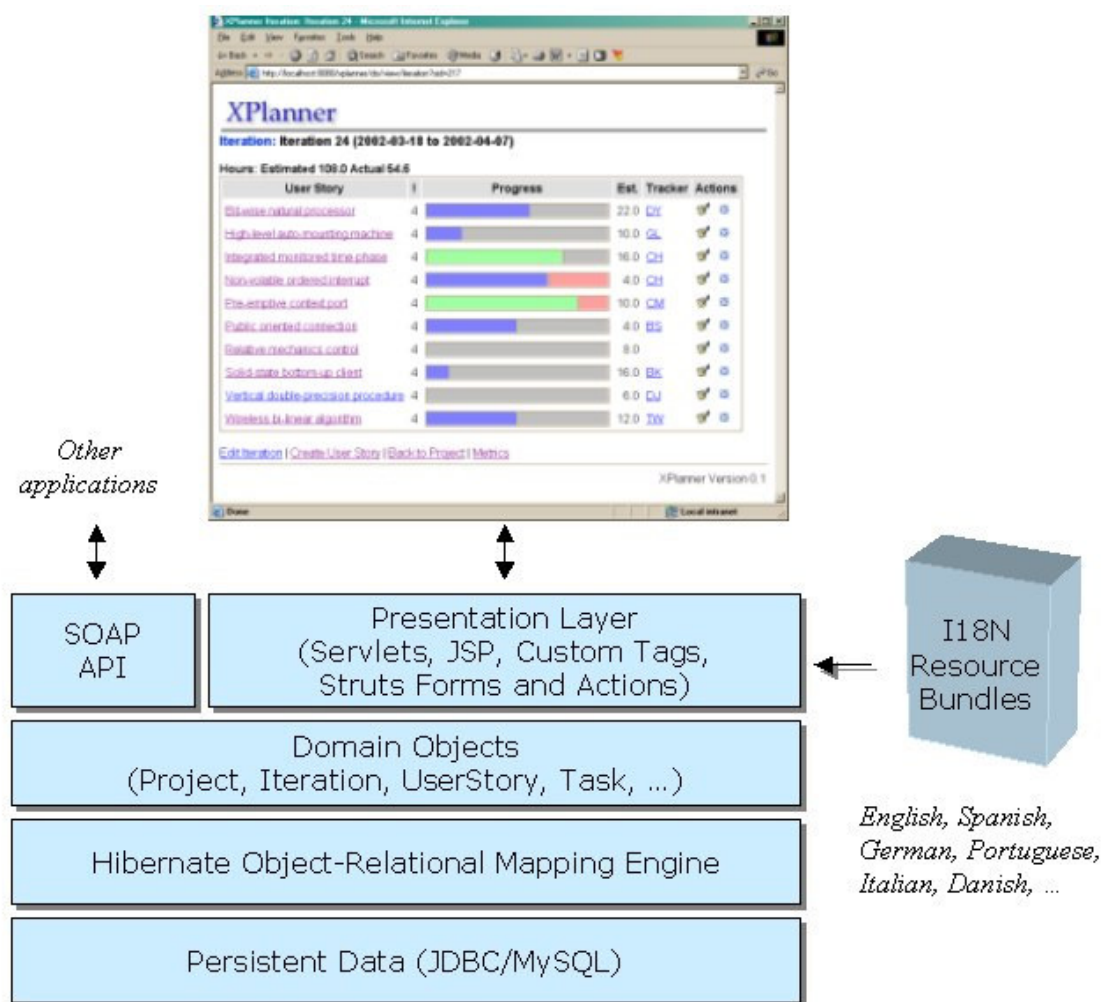


Figura 5.5 - Arquitetura da Ferramenta XPlanner. Fonte: (XPlanner, 2008)



### 5.2.2. Implementação

A ferramenta de implementação utilizada para a personalização da ferramenta foi o NetBeans 6.1 (NetBeans, 2008) e o Tomcat 6.0.16 (Tomcat, 2008) foi utilizado como servidor de aplicações *Web*. A escolha deste ambiente foi totalmente baseada no conhecimento prévio do autor sobre estas tecnologias. No sítio da ferramenta, não há indicações sobre qual o melhor ambiente para sua personalização.

Para acesso e manipulação do banco de dados, foram utilizadas as ferramentas MySQL Administrator 1.2.12 e MySQL Query Browser 1.2.12 ambas disponíveis no sítio do MySQL (MySQL, 2008).

O Internet Explorer 6.0 foi o *Web browser* utilizado para testes e validação do sistema.

Foi utilizado também a ferramenta TortoiseSVN 1.5.5 (TortoiseSVN, 2008) como ferramenta de gerência de configuração. Novamente, o conhecimento prévio determinou a escolha desta ferramenta.

Durante a personalização, a arquitetura e estrutura da ferramenta foi mantida com o objetivo de não aumentar a complexidade do esforço de adaptação da XPlanner.

O suporte ao desenvolvimento da ferramenta é muito diversificado. A partir do sítio dos desenvolvedores são disponibilizados os seguintes mecanismos:

- Lista de e-mails dos desenvolvedores;
- Fóruns de ajuda e de discussões abertas; e
- Ferramenta de registro de *bugs* e demais questões.

A fim de facilitar a explicação sobre detalhes da implementação, a Tabela 5.1 lista as funcionalidades personalizadas e uma abreviação para cada uma deles.

**Tabela 5.1 - Funcionalidades personalizadas na Ferramenta XPlanner**

Funcionalidade	Abreviação
----------------	------------

4.2.1. Requisito 1: cadastrar dados do projeto	R01
4.2.2. Requisito 2: cadastrar recursos	R02
4.2.3. Requisito 3: cadastrar, alterar, remover e listar riscos	R03
4.2.4. Requisito 4: cadastrar, alterar, remover e listar obstáculos (impedimentos) ao projeto	R04
4.2.5. Requisito 5: cadastrar informação sobre avaliação final do projeto ( <i>post-mortem</i> )	R05
4.2.6. Requisito 6: gerar orçamento	R06
4.2.7. Requisito 7: definir sucesso do projeto	R07
4.2.8. Requisito 8: ajuda sobre GP ágil e tradicional	R08

O primeiro requisito desenvolvido foi R01. Durante o desenvolvimento deste requisito, alguns aspectos básicos da arquitetura da XPlanner foram assimilados. Tais como:

- Arquivos de Configuração da Ferramenta (disponíveis nas pastas `xplanner\WEB-INF\` e `xplanner\WEB-INF\classes`);
- Arquivo de Mensagens Estáticas (`xplanner\WEB-INF\classes\ResourceBundle.properties`);
- Interfaces *Web*;
- Tabelas e relacionamentos no Banco de Dados; entre outras.

O segundo requisito, de complexidade um pouco maior, foi o R03. Como se tratava de uma nova entidade na ferramenta foi necessário a criação de uma tabela no banco de dados, bem como novos arquivos de:

- Mapeamento em arquivo XML da tabela “Risk” de acordo com o padrão Hibernate (localizado na pasta `xplanner\WEB-INF\classes\mappings`);

- Classe de domínio da entidade (localizado na pasta `xplanner\WEB-INF\classes\com\technoetic\xplanner\domain`);
- Classe de formulário da entidade (localizado na pasta `xplanner\WEB-INF\classes\com\technoetic\xplanner\forms`);
- Interface de consulta dos riscos (`xplanner\WEB-INF\jsp\view`);
- Interface de cadastro e alteração dos riscos (`xplanner\WEB-INF\jsp\edit`);

O desenvolvimento do requisito R02 foi muito similar ao do requisito R01 e os requisitos R04 e R05 seguiram os moldes do R03. Devido a isto, estes requisitos não representaram dificuldades de análise e implementação.

Já o requisito R06, apesar de apresentar apenas um orçamento simples, baseado no custo valor homem / hora, foi de complexidade média de implementação. Sua implementação esteve basicamente restrita ao arquivo de exibição do orçamento, contudo, exigiu conhecimento de aspectos diferentes de implementação.

Sem dúvidas, o requisito R07 foi o mais difícil de ser desenvolvido. Inclusive, a estimativa inicial (30 horas) foi ultrapassada em 87%. A primeira parte da funcionalidade envolveu aspectos similares aos requisitos R03, R04 e R05, isto é, armazenamento e recuperação de dados do banco de dados. Contudo, a interface de definição de sucesso do projeto é totalmente diferente dos padrões adotados pela ferramenta. O uso de um componente que representasse uma régua de sucesso criou complicações devido à integração deste com os padrões da ferramenta, bem como o aprendizado de sua utilização não foram considerados na estimativa inicial. Além disso, o componente utilizado pertence a uma biblioteca disponível na Internet, o que criou a obrigação do usuário estar conectado à Internet para o uso apropriado desta funcionalidade.

Por fim, o requisito R08 teve seu trabalho concentrado na análise e preparação da ajuda sobre gerência ágil e tradicional de projetos de software.

O número de linhas de código-fonte da XPlanner era de 46.532. Já a XPlannerPlus possui 51.063, o que corresponde a um acréscimo de 4.531 linhas implementadas neste trabalho.

### 5.3. Considerações Finais

O desenvolvimento da XPlannerPlus mostrou-se medianamente complexo. Apesar de conter informação de auxílio, o sítio da XPlanner não fornece informação detalhada sobre o seu processo de criação. Além disso, a necessidade de pesquisar e adicionar bibliotecas ausentes pode trazer um pouco de dificuldade para usuários leigos em desenvolvimento.

Conforme mencionado, a XPlanner é de fácil utilização. Seus cadastros e listagem são de fácil acesso e bom desempenho de execução. Contudo, alguns termos foram de difícil compreensão, dentre eles, “disposição”, “usuário aceitador” e “usuário rastreador”.

Em relação à implementação, um dos pontos de maior dificuldade foi em relação ao entendimento da arquitetura utilizada. Por exemplo, para se incluir uma nova entidade na ferramenta (como a descrita em R03), diversos arquivos precisam ser criados ou editados. No entanto, depois do conhecimento sobre a arquitetura, a implementação torna-se mais produtiva.

Por fim, cabe mencionar que a XPlanner suporta a implementação de testes unitários através do *framework* JUnit, contudo, devido a não ser o foco principal do trabalho, a execução dos testes unitários não foi realizada.

Esperamos que este capítulo possa servir de referência para futuras extensões da ferramenta XPlanner ou em outras ferramentas de código livre existentes.

# Capítulo 6

## Estudo de Caso

Este capítulo descreve o estudo de caso realizado neste trabalho. São descritas a metodologia e o ambiente onde foi realizado o estudo de caso. A gerência de três projetos de software é apresentada, bem como, os resultados e os aspectos observados durante a realização do estudo de caso.

### **6.1. Contextualização**

O capítulo 3 descreveu um estudo sobre a abordagem ágil e a abordagem tradicional de gerência de projetos de software. Um complemento para interpretar de forma operacional os diversos aspectos conceituais deste trabalho foi a realização de um estudo de caso com projetos reais de desenvolvimento de software. Este estudo de caso deu subsídios ao nosso estudo sobre as práticas de gerência de projetos e também sobre as demandas com relação a ferramentas de suporte ao projeto de desenvolvimento de software. Este estudo de caso também corresponde a um tipo de validação da XPlannerPlus pois, como será visto ela foi usada em diferentes situações.

Um estudo de caso é conduzido para investigar uma única entidade de um fenômeno com um espaço de tempo específico (Wohlin et. al., 2000). Uma vantagem dos estudos de caso é que eles são fáceis de serem planejados e uma desvantagem é que os resultados são difíceis de serem generalizados sendo difícil interpretá-los para situações que são diferentes do caso de exemplo (Wohlin et. al., 2000).

Alguns autores empregaram a mesma abordagem para validação de seus trabalhos. Liu (Liu, 2006) realizou um experimento que durou 4 meses e considerou 2 projetos independentes, porém executados em paralelo. A equipe era composta por engenheiros de software, gerente de projetos e o próprio autor como observador. Além disso, o autor realizou o estudo da utilização da ferramenta no ambiente dos projetos, bem como, conduziu entrevistas com os envolvidos.

Trapa e Rao (Trapa & Rao, 2006) validaram a ferramenta T3 de monitoramento de projetos ágeis através de um estudo de caso de um pequeno projeto com duas histórias de usuário para desenvolvimento. Já Ghanam e outros (Ghanam et. al., 2008) realizaram um estudo para avaliar a ferramenta desenvolvida em seu trabalho que consistia do acompanhamento de 14 voluntários avaliados isoladamente e um grupo de 5 membros que utilizaram a ferramenta em um ambiente real de projetos de desenvolvimento de software.

Morgan e Maurer (Morgan & Maurer, 2006) e Hanakawa e Okura (Hanakawa & Okura, 2004) também realizaram validações semelhantes.

Neste trabalho, o nosso estudo de caso é muito similar, principalmente aos trabalhos de Trapa e Rao (Trapa & Rao, 2006) e Ghanam e outros (Ghanam et. al., 2008), sendo o objetivo primordial desta validação a investigação sobre a eficiência e a utilização da XPlannerPlus em uma organização de desenvolvimento de software. Contudo, esta validação não tem caráter definitivo e novos estudos de caso devem ser realizados para atestar o uso da XPlannerPlus nos mais variados ambientes de desenvolvimento de software.

A seção 6.2 descreve a metodologia empregada, a seção 6.3 descreve o ambiente do estudo de caso e as seções 6.4 a 6.6 caracterizam os projetos e descrevem os resultados obtidos em cada um dos projetos.

## 6.2. Metodologia

A validação da XPlannerPlus foi dividida em duas fases: (i) utilização da XPlannerPlus em 3 projetos reais de uma organização de desenvolvimento de software, e (ii) realização de entrevistas com todos os envolvidos nos projetos da organização. Além disso, durante o estudo de caso, foi observado como as práticas de gerência de projetos se refletiam na XPlannerPlus e como foi a utilização de características ágeis no desenvolvimento dos projetos da organização.

A primeira fase da validação iniciou-se pela escolha das organizações que poderiam utilizar a XPlannerPlus como ferramenta de suporte à gerência de projetos. Idealmente, o critério para escolha das organizações além da disponibilidade para sondagens, era que a organização possuísse algum conhecimento sobre processos de desenvolvimento de software e tivesse a cultura de gerência de projetos. Além disso, deveria possuir projetos (ou iterações dos projetos) de curta duração (entre 1 a 3 meses) e que permitissem o acompanhamento da gerência dos projetos. Contudo, em função da percepção antecipada da dificuldade de conseguir locais adequados a esta caracterização investimos em uma organização ligada ao autor do trabalho. Os detalhes da organização são descritos na seção 6.3.

Antes da utilização da XPlannerPlus, um breve treinamento foi realizado com os participantes do estudo de caso. O foco foi em alguns conceitos de métodos ágeis e sobre a utilização da ferramenta, tanto na visão dos programadores e testadores, quanto na visão dos gerentes de projeto. Basicamente, o treinamento abordou os seguintes temas:

- Gerência de projetos:
  - como é a gerência ágil de projetos;
  - qual é o papel do gerente de projetos em um ambiente ágil; e
  - qual é o papel da equipe em um ambiente ágil.
- Utilização da XPlannerPlus:

- como acessar a ferramenta;
- como é o planejamento do projeto na ferramenta;
- como é o acompanhamento do projeto por parte do gerente de projetos;
- como é o lançamento do tempo gasto em cada atividade realizada pelos membros da equipe;
- como é o cadastro de riscos e obstáculos dos projetos;
- como é a definição do sucesso do projeto; e
- como é o cadastro da informação de avaliação final do projeto.

A fim de se evitar algum tipo de avaliação tendenciosa, os 3 projetos da organização escolhida foram gerenciados por 3 gerentes diferentes, sendo um deles o autor deste trabalho. Nos demais projetos, o autor foi apenas observador além de responder dúvidas que surgiram durante o estudo de caso.

Já em relação às entrevistas com os envolvidos não tivemos tempo suficiente para investir no planejamento e aplicação de questionários de avaliação. Em função disto, adotamos a mesma estratégia de Liu (Liu, 2006) e criamos nosso próprio questionário, a fim de investigar a relação de uso entre os participantes do estudo de caso e a XPlannerPlus.

As questões citadas a seguir foram enviadas por e-mail para todos os envolvidos e as respostas foram enviadas em sigilo. Depois disso, uma análise das respostas foi realizada e o resultado descrito na seção 6.7.

#### **Questionário:**

- 1) A funcionalidade **Cadastro de Riscos** da XPlannerPlus foi utilizada?
  - a. Sim
  - b. Não
  - c. Parcialmente
- 2) A funcionalidade **Cadastro de Riscos** da XPlannerPlus atendeu à sua expectativa?
  - a. Sim
  - b. Não
  - c. Parcialmente



- 3) A funcionalidade **Cadastro de Obstáculos** da XPlannerPlus foi utilizada?
  - a. Sim
  - b. Não
  - c. Parcialmente
- 4) A funcionalidade **Cadastro de Obstáculos** da XPlannerPlus atendeu à sua expectativa?
  - a. Sim
  - b. Não
  - c. Parcialmente
- 5) A funcionalidade **Cadastro da Informação de Avaliação Final de Projeto** da XPlannerPlus foi utilizada?
  - a. Sim
  - b. Não
  - c. Parcialmente
- 6) A funcionalidade **Cadastro da Informação de Avaliação Final de Projeto** da XPlannerPlus atendeu à sua expectativa?
  - a. Sim
  - b. Não
  - c. Parcialmente
- 7) A funcionalidade de **Definição do Sucesso do Projeto** da XPlannerPlus foi utilizada?
  - a. Sim
  - b. Não
  - c. Parcialmente
- 8) A funcionalidade de **Definição do Sucesso do Projeto** da XPlannerPlus atendeu à sua expectativa?
  - a. Sim
  - b. Não
  - c. Parcialmente
- 9) A usabilidade (desempenho, facilidade de uso, facilidade de aprendizagem, termos utilizados, etc.) da XPlannerPlus atendeu à sua expectativa?
  - a. Sim
  - b. Não
  - c. Parcialmente
- 10) Você recomendaria a continuação do uso da XPlannerPlus nos projetos da organização?
  - a. Sim
  - b. Não
  - c. Parcialmente

### 6.3. Ambiente

A organização escolhida para o estudo de caso é considerada de pequeno porte. Foi criada recentemente – em 2003 – e possui atualmente 23 colaboradores entre gerentes de projeto, engenheiro de processo, analistas de sistemas, engenheiros de software e testadores. A organização possui equipes geograficamente distribuídas em três unidades de desenvolvimento. Cabe ressaltar ainda que, especialmente por se tratar de uma organização de pequeno porte, é comum na organização os colaboradores assumirem mais de um papel em um ou mais projetos, por exemplo: ser analista de sistemas e desenvolvedor ou gerente de projetos e analista de sistemas.

O foco da organização é o desenvolvimento de projetos e produtos nas áreas de gestão eletrônica de documentos, gestão de terceiros e sub-contratadas, gestão imobiliária, administração escolar e geração de horários escolares, além de personalização de ferramentas de ensino à distância.

A organização pode ser considerada como tradicional, visto que possui seu processo de desenvolvimento baseado nos modelos MPS.BR níveis G e F (MPS.BR, 2007) e CMMI nível 2 (SEI, 2002). Apesar de o processo estar definido, ele não está institucionalizado na organização. No entanto, a organização possui alguns aspectos e características relacionadas aos métodos ágeis:

- Projetos pequenos e desenvolvidos em iterações;
- Documentação não muito extensa;
- Presença no local de desenvolvimento do cliente para os projetos mais importantes;
- Planejamento iterativo e com participação do cliente;
- Processo de desenvolvimento que encoraja as mudanças;
- Desenvolvedores que assumem vários papéis;

- Gerentes de Projetos com postura de liderança e colaboração, compartilhando a tomada de decisões no projeto;

A gerência dos projetos é dividida entre os gerentes de projetos e os líderes de projeto, que também desempenham papel técnico. A principal ferramenta de suporte à gerência de projetos é o Microsoft Project. Porém, usos isolados das ferramentas DotProject e OpenProject foram relatados pelos gerentes de projetos. Além do Microsoft Project, uma planilha Excel era utilizada para registro dos requisitos, lançamento das horas realizadas com as tarefas e distribuição das horas para cada requisito.

A organização optou em realizar, através da XPlannerPlus, apenas o controle e acompanhamento das atividades relacionadas ao desenvolvimento dos requisitos, isto é, não cadastrou atividades de prospecção, análise, homologação e aceitação dos projetos na ferramenta. Logo, de acordo com o processo da organização, em média, entre 65% e 70% da estimativa total do esforço do requisito seria alocado para:

- Criação do Documento de Caso de Teste (DCT): 5%;
- Implementação: 40 a 45%;
- Testes: 15%;
- Correção: 5%;

A escolha dos projetos seguiu critérios de representatividade na organização, características (tamanho do projeto, número de requisitos, tecnologia, etc.), diferença nas equipes (tamanho, conhecimento dos membros, etc.) e data de início dos projetos. Todos os projetos foram de desenvolvimento de sistemas *Web*. No total, 13 pessoas participaram do estudo de caso, sendo 11 representantes das equipes de desenvolvimento e 2 representantes da diretoria.

As seções a seguir descrevem os detalhes de cada projeto.

## **6.4. Gerência do Projeto A**

### **6.4.1. Caracterização**

O projeto A trata-se do desenvolvimento da terceira versão de um sistema de gestão de arquivos físicos. Este sistema permite o cadastro de documentos e de seu endereço em um arquivo físico, facilitando a organização e recuperação destes documentos.

O estudo de caso compreendeu somente a segunda de duas iterações. O número total de requisitos foi 12 com o total de 210 horas.

Basicamente, esta segunda iteração do projeto consistiu do desenvolvimento de:

- Cadastro e manutenção de dados básicos para funcionamento do sistema;
- Pesquisa de documentos no acervo físico da organização;
- Relatório de documentos emprestados e devolvidos; e
- Armazenamento e visualização do histórico das ações executadas no sistema.

A equipe do projeto consistia de 1 analista de sistemas, 1 desenvolvedor, 1 testador e 1 gerente de projetos (também com papel de desenvolvedor).

### **6.4.2. Uso da Ferramenta**

A etapa inicial do planejamento deste projeto aconteceu de maneira usual. A equipe de gerenciamento construiu o cronograma utilizando a ferramenta Microsoft Project e realizou a distribuição das horas de cada tarefa em planilha Excel. A gerente do projeto e os desenvolvedores definiram juntos quais seriam as tarefas básicas dos requisitos: Criação DCT (Documento de Caso de Teste), Implementação BD (Banco de Dados), Implementação ASP (interface do sistema), Testes e Correção.

O projeto iniciou-se em 11/03/2009 e foi o primeiro projeto a usar a XPlannerPlus na organização. Logo de início, a gerente do projeto detectou que seu perfil de usuário não tinha acesso aos novos elos de navegação da XPlannerPlus: *Risks*, *Obstacles*, *Post-Mortem* e *Success*, o que não a impedia de utilizar a ferramenta. A correção deste pequeno problema deu-se no mesmo instante.

Ao fim da primeira semana de uso da ferramenta, um detalhe interessante pôde ser notado no projeto, conforme Figura 6.1. Ao visualizar o progresso do RF017, a primeira impressão é que o requisito está atrasado, contudo, de fato, as suas tarefas de implementação já foram realizadas, porém, com uma quantidade de horas bem inferior ao estimado originalmente. A Figura 6.2 mostra que este progresso reflete 1,3 horas produzidas sobre 5,4 horas da estimativa atual, o que significa apenas 24% de progresso. Além disso, a figura citada também mostra que todos os requisitos foram planejados para esta iteração e todos estão na situação “Estimado”.

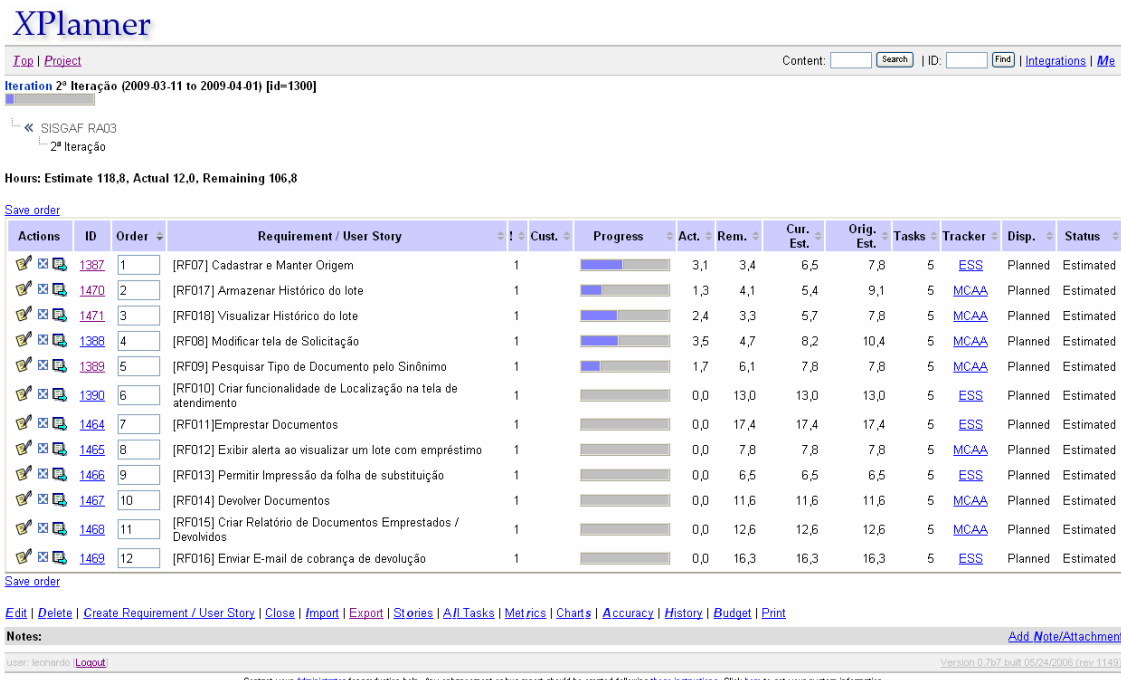


Figura 6.1 - Projeto A: acompanhamento da primeira semana de desenvolvimento

**XPlanner**

Top | Project | Iteration Content:  Search | ID:  Find | Integrations | Me

« SISGAF RA03  
 ↳ 2ª Iteração  
 ↳ « [RF017] Armazenar Histórico do lote »

**Requirement / User Story: [RF017] Armazenar Histórico do lote [id=1470]**

O sistema passará a armazenar o histórico de um lote

**Priority:** 1 **Estimated Hours:** 5,4 (9,1)  
**Actual Hours:** 1,3  
**Tracker:** [Michelle Cristina Alves de Andrade](#) **Remaining Hours:** 4,1  
**Last Update:** 2009-03-12 11:24 **Disposition:** Planned  
**Status:** Estimated

Actions	ID	Task Name	Type	Progress	Acc.	Ori.	Est.	Act.	Rem.	Disp.	Type
	<a href="#">1567</a>	Correção	Overhead	<div style="width: 100%;"></div>	<a href="#">MCAA</a>	1,3	1,3	0,0	1,3	Planned	Overhead
	<a href="#">1563</a>	Criação DCT	FTest	<div style="width: 100%;"></div>	<a href="#">SM</a>	0,7	0,7	0,0	0,7	Planned	FTest
	<a href="#">1565</a>	Implementação ASP	Feature	<div style="width: 100%;"></div>	<a href="#">MCAA</a>	2,0	2,0	0,4	0,0	Planned	Feature
	<a href="#">1564</a>	Implementação BD	Feature	<div style="width: 100%;"></div>	<a href="#">MCAA</a>	3,0	2,9	0,9	0,0	Planned	Feature
	<a href="#">1566</a>	Testes	FTest	<div style="width: 100%;"></div>	<a href="#">SM</a>	2,1	2,1	0,0	2,1	Planned	FTest

[Edit](#) | [Delete](#) | [Move/Continue](#) | [Create Task](#) | [Export](#) | [History](#) | [Print](#)

**Notes:** [Add Note/Attachment](#)

user: leonardo [Logout](#) Version 0.767 built 05/24/2006 (rev. 1149)

Contact your [Administrator](#) for production help. Any enhancement or bug report should be created following [these instructions](#). Click [here](#) to get your system information.

**Figura 6.2 - Projeto A: tarefas do requisito 17**

Um dos riscos identificados no projeto estava relacionado à natureza do projeto. Como se tratava de um sistema já existente, porém novo na organização, havia uma preocupação com o pouco conhecimento da equipe sobre as regras de negócio do sistema. A Figura 6.3 mostra os detalhes do risco.

## XPlanner

[Top](#)

[Edit Risk:](#)

**Description:** [Formatting Help](#)

Pouco conhecimento sobre as Regras de Negócio

---

**Responsibles:** [Formatting Help](#)

Michelle

---

**Details:** [Formatting Help](#)

O sistema foi inicialmente desenvolvido por outra empresa, e o mesmo não possui nenhuma documentação.

---

**Status:**

**Date:**

**Impact (1 to 10):**

**Probability (1 to 10):**

**Priority (10 to 1):**

**Figura 6.3 - Projeto A: risco identificado**

Já a Figura 6.4 mostra uma listagem dos obstáculos identificados pela equipe do projeto. Nesta listagem apenas a descrição e a data de identificação do obstáculo são exibidas.

## XPlanner

[Top](#) Content:   | ID:   | [Integrations](#) | [Me](#)

**Project:** SISGAF RA03 [id=1310]

« SISGAF RA03 »

Actions	ID	Obstacle's Description	Obstacle's Date
<a href="#">2082</a>		[RF08 - Modificar Tela de Solicitação] A maior dificuldade foi encontrar como era feita a passagem de parametros para a página seguinte. A página em questão usava 3 forms diferentes.	2009-03-18
<a href="#">2083</a>		[RF09 - Pesquisar Tipo de Documento pelo Sinônimo] Problema ao tentar abrir um pop-up e fazê-lo comunicar com a janela principal.	2009-03-19
<a href="#">2086</a>		Dificuldade na montagem e atualização da aplicação em Lavras.	2009-03-31
<a href="#">2084</a>		Dificuldade de manter o banco de teste atualizado corretamente em Lavras	2009-03-25

4 items found, displaying all items. 1

[Create Obstacle](#) | [Print](#)

**Notes:** [Add Note/Attachment](#)

User: leonardo | [Logout](#) Version 0.7b7 built 05/24/2006 (rev. 1149)

Contact your [Administrator](#) for production help. Any enhancement or bug report should be created following [these instructions](#). Click [here](#) to get your system information.

**Figura 6.4 - Projeto A: listagem de obstáculos identificados**

Por fim, a avaliação final do projeto foi realizada pela equipe. A informação inserida através da funcionalidade “Cadastrar informação sobre avaliação final do projeto” pode ser visualizada na Figura 6.5.



The screenshot displays the XPlanner web interface. At the top, the title "XPlanner" is shown in a large blue font. Below it, there is a "Top" link and an "Edit Post-Mortem:" label. The main content area is divided into four sections, each with a "Formatting Help" link:

- Best Practices:** Implementação de armazenamento de logs de execução de serviços de e-mails;
- Worst Practices:** Manutenção da utilização excessiva de Store Procedures, sem a utilização de uma camada exclusiva para acesso ao banco de dados;
- Lessons Learned:** Necessidade de revisão e estudo da documentação atual antes do início do projeto;
- Future Of Project:** O sistema foi implantado, contudo, novas funcionalidades serão necessárias para atender as demandas de modificações em relatórios e para implantação em outros clientes.

At the bottom of the form, there are two buttons: "Update" and "Reset".

Figura 6.5 - Projeto A: informação sobre a avaliação final do projeto

### 6.4.3. Resultados

Um dos pontos negativos da ferramenta, segundo a gerente do projeto A é o fato de a XPlannerPlus não disponibilizar uma data estimada para o término de cada requisito. Na verdade, esta ausência, classificada como “ponto negativo”, pode estar relacionada à cultura tradicional da organização e dos GPs, que sempre trabalharam com o planejamento detalhado proporcionado pelo Microsoft Project.

Já as funcionalidades de registro de risco e obstáculos do projeto foram muito utilizadas pela equipe do projeto, facilitando a comunicação e resolução sobre os



problemas previstos e encontrados no projeto. Segundo a gerente do projeto, antes da utilização da ferramenta, a gerência de risco praticamente não era executada e não havia histórico sobre os obstáculos enfrentados no projeto.

Por fim, cabe mencionar uma sugestão dada pela gerente do projeto: a construção de um relatório que exiba somente a descrição do requisito e o progresso de cada um deles. O objetivo deste relatório é condensar o progresso do projeto em poucos elementos a fim de facilitar o acompanhamento pelo cliente.

## **6.5. Gerência do Projeto B**

### **6.5.1. Caracterização**

O projeto B é o desenvolvimento de um sistema de controle de comprovantes e solicitações de entrega de lista telefônica. A partir do sistema será possível visualizar comprovantes digitalizados de entrega de lista telefônica, bem como acompanhar o fluxo de solicitações de entrega de lista telefônica.

O estudo de caso abordou somente a segunda de três iterações, que possuía 16 requisitos totalizando 257 horas.

Basicamente, esta segunda iteração compreendeu o desenvolvimento de:

- Cadastro e manutenção de dados básicos para funcionamento do sistema; e
- Funcionalidades de controle de fluxo das solicitações de lista telefônica;

Três desenvolvedores, 1 testador e 1 gerente de projetos (também com papel de analista de sistemas) compunham a equipe do projeto.

### 6.5.2. Uso da Ferramenta

O uso da XPlannerPlus neste projeto começou com duas particularidades: o gerente do projeto teve a iniciativa de utilizar uma funcionalidade que importava os requisitos de uma planilha Excel o que resultou em ganho de tempo considerável. Contudo, depois de cadastrar as tarefas do primeiro requisito, um dos desenvolvedores lançou os tempos gastos, com isto, a iteração iniciou com os demais requisitos sem a estimativa original. Foi preciso excluir a iteração e importar novamente os requisitos.

O início do projeto ocorreu em 16/03/2009. E ainda no início do projeto outra particularidade ocorreu: a inclusão de dois requisitos após o início da iteração. A Figura 6.6 mostra que o RF046 e o RF010 não possuem estimativa original e estão marcados como “*Added*”, isto é, foram adicionados à iteração após o seu início. Além disso, pode-se notar também que o RF010 já havia sido implementado, tanto visualizando sua barra de progresso em verde, quanto a partir de sua situação “*Implemented*”.

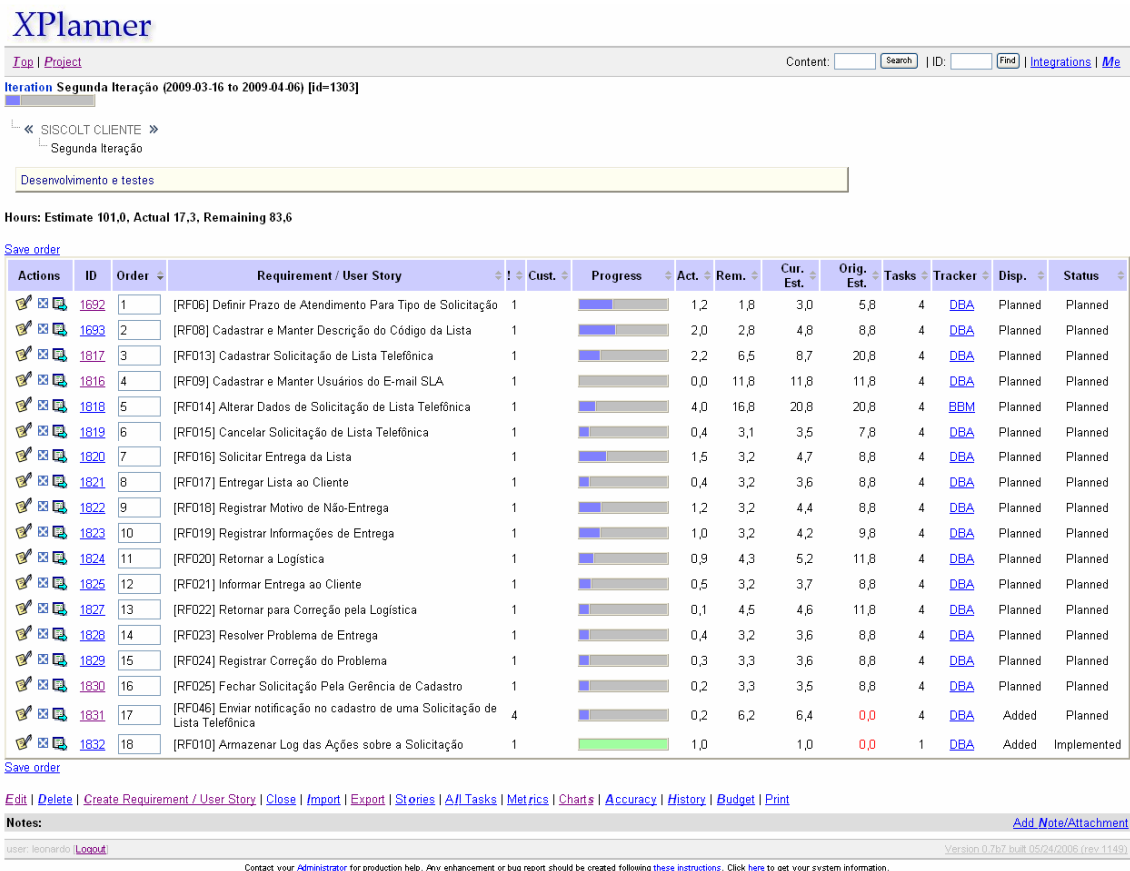


Figura 6.6 - Projeto B: acompanhamento da primeira semana de desenvolvimento

Durante o desenvolvimento da iteração, os gráficos gerados pela XPlannerPlus chamaram a atenção do gerente do projeto e da equipe de desenvolvimento. Ambos mostraram uma queda acentuada da estimativa atual (Figura 6.7) e do trabalho restante da iteração (Figura 6.8), principalmente entre o oitavo e o décimo dia. Após análise, verificou-se que dois fatores causaram esta queda abrupta:

- Apesar de realizar as tarefas anteriormente, o colaborador responsável pelos testes deixou para lançar as tarefas de “Criação de DCT” nestes dois dias;
- Um dos desenvolvedores teve uma alta produtividade em seus requisitos, pois na primeira iteração do projeto, um componente criado facilitou a implementação dos mesmos.

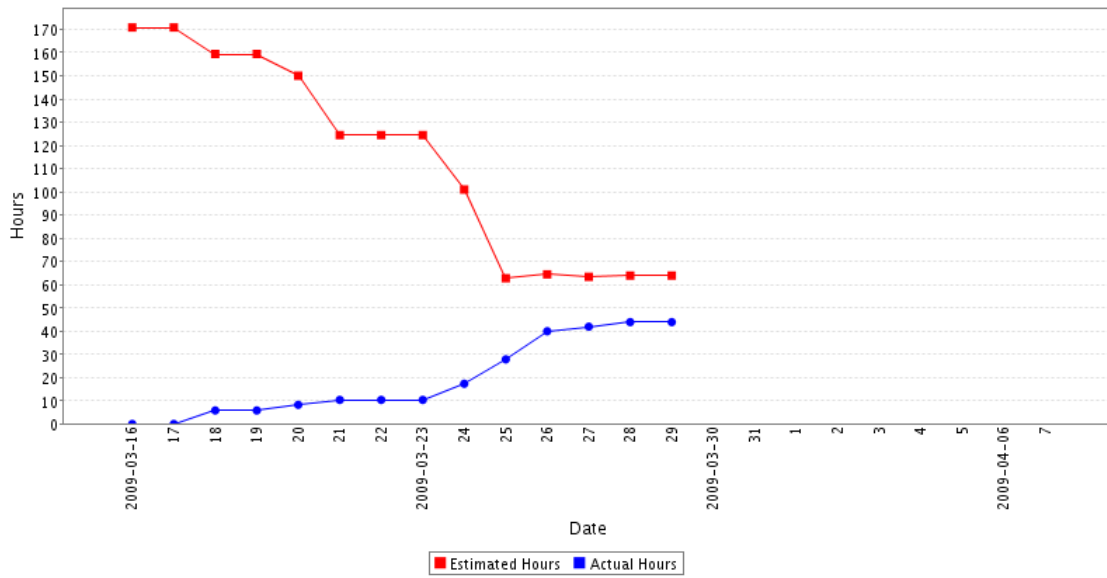


Figura 6.7 - Projeto B: gráfico de progresso da iteração

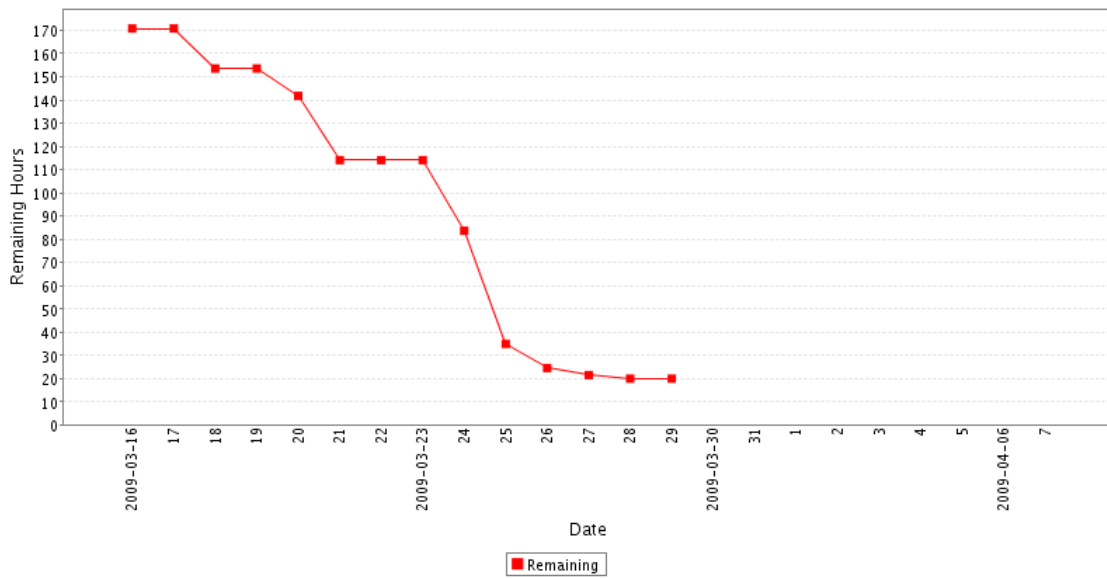


Figura 6.8 - Projeto B: gráfico do trabalho restante

Este projeto não realizou o registro de riscos, porém a equipe do projeto inseriu 4 obstáculos que surgiram durante o desenvolvimento do projeto. A Figura 6.9 mostra o detalhamento de um destes obstáculos.

**XPlanner**

[Top](#)

**Edit Obstacle:**

**Description:** [Formatting Help](#)  
O desenvolvimento do [RF05] Login Integrado (1a iteração) está impactando a segunda iteração, pois a equipe está com dificuldade de encontrar uma solução para ele na

**Relator:** [Formatting Help](#)  
André Ribeiro Barros

**Observation:** [Formatting Help](#)  
Uma das possíveis soluções depende de configurações no browser do cliente. Estamos tentando uma segunda solução que não vai depender de configurações no cliente mas sua implementação está sendo difícil.

**Date:** 2009-03-17

**Status:** New

**Category:** Technic

Figura 6.9 - Projeto B: obstáculo identificado

### 6.5.3. Resultados

O gerente do Projeto B elogiou a facilidade de entrada dos dados na ferramenta. Apesar de ter gastado um tempo inicial com a importação dos requisitos e o lançamento das tarefas, o gerente do projeto classificou a ferramenta como um importante auxílio no planejamento e acompanhamento do projeto.

A equipe de desenvolvimento também se mostrou satisfeita com o uso de uma ferramenta *Web* ao contrário de atualizar a informação em uma planilha Excel. A funcionalidade de registro de obstáculos construído na ferramenta como parte deste trabalho, também foi muito utilizado pela equipe e facilitará na resolução de problemas no restante do projeto B, bem como nos demais projetos da organização.

Ainda segundo a equipe do projeto, a utilização da ferramenta deu caráter mais ágil ao projeto, permitindo um acompanhamento mais próximo pelo gerente do projeto, bem como agilizando a comunicação dentro do projeto e diminuindo a

documentação gerada, visto que as funcionalidades personalizadas possibilitaram o armazenamento de diversos elementos em um único local.

## **6.6. Gerência do Projeto C**

### **6.6.1. Caracterização**

O projeto C trata-se do desenvolvimento de novas funcionalidades em um sistema de gestão de vagas de emprego e capacitações para jovens.

O estudo de caso compreendeu somente a segunda de três iterações. O número de requisitos foi 11 com o total de 204 horas.

Basicamente, o escopo desta iteração foi o desenvolvimento de:

- Cadastro e manutenção de oportunidades de empregos;
- Associação de jovens às vagas, de acordo com seu perfil; e
- Criação de cinco novos relatórios.

A equipe do projeto consistia de 2 desenvolvedores (sendo que um deles possuía papel de analista de sistemas), 1 testador e 1 gerente de projetos (autor do trabalho).

### **6.6.2. Uso da Ferramenta**

Como foi o último projeto a iniciar o uso da XPlannerPlus neste estudo de caso, seu uso foi bem facilitado. Apesar de se tratar de uma equipe diferente, diversas dúvidas que poderiam requerer um tempo maior, foram prontamente respondidas. Contudo, como pode ser observado na Figura 6.10, apesar do início do projeto ter sido dia 23/03/2009, somente dia 25/03/2009 é que os colaboradores começaram a lançar as atividades.

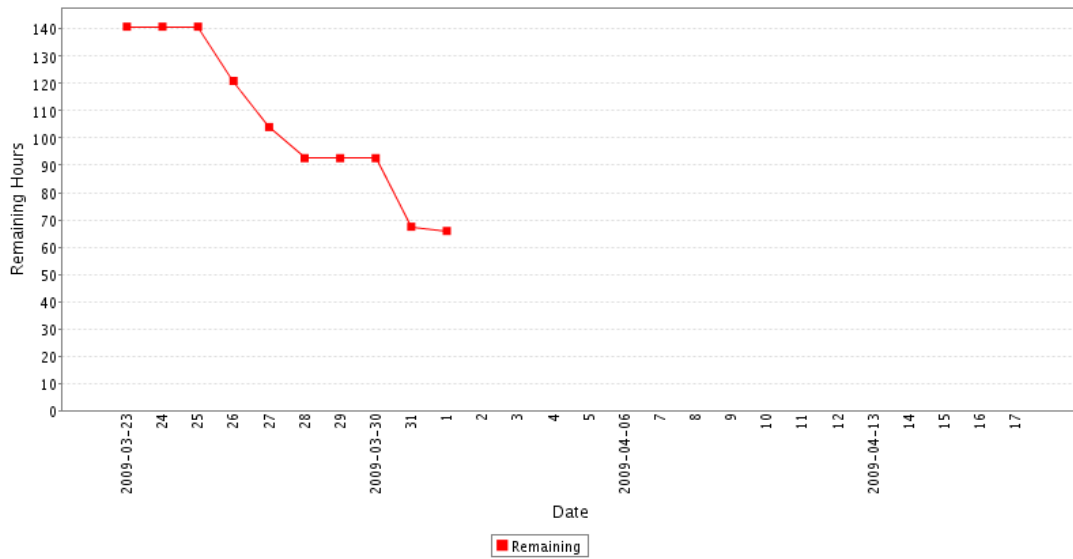
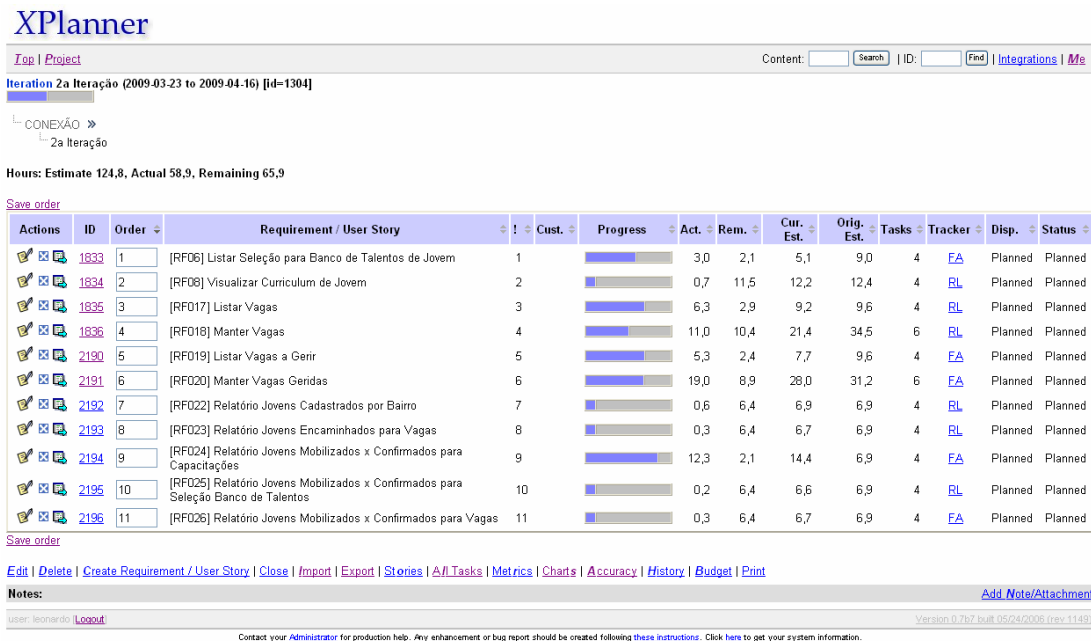


Figura 6.10 - Projeto C: gráfico do trabalho restante

Assim como os demais projetos, o planejamento da iteração iniciou-se com a construção de uma planilha em Excel para facilitar a distribuição das estimativas entre as tarefas definidas para os requisitos do projeto: Criação DCT, Implementação Cadastro, Implementação Alteração, Implementação Exclusão, Testes e Correção, para os requisitos “Manter” (RF018 e RF020) e Criação DCT, Implementação, Testes e Correção para os demais requisitos.

A Figura 6.11 lista os requisitos da iteração.



**Figura 6.11 - Projeto C: acompanhamento da segunda semana de desenvolvimento**

Durante o desenvolvimento do projeto, as funcionalidades de cadastro de riscos e obstáculos foram utilizadas pela equipe. A definição do sucesso do projeto também foi utilizada por toda a equipe interna do projeto.

A Figura 6.12 exibe uma das definições do sucesso do projeto de acordo com o gerente do projeto.



The screenshot shows the XPlanner web interface. At the top, the logo 'XPlanner' is displayed in blue. Below it, there are links for 'Top' and 'Edit Success:'. The main content area is a form titled 'User: leonardo' with a link 'Update Sliders'. Below this is a slider control with markers for 'Off', '25%', '50%', '75%', and 'On'. The slider is currently positioned at 50%. Below the slider are several rows of settings, each with a shield icon and a text input field:

Category	Value
Stakeholders's Satisfaction:	3
Customers's Requirements:	4
Within the budget:	4
Within the deadline:	1
Value added:	2
Within the Standard of quality:	3
Team's satisfaction:	2

At the bottom of the form are two buttons: 'Update' and 'Reset'.

Figura 6.12 - Projeto C: definição do sucesso do projeto

Já a avaliação da iteração não foi realizada, visto que o gerente do projeto optou pela conclusão da terceira iteração para a realização da avaliação final do projeto.

### 6.6.3. Resultados

Uma das primeiras dificuldades encontradas no uso da XPlannerPlus foi a distribuição e lançamento das estimativas de esforço para cada uma das tarefas dos requisitos. Uma planilha Excel foi utilizada para facilitar a distribuição da estimativa de esforço de desenvolvimento para cada tarefa de cada requisito.

Outro ponto negativo, já citado pelo gerente do projeto A, foi a ausência na XPlannerPlus de uma data específica para a conclusão de cada requisito. Na verdade, este foi um problema levantado também pela direção da organização. Como já mencionado, provavelmente esta necessidade foi levantada devido à cultura da organização de trabalhar com cronogramas e marcos definidos com os clientes, características da gerência tradicional de projetos e do uso do Microsoft Project.

Em contrapartida, a facilidade no lançamento da folha de registro de horas foi elogiada por todos os envolvidos no projeto. Decorrente disso, o acompanhamento em tempo-real do projeto tornou-se algo verdadeiramente possível na organização, o que não ocorria anteriormente.

Em relação ao cadastro dos riscos e obstáculos, mais uma vez esta iniciativa foi elogiada, tanto pela equipe de desenvolvimento quanto pela direção da organização, que enxergou ali um repositório para estas questões que sempre podem surgir nos projetos.

A definição do sucesso do projeto, mesmo sendo utilizada apenas pela equipe interna do projeto facilitou o direcionamento das atenções do gerente do projeto. Pela primeira vez na organização, os desenvolvedores foram ouvidos quanto às metas e objetivos do projeto.

## 6.7. Resultados dos Questionários

Conforme mencionado, a pesquisa através do questionário foi realizada com 13 pessoas, sendo que o envio das respostas foi sigiloso. Cabe ressaltar que esta pesquisa foi realizada após o uso da XPlannerPlus, conforme descrito neste capítulo. O objetivo foi coletar a percepção dos usuários frente às personalizações realizadas.

A Tabela 6.1 mostra uma síntese das respostas do questionário.

**Tabela 6.1 - Síntese das respostas do Questionário**

Questão	SIM	NÃO	PARCIAL-MENTE
1. A funcionalidade Cadastro de Riscos da XPlannerPlus foi utilizada?	7	3	3
2. A funcionalidade Cadastro de Riscos da	7	3	3

XPlannerPlus atendeu à sua expectativa?			
3. A funcionalidade <b>Cadastro de Obstáculos</b> da XPlannerPlus foi utilizada?	10	2	1
4. A funcionalidade <b>Cadastro de Obstáculos</b> da XPlannerPlus atendeu à sua expectativa?	10	1	2
5. A funcionalidade <b>Cadastro da Informação de Avaliação Final de Projeto</b> da XPlannerPlus foi utilizada?	7	2	4
6. A funcionalidade <b>Cadastro da Informação de Avaliação Final de Projeto</b> da XPlannerPlus atendeu à sua expectativa?	7	2	4
7. A funcionalidade de <b>Definição do Sucesso do Projeto</b> da XPlannerPlus foi utilizada?	6	2	5
8. A funcionalidade de <b>Definição do Sucesso do Projeto</b> da XPlannerPlus atendeu à sua expectativa?	5	2	6
9. A usabilidade (desempenho, facilidade de uso, facilidade de aprendizagem, termos utilizados, etc.) da XPlannerPlus atendeu à sua expectativa?	12	0	1
10. Você recomendaria a continuação do uso da XPlannerPlus nos projetos da organização?	13	0	0

Devido à escassez de tempo e ao número reduzido de participantes da entrevista não foram realizadas análises estatísticas e o adequado tratamento aos questionários.

Contudo, com base nas respostas e na função de cada participante temos evidências que as funcionalidades indicadas nas questões 1, 3, 5 e 7 estiveram mais relacionadas às funções de gerente de projeto e diretoria. Já a maioria dos que responderam que não utilizaram as funcionalidades indicadas nas questões 1, 5 e 7 são os desenvolvedores, ou seja, possuem a tendência de se preocupar mais com os obstáculos do projeto. Ao contrário disso, o gerente do projeto que tem uma necessidade maior de prever os riscos e problemas que afetarão o projeto e a diretoria da organização tem interesse na definição do sucesso do projeto e na informação coletada durante as avaliações finais dos projetos.

De maneira geral, temos evidências que as novas funcionalidades da XPlannerPlus foram utilizadas e atenderam as expectativas da maioria dos participantes do estudo de caso. Além disso, temos evidências também, especialmente a partir das respostas das questões 9 e 10, que a aceitação do uso da XPlannerPlus foi positiva. Sendo que 100% dos participantes recomendam a continuação de seu uso na organização.

## **6.8. Aspectos Observados**

O estudo de caso compreendeu 3 projetos e 13 participantes em uma pequena organização que se considera como “tradicional”. Apesar de ter sido um estudo de caso relativamente pequeno e que durou em torno de 45 dias, seus objetivos foram atingidos.

Foi possível verificar que:

- As funcionalidades personalizadas “Cadastrar, alterar, remover e listar Riscos”, “Cadastrar, alterar, remover e listar Obstáculos” e “Cadastrar Informação sobre Avaliação Final do Projeto” mostraram-se úteis ao projeto e tiveram boa aceitação pela equipe de desenvolvimento e gerentes de projetos;

- A funcionalidade personalizada “Definir Sucesso do Projeto”, apesar de ter sido declarada como útil pela equipe interna, não foi integralmente utilizada, em virtude do não acesso pelos clientes da organização;
- A funcionalidade personalizada “Gerar Orçamento” foi parcialmente utilizada pela organização, devido a questões de permissão de acesso à funcionalidade e por envolver os custos com mão de obra do projeto;

Contudo, os resultados alcançados não podem ser generalizados em larga escala. Acreditamos que estudos mais longitudinais são necessários para eliminar fatores tendenciosos que impactaram o estudo de caso. Primeiramente, outras equipes em outros projetos precisam ser observadas antes que generalizações possam ser feitas sobre o uso e eficácia da XPlannerPlus. Alguns fatores, em especial, precisam ser considerados:

- O estudo de caso e entrevista através dos questionários foram conduzidos e analisados pelo mesmo investigador, que também é o desenvolvedor da XPlannerPlus;
- O pesquisador pode de alguma forma ter exercido algum tipo influência sobre o uso da XPlannerPlus e a resposta ao questionário;
- O fato de os gerentes de projeto já conhecerem bem os demais membros da equipe pode ter gerado algum resultado tendencioso.

Além disso, a ajuda sobre as abordagens ágil e tradicional de gerência de projetos disponibilizada na XPlannerPlus também contribuiu para o emprego da ferramenta e de características ágeis na organização do estudo de caso, que se considera como “tradicional”. A respeito disso, pode-se afirmar que o uso da ferramenta de gerência ágil no ambiente “tradicional” da organização não causou estranheza aos usuários. Apesar de algumas diferenças relacionadas ao planejamento e monitoramento do projeto, as características ágeis introduzidas na gerência de projetos por este estudo de caso foram consideradas muito bem vindas por todos os envolvidos, além de não terem afetado negativamente o desenvolvimento dos projetos.

Ainda relacionado às características ágeis, a XPlannerPlus permitiu uma maior comunicação e colaboração entre os envolvidos no projeto, promoveu a facilitação no replanejamento do projeto, sempre que foi necessário, além de ter permitido a gestão dos riscos e obstáculos encontrados no projeto.

No entanto, alguns pontos fracos da XPlannerPlus foram levantados pelas equipes de desenvolvimento, gerentes de projeto e diretoria da organização:

- Ausência de suporte à geração de cronograma, mesmo simplificado, com uma previsão da data de término de cada requisito;
- Dificuldade para distribuição das estimativas de esforço para cada uma das tarefas dos requisitos; e
- Criação de perfil específico para o “Cliente”. Atualmente a ferramenta disponibiliza apenas o perfil “Visualizador”.

## 6.9. Considerações Finais

Apesar de envolver poucos projetos e poucos colaboradores, o estudo de caso mostrou-se muito útil. Diversos aspectos da XPlannerPlus foram avaliados.

Além dos aspectos citados na seção anterior, outras considerações do estudo de caso precisam ser feitas:

(i) Uma das funcionalidades personalizadas que não foram utilizadas pela equipe dos projetos, mas somente pela direção da organização foi a Geração do Orçamento do Projeto. Como esta funcionalidade estava disponível para todos os usuários, a direção solicitou alteração no perfil do usuário “administrador” para que somente este tivesse acesso a esta funcionalidade.

(ii) Em relação ao uso da ferramenta pelos clientes do projeto, a organização não autorizou seu uso, alegando que gostaria de uma avaliação interna sobre o uso da

ferramenta neste aspecto. Contudo, sinalizou que futuramente pretende disponibilizar a ferramenta para seus clientes.

(iii) Como os clientes e usuários não tiveram acesso a XPlannerPlus, a funcionalidade personalizada “Definir Sucesso do Projeto” não foi adequadamente validada, já que contou apenas com as definições da equipe interna dos projetos.





# Capítulo 7

## Conclusão

### **7.1. Considerações Finais**

Uma gerência de projetos eficaz é de fundamental importância para o sucesso de um projeto, mas é ainda mais crítica para projetos de software, onde segundo Schnaider (Schnaider, 2003), a qualidade dos produtos a serem desenvolvidos e o nível de produtividade obtido no desenvolvimento é altamente dependente da competência e alocação dos profissionais, bem como da gerência sobre o projeto e equipe.

A sociedade atual cada vez mais depende de aplicações de software, e as organizações que desenvolvem software precisam estar atentas aos padrões de qualidade do mercado, bem como ao atendimento dos prazos, orçamentos e necessidades dos usuários. As organizações precisam analisar o ambiente de seus projetos, a fim de determinar parâmetros que auxiliem na definição de qual a melhor estratégia a utilizar, tanto para o desenvolvimento quanto para a gerência de seus projetos. De acordo com Boehm (Boehm, 2002), a análise de risco das características do projeto versus determinadas características básicas dos métodos podem ajudar a determinar o melhor equilíbrio entre disciplinas ágeis e tradicionais.

Para tanto, conhecer as características e possibilidades provenientes da gerência ágil e da gerência tradicional auxilia a tarefa tanto dos responsáveis pela organização quanto dos responsáveis por seus projetos. Entretanto, segundo Nerur e outros (Nerur et. al., 2005) os métodos ágeis são ideais para projetos que apresentam alta variabilidade

em tarefas (devido a mudança nos requisitos), nas capacidades das pessoas e nas tecnologias utilizadas.

Cabe, principalmente ao gerente de projetos, buscar o ponto de equilíbrio entre as abordagens, avaliando os riscos pertinentes a fim de se aproveitar o melhor que cada abordagem tem a oferecer. Uma abordagem mista, com o planejamento aperfeiçoando a agilidade, e, em contra-partida, com a agilidade dando eficiência ao planejamento é uma solução recomendada mas que é de difícil caracterização quantitativa.

Considerando a importância sobre o conhecimento em gerência de projetos, bem como as características que podem levar o gerente de projetos a adotar uma abordagem ágil ou tradicional de gerência, este trabalho apresentou um estudo de alguns aspectos das duas abordagens. Com este estudo, esperamos ter contribuído na facilitação das decisões do gerente de projetos acerca das práticas assim como incentivar a adoção de práticas menos burocráticas de gerência por parte, especialmente, de organizações de médio e pequeno porte que utilizam técnicas de gerência de maneira informal e isolada.

Aliado a isto, apresentamos a XPlannerPlus, extensão da XPlanner, uma ferramenta que agrega funções de gerência de projetos de forma menos burocrática e mais ágil tornando-se um aspecto importante em busca do atendimento das necessidades e expectativas dos clientes da organização de desenvolvimento de software.

Portanto, como principais contribuições deste trabalho podem-se citar:

- Definição de um paralelo entre a abordagem ágil e a abordagem tradicional de gerência de projetos de software;
- Definição das características dos projetos que influenciam na decisão do gerente de projetos sobre qual a melhor abordagem a ser utilizada;
- Elaboração de um guia que visa orientar os gerentes de projeto a cerca das diferenças entre as abordagens ágil e tradicional de gerência e definir quais são os papéis do gerente de projetos no desenvolvimento ágil;

- Personalização da ferramenta XPlanner, resultando na XPlannerPlus, de acordo com a listagem de requisitos elaborada; e
- Validação através de um estudo de caso em uma organização de pequeno porte utilizando a ferramenta XPlannerPlus;

Como forma de validação, a ferramenta XPlannerPlus foi utilizada na gerência de três projetos de uma organização de desenvolvimento de software de pequeno porte. A ferramenta foi utilizada a partir do planejamento e durante todo o desenvolvimento de uma das iterações de cada um dos projetos. Os resultados obtidos (apresentados no capítulo 6) apontam que as personalizações realizadas neste trabalho acrescentaram valor à ferramenta e mostraram-se úteis no ambiente de desenvolvimento da organização do estudo de caso. A organização, mesmo considerada “tradicional” não teve problemas em utilizar uma ferramenta de apoio a práticas ágeis de gerência de projetos. Tanto a equipe e o gerente de projeto, que não possuíam conhecimentos consistentes em métodos ágeis, não tiveram problemas no uso da ferramenta.

Acredita-se que a utilização da XPlannerPlus, bem como o conhecimento associado entre as características dos projetos e os princípios e processos das abordagens de gerência de projeto existentes possam contribuir substancialmente com a qualidade dos produtos e produtividade dos projetos. Utilizando o conteúdo deste trabalho, tanto o gerente de projetos quanto os gestores de uma organização terão subsídio em suas decisões gerenciais, o que pode, inclusive, facilitar a adoção de gerência de projetos por organizações que não o fazem.

## **7.2. Trabalhos Futuros**

Em busca de maior aprimoramento sobre estudos de abordagens de gerência de projetos de software e a utilização de ferramentas como suporte a tais abordagens, alguns trabalhos futuros são destacados.

Inicialmente, o estudo comparativo poderia abranger outras abordagens de gerência, tais como PRINCE2 (<<http://www.prince2.com>>), TENSTEP (<<http://www.tenstep.com.br>>) e abordagens baseadas na norma ISO / IEC 15504. O estudo poderia incluir os mesmos aspectos apresentados neste trabalho assim como considerar demais áreas não consideradas.

A continuação na personalização da ferramenta seria um outro trabalho futuro. Algumas das funcionalidades que poderiam ser acrescentadas são:

- Cadastro, Alteração e Listagem de Defeitos encontrados no projeto;
- Exibição de um painel único que mostrasse o progresso de todos os projetos em desenvolvimento na organização;
- Relatório simplificado com a descrição dos requisitos e o progresso de cada um visando à facilitação do acompanhamento do projeto pelos clientes;

Além destas, algumas outras necessidades, descritas no Capítulo 6, poderiam ser consideradas na implementação da ferramenta.

Ainda relacionado à XPlannerPlus, outra sugestão de trabalho futuro seria a sua utilização e validação em outras organizações e projetos. A validação em diferentes tamanhos, características e culturas é importante para atestar a utilidade da ferramenta em ambientes ágeis ou não.

Faz-se também necessário o desenvolvimento, aplicação e análise estatística de um questionário mais elaborado sobre a avaliação dos usuários a respeito da XPlannerPlus e futuros desenvolvimentos sobre ela.

Por fim, ao longo do trabalho foi identificada a ausência de trabalhos que consolidem a gerência de projetos à luz da metodologia ágil, porém que considerassem alguns aspectos da gerência tradicional. Devido a isto, seria interessante investigar a criação de uma abordagem híbrida, que agregasse conceitos e práticas de gerência ágil e tradicional. Esta abordagem poderia também considerar os aspectos e premissas de modelos de avaliação de processo de desenvolvimento, tais como CMMI e MPS.BR.

# Referências Bibliográficas

- ABRAHAMSSON, P.; WARSTA, J.; SIPONEN, M. T.; RONKAINEN, J. (2003). **New Directions on Agile Methods: A Comparative Analysis**. In Proceedings of the 25th IEEE International Conference on Software Engineering (ICSE), pp. 244 - 254.
- AGILE ALLIANCE. (2008). Disponível em: <<http://www.agilealliance.org>>. Visitado em 2008-08-03.
- ALLEMAN, G. B.; HENDERSON M. (2003). **Making Agile Development Work in a Government Contracting Environment Measuring velocity with Earned Value**. In Proceedings of the Agile Development Conference (ADC), pp. 1 – 6.
- ANDERSON, D. J.; SCHRAGENHEIM, E. (2003). **Agile Management for Software Engineering: Applying the Theory of Constraints for Business Results**. Prentice Hall PTR.
- ANDERSON, L.; ALLEMAN, G. B.; BECK, K.; BLOTNER, J.; CUNNINGHAM, W.; POPPENDIECK, M.; WIRFS-BROCK, R. (2003). **Agile Management – An Oxymoron? Who needs managers anyway?** 18th Conference on Object Oriented Programming Systems Languages and Applications (OOPSLA), pp. 275 - 277.
- ARAÚJO, C. (2008). **Softwares de apoio ao gerenciamento ágil de projetos colaborativos de novos produtos: análise teórica e identificação de requisitos**. Dissertação de Mestrado. Escola de Engenharia de São Carlos, USP.

- AUGUSTINE, S.; PAYNE, B.; SENCINDIVER, F.; WOODCOCK, S. (2005). **Agile Project Management: Steering from the Edges**. Communications of the ACM, v. 48, n. 12, pp. 85 - 89.
- BAUER, C.; KING, G. (2004). **Hibernate in Action**. 1<sup>st</sup> ed. Manning Publications.
- BECK, K. (1999). **eXtreme Programming Explained: Embrace Change**. Addison Wesley, Reading, MA.
- BOEHM, B. (2002). **Get ready for agile methods, with care**. IEEE Computer, vol. 35, n. 1, pp. 64 – 69.
- BOEHM, B.; TURNER, R. (2003). **Balancing agility and discipline: a guide for the perplexed**. Addison Wesley.
- BOEHM, B.; TURNER R. (2004). **Balancing agility and discipline: evaluating and integrating agile and plan-driven methods**. In Proceedings of the 26th IEEE International Conference on Software Engineering (ICSE), pp. 718 - 719.
- BOEHM, B.; TURNER R. (2005). **Management challenges to implementing agile processes in traditional development organizations**. IEEE Software, vol. 22, n. 5, pp. 30 - 39.
- CANNIZZO, F.; MARCIONETTI, G.; MOSER, P. (2008). **Evolution of the Tools and Practices of a Large Distributed Agile Team**. In Proceedings of the IEEE AGILE Conference, pp. 513 - 518.
- CESCHI, M.; SILLITTI, A.; SUCCI, G.; DE PANFILIS, S. (2005). **Project management in plan-based and agile companies**. IEEE Software, vol. 22, n. 3, pp. 21 - 27.
- CHEEMA, A.; SHAHID, A. A. (2005). **Customizing Project Management Methodology**. 9th International Multitopic Conference, IEEE INMIC, pp. 1 - 6.

- CHIN, G. (2004). **Agile Project Management: How to Succeed in the Face of Changing Project Requirements**. Amacom.
- COCKBURN, A. (2004). **Crystal Clear: A Human-Powered Methodology for Small Teams**. Addison-Wesley Professional.
- COCKBURN, A.; HIGHSMITH, J. (2001). **Agile Software Development: The People Factor**. IEEE Computer, vol. 34, n. 11, pp. 131 - 133.
- COHN, M.; FORD, D. (2003). **Introducing an Agile Process to an Organization**. IEEE Computer, vol. 36, n. 6, pp. 74 - 78.
- CORAM, M.; BOHNER, S. (2005). **The impact of agile methods on software project management**. In Proceedings of the 12th IEEE International Conference and Workshops on the Engineering of Computer-Based Systems (ECBS), pp. 363 - 370.
- DEPARTMENT OF DEFENSE / USA - DOD. (1994). **Report of the Defense Science Board Task Force on Acquiring Defense Software Commercially**. Junho / 1994. Disponível em: <[http://www.dod.mil/pubs/foi/reading\\_room/859.pdf](http://www.dod.mil/pubs/foi/reading_room/859.pdf)>. Visitado em 2008-09-05.
- GHANAM, Y.; WANG, X. MAURER, F. (2008). **Utilizing Digital Tabletops in Collocated Agile Planning Meetings**. In Proceedings of the IEEE AGILE Conference, pp. 51 - 62.
- GNU. (2009). Lesser General Public License. Disponível em: <<http://www.gnu.org/licenses/lgpl.html>>. Visitado em 2009-01-05.
- HANAKAWA, N.; OKURA, K. (2004). **A project management support tool using communication for agile software development**. In Proceedings of the 11th IEEE Asia-Pacific Software Engineering Conference (APSEC), pp. 316 - 323.

- HARTMANN, D.; DYMOND, R. (2006) **Appropriate agile measurement: using metrics and diagnostics to deliver business value**. In Proceedings of the IEEE AGILE Conference, pp. 128 - 134.
- HIBERNATE. (2008). Disponível em: <<http://www.hibernate.org/>>. Visitado em 2008-09-02.
- HIGHSMITH, J. (2004). **Agile project management: creating innovative products**. Addison Wesley. 2004.
- HIGHSMITH, J. (2002). **Agile Software Development Ecosystems**. Addison Wesley.
- HIGHSMITH, J.; COCKBURN, A. (2001). **Agile Software Development: The Business of Innovation**. IEEE Computer, vol. 34, n. 9, pp. 120 - 127.
- HIRSCH, M. (2005). **Moving from a plan driven culture to agile development**. In Proceedings of the 27th IEEE International Conference on Software Engineering (ICSE), pp. 244 - 254.
- ISO/IEC 15504-5. (2006). The International Organization for Standardization and the International Electrotechnical Commission, ISO/IEC 15504 - Information Technology - Process Assessment.
- JAVA. (2008). Disponível em: <<http://java.sun.com/>>. Visitado em 2008-09-02.
- JDBC. (2008). Disponível em: <<http://java.sun.com/javase/technologies/database/>>. Visitado em 2008-09-02.
- JOHNSON, J. (2001). **Micro Projects Cause Constant Change**. The Standish Group International, 2001.



- KERZNER, H. (2001). **Project Management**. 7ª. Edição. John Wiley & Sons.
- LEHMAN, M.; RAMIL, J. (2001). **Rules and Tools for Software Evolution Planning and Management**. Annals of Software Engineering, vol. 11, n. 1, pp. 15 - 44.
- LEUF, B.; CUNNINGHAM, W. (2001). **The Wiki Way: Quick Collaboration on the Web**. Addison-Wesley.
- LIU, L. (2006). **An Environment for Collaborative Agile Planning**. Dissertação de Mestrado. University of Calgary. Canadá.
- MARÇAL, A. S. C.; FREITAS, B. C. C.; SOARES, F. S. F.; FURTADO, M. E. S.; MACIEL, T. M.; BELCHIOR, A. D. (2008). **Blending Scrum practices and CMMI project management**. Innovations in Systems and Software Engineering, Springer, vol. 4, n. 1.
- MICROSOFT FRAMEWORK SOLUTION - MSF. (2008). Disponível em: <<http://www.microsoft.com/technet/solutionaccelerators/msf/default.aspx>>. Visitado em 2008-05-13.
- MORGAN R.; MAURER, F. (2006). **MasePlanner: A Card-Based Distributed Planning Tool for Agile Teams**. In Proceedings of the IEEE International Conference on Global Software Engineering (ICGSE), pp. 132 - 138.
- MORIEN, R. (2005). **Agile Management and the Toyota Way for Software Project Management**. In Proceedings of the 3rd IEEE International Conference on Industrial Informatics (INDIN), pp. 516 - 522.
- MPS.BR. (2007). **Guia MPS.BR**. Disponível em: <[http://www.softex.br/mpsbr/\\_home/default.asp](http://www.softex.br/mpsbr/_home/default.asp)>. Visitado em 2007-10-19.
- MYSQL. (2008). Disponível em: <<http://www.mysql.com/>>. Visitado em 2008-09-02.

- NERUR S.; MAHAPATRA, R.; MANGALARAJ, G. (2005). **Challenges of Migrating to Agile Methodologies**. Communications of the ACM, v. 48, n. 5.
- NETBEANS. (2008). Disponível em: <<http://www.netbeans.org/>>. Vistado em 2008-09-02.
- PALMER, S. R.; FELSING, J. M. (2001). **A Practical Guide to Feature-Driven Development**. Prentice Hall.
- PRESSMAN, R. S. (2002). **Engenharia de Software**. 5ª. Edição, McGrawHill.
- PROJECT MANAGEMENT INSTITUTE - PMI. (2004). **A guide to the project management body of knowledge**. Syba: PMI Publishing Division, 3ª. Ed. 2004. Disponível em: <<http://www.pmi.org>>. Acesso em 2008-03-10.
- PROJECT MANAGEMENT SOFTWARE. (2008). Disponível em: <<http://www.project-management-software.org/>>. Acesso em 2008-05-03.
- REHMAN, A. (2007). **Software Project Management Methodologies/Frameworks Dynamics “A Comparative Approach”**. In Proceedings of the IEEE International Conference on Information and Emerging Technologies (ICIET), pp. 1 - 5.
- ROUILLER, A. C. (2001). **Gerenciamento de Projetos de Software para Empresas de Pequeno Porte**. Doutorado em Ciência da Computação pela UFPE. Engenharia de Software e Qualidade de Software.
- RUP. (2003). **Rational Unified Process**, Version 2003.06.00.65, CD-ROM. Rational Software Corporation, Cupertino, California.
- SEI-2002-TR-011. (2002). **Capability Maturity Model Integration (CMMI)**, version 1.1, Continuous Representation.

- SCHNAIDER, L. R. C. (2003). **Planejamento da Alocação de Recursos Humanos em Ambientes e Desenvolvimento de Software Orientados à Organização**. Dissertação de Mestrado, COPPE / UFRJ.
- SCHWABER K. (2004). **Agile Project Management with Scrum**. Microsoft Press, Redmond.
- SHENHAR, A. J.; DVIR, D.; LEVY, O. (1997). **Mapping the dimensions of project success**. Project Management Journal, vol. 28, n. 2, pp. 5 - 13.
- SOMMERVILLE, I. (2007). **Software Engineering**. 8th ed. Harlow: Addison Wesley.
- STRUTS. (2008). Disponível em: <<http://struts.apache.org/>>. Visitado em 2008-09-02.
- TENSTEP. (2008). Disponível em: <<http://www.tenstep.com>>. Visitado em 2008-12-08.
- THE STANDISH GROUP. (1994). **The Chaos Report**. Disponível em: <<http://www.standishgroup.com/>>. Visitado em 2008-03-04.
- THE STANDISH GROUP. (2006). **The Chaos Report**. Disponível em: <<http://www.standishgroup.com/>>. Visitado em 2008-03-04.
- THOMSETT, R. (2002). **Radical Project Management**. New Jersey: Prentice Hall PTR.
- TOMCAT. (2008). Disponível em: <<http://tomcat.apache.org/>>. Visitado em 2008-09-02.
- TORTOISESVN. (2008). Disponível em: <<http://tortoisetsvn.tigris.org/>>. Visitado em 2008-09-10.

TRAPA V.; RAO S. (2006). **T3 - Tool for Monitoring Agile Development**. In Proceedings of the IEEE AGILE Conference, pp. 243 - 248.

VRIENS, C.; BARTO, R. (2008). **7 Years of Agile Management**. In Proceedings of the IEEE AGILE Conference, pp. 390 - 394.

XPLANNER. (2008). Disponível em: <<http://www.xplanner.org/index.html>>. Visitado em 2008-09-02.

WHITE D.; FORTUNE, J. (2002). **Current practice in project management – an empirical study**. International Journal of Project Management, vol. 20, n. 1, pp. 1 - 11.

WOHLIN, C.; RUNESON, P.; HÖST, M.; OHLSSON, M. C.; REGNELL B.; WESSLÉN, A. (2000). **Experimentation in Software Engineering: An Introduction**. Kluwer Academic Publishers. USA.