

Universidade Federal de Pernambuco  
Centro de Informática

**CIRO CARNEIRO COELHO**

**MAPS: UM MODELO DE ADAPTAÇÃO DE PROCESSOS DE  
SOFTWARE**

**ORIENTADOR**

Prof. Hermano Perrelli de Moura

Dissertação apresentada ao Centro  
de Informática da Universidade  
Federal de Pernambuco, como  
requisito parcial para a obtenção  
do grau de Mestre em Ciência da  
Computação

Recife, abril de 2003



# Agradecimentos

Em primeiro lugar, aos meus pais, Albanir e Célio, pelo apoio que sempre me deram. Se hoje tenho a possibilidade de estar concluindo um curso de mestrado, devo isso ao esforço e dedicação deles.

À minha irmã, Mariana, pelo carinho e por me manter informado das notícias de casa.

Ao Arthur e ao Joaquim, amigos e irmãos, que dividiram comigo a saudade de Fortaleza, além de muitas horas de trabalho, bons momentos de diversão e memoráveis partidas de Age of Empires e Counter-Strike!

Aos meus amigos, novos, de Recife, e antigos, de Fortaleza, pelo apoio e por me incentivarem a terminar logo a dissertação: “E aí, quando é a defesa?”.

Ao Professor Hermano, pela orientação no trabalho e pelas oportunidades oferecidas, que contribuíram para me tornar um profissional melhor.

Aos professores Alexandre Vasconcelos e Wilson de Pádua Filho, pela avaliação da dissertação.

Aos professores da pós-graduação do Centro de Informática da UFPE, pelo conhecimento compartilhado e por transformarem o CIn-UFPE em um centro de excelência, dando a oportunidade de uma formação de alto nível aos seus alunos.

A todas as pessoas que contribuíram para o estudo de caso e que não podem ter seus nomes citados aqui.



## Resumo

Como consequência do aumento da complexidade dos softwares e das maiores exigências do mercado, a busca de processos que venham organizar e melhorar o desenvolvimento de software tem crescido nos últimos anos. Apesar do grande número de processos disponíveis atualmente, não existe um processo de software único que se adeqüe a todas as situações. A eficiência de um processo varia de organização para organização e até entre os diferentes projetos de uma mesma organização. Uma solução comumente adotada é a definição de um processo padrão para a organização, em conjunto com diretrizes e critérios para a adaptação desse processo. A definição das diretrizes e dos critérios de adaptação é uma tarefa não-trivial, e vem sendo abordada de várias formas diferentes dentro da comunidade de Engenharia de Software. Este trabalho apresenta o Modelo de Adaptação de Processos de Software - MAPS, um modelo compatível com o Capability Maturity Model – CMM, e que auxilia a adaptação de um processo padrão para projetos específicos e promove o reuso e melhoria de processos de software. O MAPS é constituído por três componentes principais. A Base de Processos armazena o conhecimento adquirido sobre a utilização de processos em projetos passados. O Modelo de Caracterização de Projetos realiza uma comparação de projetos de software, permitindo identificar projetos semelhantes e facilitando, assim, o reuso de processos. O PConfig é responsável por configurar o processo padrão para projetos específicos com base nos artefatos do processo padrão. O MAPS objetiva a criação de uma base de processos adaptados, todos gerados a partir do processo padrão e adaptados às características específicas dos projetos, definindo, também, como esses processos adaptados podem ser reusados em projetos futuros de acordo com as características dos projetos. Para avaliar o MAPS, foi realizado um estudo de caso comparando os processos utilizados em dois projetos reais com os processos sugeridos pelo MAPS.

**Palavras-chave:** Processos de Desenvolvimento de Software, Adaptação de Processos de Software, Melhoria de Processos de Software.



# Abstract

The increasing software complexity and market challenges have caused a search for processes to organize and improve software development. Although there are many software processes currently available, there is not a single process that fits all organizations and projects. Process efficiency varies among organizations and even among different projects at the same organization. One of the strategies followed by the organizations is to define a standard process together with tailoring guidelines and criteria. The definition of tailoring guidelines and criteria is a non-trivial task that has been performed in many different ways by the Software Engineering community. In this work we introduce the MAPS<sup>1</sup> model, a CMM-compatible model that helps the tailoring of a standard process to specific projects and promotes software processes reuse and improvement. The MAPS model is composed by three main components. The Process Database contains information about processes used in previous projects. The Project Characterization Model compares software projects and identifies those that are similar in order to facilitate process reuse. PConfig, the third component, is responsible for standard software process configuration to specific projects based on process artifacts. The MAPS model aims to generate a database of tailored processes, all of them derived from the standard process and adapted to particular project characteristics. It also defines how these tailored processes can be reused in future projects according to the projects characteristics. The MAPS model was evaluated through a case study where the processes used in two real projects were compared with the processes suggested by MAPS.

**Keywords:** Software Development Processes, Software Processes Tailoring, Software Processes Improvement.

---

<sup>1</sup> MAPS stands for Software Processes Tailoring Model in Portuguese.





# Índice

<b><u>CAPÍTULO 1</u></b>	<b><u>INTRODUÇÃO</u></b>	<b><u>1</u></b>
<b>1.1</b>	<b>CONTEXTO E MOTIVAÇÃO</b>	<b>1</b>
1.1.1	METODOLOGIAS ÁGEIS VERSUS METODOLOGIAS TRADICIONAIS	4
<b>1.2</b>	<b>NOMENCLATURA E CONCEITOS</b>	<b>5</b>
<b>1.3</b>	<b>CONTRIBUIÇÕES ESPERADAS</b>	<b>6</b>
<b>1.4</b>	<b>ESTRUTURA DA DISSERTAÇÃO</b>	<b>6</b>
<b><u>CAPÍTULO 2</u></b>	<b><u>ADAPTAÇÃO DE PROCESSOS DE SOFTWARE</u></b>	<b><u>9</u></b>
<b>2.1</b>	<b>ADAPTAÇÃO DE PROCESSOS NO CMM</b>	<b>9</b>
2.1.1	DEFINIÇÃO E ADAPTAÇÃO DE PROCESSOS DE SOFTWARE NO CMM	10
2.1.2	CMM NÍVEL 3: ÁREAS-CHAVE PARA DEFINIÇÃO E ADAPTAÇÃO DE PROCESSOS DE SOFTWARE	13
<b>2.2</b>	<b>ADAPTAÇÃO DE PROCESSOS NAS METODOLOGIAS DE DESENVOLVIMENTO DE SOFTWARE</b>	<b>17</b>
2.2.1	CLASSIFICAÇÃO DOS PROCESSOS	17
2.2.2	ADAPTAÇÃO DE PROCESSOS NO OPEN	19
2.2.3	ADAPTAÇÃO DE PROCESSOS NAS METODOLOGIAS ÁGEIS	22
<b>2.3</b>	<b>ADAPTAÇÃO DE PROCESSOS NA INDÚSTRIA DE SOFTWARE</b>	<b>23</b>
2.3.1	ALCATEL	23
2.3.2	GODDARD SPACE FLIGHT CENTER	25
2.3.3	RAYTHEON	27
2.3.4	SPAWAR	29
2.3.5	BOMPREÇO	31
<b>2.4</b>	<b>CONSIDERAÇÕES</b>	<b>33</b>
<b><u>CAPÍTULO 3</u></b>	<b><u>RATIONAL UNIFIED PROCESS</u></b>	<b><u>35</u></b>
<b>3.1</b>	<b>VISÃO GERAL DO RUP</b>	<b>35</b>
3.1.1	ELEMENTOS DO RUP	36
3.1.2	BOAS PRÁTICAS DE DESENVOLVIMENTO DO RUP	38

<b>3.2 CONFIGURAÇÃO DO RUP: A DISCIPLINA AMBIENTE</b>	<b>40</b>
3.2.1 TIPO DE DESENVOLVIMENTO	42
3.2.2 TAMANHO DO PROJETO OU ESFORÇO DE DESENVOLVIMENTO	42
3.2.3 GRAU DE INOVAÇÃO	42
3.2.4 TIPO DE APLICAÇÃO	43
3.2.5 PROCESSO DE DESENVOLVIMENTO ATUAL	43
3.2.6 FATORES ORGANIZACIONAIS	43
3.2.7 COMPLEXIDADE TÉCNICA E GERENCIAL	43
<b>3.3 FLUXO DE PLANEJAMENTO E GERENCIAMENTO DO RUP</b>	<b>44</b>
<b>3.4 CONSIDERAÇÕES</b>	<b>47</b>

---

**CAPÍTULO 4      MODELO DE ADAPTAÇÃO DE PROCESSOS DE SOFTWARE** **49**

---

<b>4.1 VISÃO GERAL DO MODELO</b>	<b>49</b>
<b>4.2 ESCOPO DO TRABALHO</b>	<b>55</b>
<b>4.3 ESTRATÉGIAS DE IMPLANTAÇÃO DO MAPS</b>	<b>57</b>
<b>4.4 AUTOMAÇÃO DO PROCESSO</b>	<b>58</b>
<b>4.5 CONSIDERAÇÕES</b>	<b>59</b>

---

**CAPÍTULO 5      COMPARAÇÃO DE PROJETOS E REUSO DE PROCESSOS** **61**

---

<b>5.1 MODELO DE CARACTERIZAÇÃO DE PROJETOS</b>	<b>62</b>
5.1.1 DEFINIÇÃO DAS CARACTERÍSTICAS	63
5.1.2 ANÁLISE DAS CARACTERÍSTICAS	64
5.1.3 MÉTODO DE COMPARAÇÃO	76
5.1.4 ESTRATÉGIA RECOMENDADA PARA REUSO DE PROCESSOS	77
5.1.5 EXTENSÃO E ADAPTAÇÃO DO MODELO	78
<b>5.2 BASE DE PROCESSOS</b>	<b>79</b>
5.2.1 ESTRUTURA DA BASE DE PROCESSOS	80
5.2.2 REUSO DE PROCESSOS	83
5.2.3 AVALIAÇÃO DO PROCESSO	83
<b>5.3 CONSIDERAÇÕES</b>	<b>84</b>

**CAPÍTULO 6      PCONFIG: UM PROCESSO PARA CONFIGURAÇÃO**  
**DE PROCESSOS      87**

---

<b>6.1</b>	<b>VISÃO GERAL</b>	<b>87</b>
<b>6.2</b>	<b>CONCEITOS E MÉTODOS UTILIZADOS</b>	<b>89</b>
<b>6.3</b>	<b>O PROCESSO DE CONFIGURAÇÃO DE PROCESSOS</b>	<b>90</b>
<b>6.4</b>	<b>ARTEFATOS DO RUP</b>	<b>92</b>
6.4.1	PLANO DE DESENVOLVIMENTO DE SOFTWARE	93
6.4.2	ESTUDO DE VIABILIDADE	95
6.4.3	PLANO DE ITERAÇÃO	95
6.4.4	AVALIAÇÃO DA ITERAÇÃO	96
6.4.5	AVALIAÇÃO DE STATUS	96
6.4.6	PLANO DE RESOLUÇÃO DE PROBLEMAS	97
6.4.7	PLANO DE GERENCIAMENTO DE RISCOS	97
6.4.8	LISTA DE RISCOS	98
6.4.9	ORDEM DE TRABALHO	98
6.4.10	PLANO DE ACEITAÇÃO DO PRODUTO	99
6.4.11	PLANO DE GARANTIA DA QUALIDADE	99
6.4.12	PLANO DE MEDIÇÕES	100
6.4.13	REGISTRO DE REVISÃO	101
6.4.14	LISTA DE PENDÊNCIAS	102
6.4.15	MEDIÇÕES DO PROJETO	102
<b>6.5</b>	<b>RELAÇÃO ENTRE CARACTERÍSTICAS E ARTEFATOS</b>	<b>102</b>
<b>6.6</b>	<b>DIRETRIZES PARA A IMPLEMENTAÇÃO DO PCONFIG</b>	<b>109</b>
<b>6.7</b>	<b>CONSIDERAÇÕES</b>	<b>113</b>

**CAPÍTULO 7      AVALIAÇÃO DO MAPS**  
**115**

---

<b>7.1</b>	<b>OBJETIVOS</b>	<b>115</b>
<b>7.2</b>	<b>ABORDAGEM UTILIZADA</b>	<b>116</b>
<b>7.3</b>	<b>AVALIAÇÃO REALIZADA</b>	<b>119</b>
7.3.1	PROJETO A	119
7.3.2	PROJETO B	126
<b>7.4</b>	<b>ANÁLISE DOS RESULTADOS</b>	<b>135</b>

<b><u>CAPÍTULO 8</u></b>	<b><u>CONCLUSÕES</u></b>	<b><u>137</u></b>
<b>8.1</b>	<b>CONSIDERAÇÕES GERAIS E PRINCIPAIS CONTRIBUIÇÕES</b>	<b>137</b>
<b>8.2</b>	<b>DIFICULDADES ENCONTRADAS</b>	<b>138</b>
8.2.1	COMPLEXIDADE DO PROCESSO DE SOFTWARE	138
8.2.2	ESTUDO DE CASO	139
<b>8.3</b>	<b>TRABALHOS RELACIONADOS</b>	<b>140</b>
<b>8.4</b>	<b>TRABALHOS FUTUROS</b>	<b>142</b>
8.4.1	DETALHAMENTO DE OUTRAS DISCIPLINAS DO PROCESSO	142
8.4.2	DESENVOLVIMENTO DO PCONFIG PARA OUTROS PROCESSOS	142
8.4.3	AUTOMAÇÃO DA COMPARAÇÃO DE PROJETOS	143
8.4.4	AVALIAÇÃO DO MAPS	143
8.4.5	INTEGRAÇÃO COM OUTROS TRABALHOS	143
8.4.6	COMPARAÇÃO DE PROJETOS E REUSO DE PROCESSOS BASEADOS EM ARTEFATOS	144
<b>8.5</b>	<b>CONSIDERAÇÕES FINAIS</b>	<b>144</b>
<b><u>REFERÊNCIAS</u></b>		<b><u>147</u></b>
<b><u>APÊNDICE A</u></b>	<b><u>CARACTERIZAÇÃO DO PROJETO</u></b>	<b><u>153</u></b>
<b><u>APÊNDICE B</u></b>	<b><u>AVALIAÇÃO PARCIAL DO PROJETO</u></b>	<b><u>155</u></b>
<b><u>APÊNDICE C</u></b>	<b><u>AVALIAÇÃO FINAL DO PROJETO</u></b>	<b><u>157</u></b>
<b><u>ÍNDICE REMISSIVO</u></b>		<b><u>161</u></b>

## Lista de Figuras

Figura 1-1: Probabilidade de sucesso de acordo com o peso do processo utilizado no projeto.....	3
Figura 2-1: Estrutura de processos de software do CMM .....	11
Figura 2-2: Processo de adaptação de processos de software do CMM.....	12
Figura 2-3: Funcionamento do OPEN.....	20
Figura 2-4: Adaptação do OPEN.....	21
Figura 2-5: Classificação de projetos da metodologia Crystal .....	22
Figura 2-6: Elementos do <i>framework</i> de processos do GSFC .....	25
Figura 2-7: Matriz para adaptação de processos .....	26
Figura 2-8: Estrutura para processos de software da Raytheon .....	28
Figura 3-1: Visão geral do RUP .....	38
Figura 3-2: Complexidade do projeto <i>versus</i> nível de cerimônia do processo.....	44
Figura 3-3: Fluxo de Planejamento e Gerenciamento .....	46
Figura 4-1: Modelo de Adaptação de Processos de Software: entradas e saída.....	50
Figura 4-2: Modelo de Adaptação de Processos de Software (MAPS).....	53
Figura 5-1: Modelo de classes da Base de Processos.....	81
Figura 5-2: Modelo de classes modificado da Base de Processos.....	82
Figura 6-1: Artefatos de Planejamento e Gerenciamento do RUP.....	93



## Lista de Tabelas

Tabela 4-1: Estratégias para implantação do MAPS em uma organização.....	58
Tabela 5-1: Características de P&G de acordo com o tamanho da equipe.....	65
Tabela 6-1: Níveis da característica tamanho da equipe versus artefatos .....	91
Tabela 6-2: Coluna da matriz correspondente à característica tamanho da equipe do projeto.....	91
Tabela 6-3: Artefatos <i>versus</i> tamanho da equipe.....	104
Tabela 6-4: Artefatos <i>versus</i> experiência da equipe.....	105
Tabela 6-5: Artefatos <i>versus</i> distribuição geográfica da equipe.....	106
Tabela 6-6: Artefatos <i>versus</i> criticidade do software .....	107
Tabela 6-7: Artefatos <i>versus</i> tamanho do projeto.....	108
Tabela 6-8: Interdependência entre artefatos de Planejamento e Gerenciamento.....	110
Tabela 6-9: Artefatos de Planejamento e Gerenciamento que impactam outras disciplinas .....	111
Tabela 6-10: Artefatos de outras disciplinas que impactam Planejamento e Gerenciamento.....	111
Tabela 7-1: Processo utilizado no Projeto A .....	120
Tabela 7-2: Processo sugerido pelo MAPS para o Projeto A, sem características restritivas .....	122
Tabela 7-3: Processo sugerido pelo MAPS para o Projeto A.....	123
Tabela 7-4: Avaliação dos artefatos utilizados no Projeto A .....	124
Tabela 7-5: Avaliação dos artefatos não utilizados no Projeto A.....	124
Tabela 7-6: Processo utilizado no Projeto B .....	127
Tabela 7-7: Comparação do Projeto A e do Projeto B quanto ao tamanho da equipe .	129
Tabela 7-8: Processo sugerido pelo MAPS para o Projeto B .....	130
Tabela 7-9: Processo sugerido pelo MAPS para o Projeto B, sem reuso e sem características restritivas.....	131
Tabela 7-10: Avaliação dos artefatos utilizados no Projeto B .....	132
Tabela 7-11: Avaliação dos artefatos não utilizados no Projeto B.....	132





---

# Capítulo 1

## Introdução

---

Este capítulo apresenta uma visão geral do trabalho e está estruturado da seguinte forma:

- A Seção 1.1 discorre sobre os fatores que motivaram o desenvolvimento deste trabalho, apresentando um breve panorama da área de processos de software e os problemas que serão atacados.
- A Seção 1.2 define a nomenclatura e alguns conceitos que serão utilizados ao longo do trabalho.
- A Seção 1.3 lista as principais contribuições esperadas do trabalho.
- A Seção 1.4 apresenta a estrutura, em capítulos, do restante do trabalho.

### 1.1 Contexto e Motivação

A indústria do software vem experimentando um grande crescimento nas últimas décadas. Hoje, o software está presente na vida de praticamente todas as pessoas, seja através do uso de computadores, sistemas de automação comercial e industrial ou software embutido em eletrodomésticos e telefones celulares. Duas das principais conseqüências desse crescimento são o aumento da complexidade do software e as exigências cada vez maiores do mercado. É exigido das empresas de software que os sistemas sejam desenvolvidos com prazo e custo determinados e obedeçam a padrões de qualidade. Para atender essas exigências, tornou-se necessário investir no processo de desenvolvimento de software, já que é cada vez mais evidente a correlação entre a qualidade do produto de software desenvolvido e o processo de desenvolvimento adotado [1].

Um processo de desenvolvimento de software pode ser definido como “um conjunto de atividades, métodos, práticas e transformações que as pessoas empregam para desenvolver e manter software e produtos associados (por exemplo, planos de

projeto, documentos de projeto, projeto de software, código, casos de teste e manual do usuário)” [2]. Um processo de software tem por objetivo final possibilitar o desenvolvimento de software com qualidade e obedecendo a prazo e orçamento determinados. Através da adoção de processos, as organizações tentam criar estruturas que facilitem o desenvolvimento e garantam a qualidade do produto, fazendo com que o sucesso de um projeto não dependa apenas do esforço e talento dos membros das equipes de desenvolvimento.

Possuir um processo de desenvolvimento de software definido e padronizado, chamado comumente de processo padrão, apresenta uma série de vantagens para a organização:

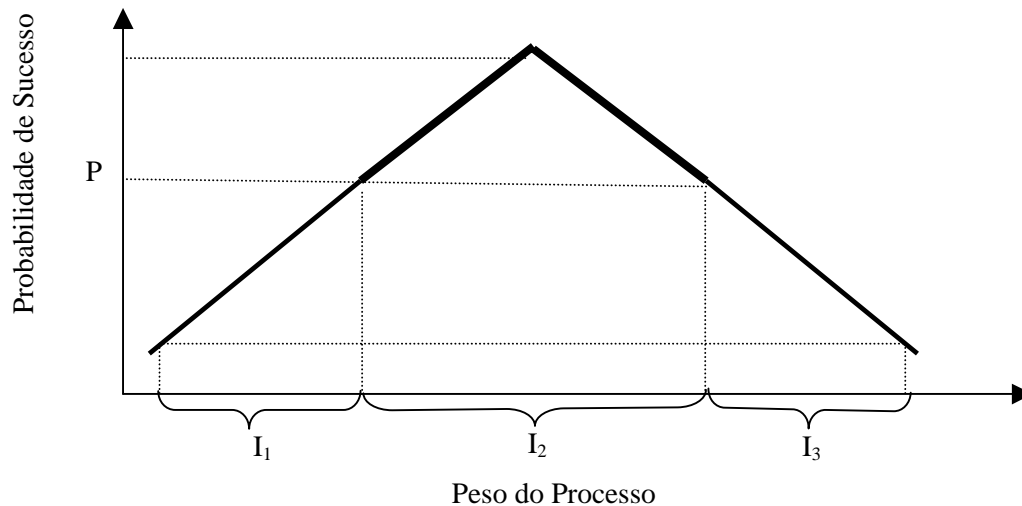
- Redução dos problemas relacionados a treinamento, revisões e suporte de ferramentas [3];
- Experiências adquiridas em cada projeto podem ser incorporadas ao processo padrão, contribuindo para sua melhoria [3];
- Um processo padronizado facilita medições de processo e qualidade [3];
- Maior facilidade na definição de processos para projetos específicos, já que muitas das decisões necessárias para realizar essa definição já foram tomadas no processo padrão [3];
- Maior facilidade de comunicação entre os membros da equipe [4];
- Maior consistência dos artefatos produzidos [4].

A demanda por processos de software fez surgir inúmeros processos diferentes (RUP [5, 6], OPEN [7], PSP [8]/TSP [9], XP [10] etc) que podem ser adotados, diretamente ou com modificações, como processo padrão de uma organização. Uma outra alternativa é a definição de um processo próprio, através de consultorias ou de um grupo de processos de software interno.

Qualquer que seja a estratégia adotada, a simples definição de um processo padrão não é suficiente, já que não existe um processo único que seja adequado a todas as situações. A eficiência de um processo varia de organização para organização e até entre os diferentes projetos de uma mesma organização. As tentativas de uniformizar o desenvolvimento através de um processo padrão que seja utilizado, sem qualquer modificação, em todos os projetos da organização, raramente obtêm resultados satisfatórios devido à incapacidade do processo de adaptar-se à grande diversidade de projetos de software. Essas tentativas de uniformização muitas vezes decorrem de

razões culturais, de acomodação em relação à simplicidade da existência de um processo único e imutável. A falta de flexibilidade do processo padrão leva, como consequência imediata, à utilização de processos inadequados em projetos.

A utilização de um processo inadequado pode causar o fracasso de um projeto. Um processo excessivamente formal causa um *overhead* desnecessário de atividades, tornando o desenvolvimento lento e caro e, conseqüentemente, diminuindo a produtividade e a competitividade da organização. Segundo Cockburn [11], um aumento relativamente pequeno no peso do processo causa um aumento relativamente grande no esforço de desenvolvimento. Utilizar um processo muito informal, por outro lado, pode ter como consequência a perda de controle sobre o projeto, tornando impossível garantir a qualidade do produto.



**Figura 1-1:** Probabilidade de sucesso de acordo com o peso do processo utilizado no projeto

A Figura 1-1 ilustra como a utilização de um processo inadequado pode influenciar a probabilidade de sucesso de um projeto. Na figura, o ponto P representa um nível aceitável de probabilidade de sucesso. O intervalo I<sub>1</sub>, no eixo Peso do Processo, representa os projetos que apresentam uma baixa probabilidade de sucesso por causa de um processo muito leve. Nesse caso, os projetos são mal sucedidos porque existe pouco planejamento e controle sobre as atividades realizadas. No intervalo I<sub>3</sub> estão compreendidos projetos com baixa probabilidade de sucesso por conta da utilização de um processo excessivamente burocrático. Nesse caso, os projetos falham

porque a produtividade é prejudicada pelo grande esforço despendido em atividades desnecessárias. O intervalo  $I_2$  representa os projetos que utilizam um processo de desenvolvimento adequado e que, conseqüentemente, possuem uma maior probabilidade de sucesso, acima do nível mínimo estabelecido. O problema passa a ser, então, como identificar o melhor processo a ser utilizado em cada projeto.

Descobrir o processo ideal, com o grau de formalismo adequado para cada projeto, não é uma tarefa trivial, já que o processo sofre influência de vários fatores, que tanto podem ser fatores ligados a características da organização desenvolvedora, como podem ser fatores ligados às características de um projeto específico.

Para capturar as características da organização em um processo de software, costuma-se adotar um processo padrão de desenvolvimento. Esse processo padrão deve ser adaptado para se adequar às características de cada projeto da organização. Essa estratégia, estabelecimento de um processo padrão e posterior adaptação para projetos específicos, tem sido largamente utilizada, tanto como base para pesquisas acadêmicas como na indústria de software [1, 12, 13, 14, 15, 16, 17].

No presente trabalho, será analisada a influência das características dos projetos no processo de desenvolvimento e será apresentado um modelo para a adaptação de um processo padrão para um projeto específico com base nessas características. Por se tratar de uma tarefa complexa, a análise das características dos projetos será focada nas características que causam maior impacto no processo de planejamento e gerenciamento de projetos. Apesar de tratar, com detalhes, apenas de uma disciplina específica do processo de software, o modelo definido provê toda a estrutura necessária para a futura inclusão das demais disciplinas do processo de software.

A relevância deste trabalho está ligada ao fato de que a qualidade do software produzido e a eficiência e eficácia do desenvolvimento estão diretamente ligadas à qualidade do processo adotado. Assim, utilizando o processo mais adequado ao projeto, pode-se conseguir melhorias no desenvolvimento e na qualidade do produto.

### **1.1.1 Metodologias Ágeis versus Metodologias Tradicionais**

Atualmente, um dos temas mais interessantes na comunidade de Engenharia de Software, particularmente da área de processos de desenvolvimento, é o debate entre os defensores das metodologias tradicionais, baseadas em planos, e os defensores das metodologias ágeis, mais informais.

Em seu manifesto [18], o grupo de defensores das metodologias ágeis defende que deve ser dada maior importância às pessoas da equipe de desenvolvimento e à interação entre elas, aos artefatos executáveis, à colaboração dos clientes no desenvolvimento e à capacidade de responder às mudanças. Esses itens devem, segundo esse grupo, ser mais importantes que processos, ferramentas, documentação, negociação de contratos e planos. Em outras palavras, defende-se um processo que seja mais adaptativo e menos preditivo.

As idéias básicas das metodologias ágeis estão descritas em várias publicações [19, 20, 21] e muitas metodologias ágeis têm surgido e sido utilizadas [22], com destaque para eXtreme Programming [10].

Apesar de algumas discordâncias sobre a aplicabilidade e eficiência de alguns conceitos das metodologias ágeis [23, 24], a tendência parece ser a aceitação de que essas metodologias possuem limitações [25], mas podem ser aplicadas, total ou parcialmente, em muitas situações. Tem-se presenciado, inclusive, uma “união de forças” entre metodologias tradicionais e ágeis na tentativa de alcançar uma solução intermediária que contemple idéias das duas correntes [26].

Neste trabalho é defendida a idéia de que as duas correntes são complementares, e não excludentes. A aplicabilidade dos conceitos de desenvolvimento ágil e de desenvolvimento dirigido por planos depende do contexto em que se está trabalhando. Essa idéia é compartilhada por vários pesquisadores, metodologistas e desenvolvedores como, por exemplo, [11, 24, 27, 28].

## 1.2 Nomenclatura e Conceitos

Neste trabalho, não é feita qualquer distinção entre “processo” e “metodologia”, embora esses termos não sejam totalmente sinônimos. O termo “adaptação” será utilizado como correspondente do termo “*tailoring*”, em inglês, de acordo com a recomendação do Subcomitê de Software da Associação Brasileira de Normas Técnicas (ABNT Software), através da sua Comissão de Estudos em Gerência do Ciclo de Vida do Software [29]. O RUP, que será utilizado como exemplo de processo padrão a ser adaptado, trata da sua adaptação como “*configuration*”. Assim, a palavra “configuração” será utilizada como sinônimo de “adaptação”.

O “tamanho” de um processo será tratado como o número de elementos desse processo, incluindo artefatos, atividades e papéis produzidos, executados e

desempenhados pela equipe de desenvolvimento. A “densidade” de um processo significa os níveis de detalhes e consistência requeridos para elementos do processo. O “peso” do processo é definido, conceitualmente, como o tamanho multiplicado pela densidade do processo [11].

Um processo padrão será considerado como um processo concreto e funcional que pode, inclusive, ser utilizado sem nenhuma modificação em um projeto. O processo padrão, tal como é tratado neste trabalho, refere-se a um processo abrangente que constitui uma base de conhecimento que pode ser utilizada total ou parcialmente em um projeto. A escolha de quais elementos (atividades e passos de atividades, artefatos, papéis desempenhados etc) farão parte de uma determinada aplicação do processo padrão será referenciada como adaptação do processo e o processo gerado será tratado como processo adaptado ou processo específico do projeto. No contexto deste trabalho, não estará sendo considerada a situação em que o processo padrão é um processo em um nível de abstração mais alto que deve ser adaptado para cada situação.

Caso o processo padrão seja especializado para vários tipos de desenvolvimento, como previsto no modelo sugerido em [1], a adaptação será feita a partir de cada processo especializado.

### **1.3 Contribuições Esperadas**

As principais contribuições do trabalho são:

- Análise do impacto das características dos projetos nos processos de desenvolvimento, em especial na disciplina Planejamento e Gerenciamento.
- O MAPS, Modelo de Adaptação de Processos de Software, modelo que permite a adaptação de um processo padrão para processos específicos de uma forma sistemática e orientada a projetos, buscando sempre facilitar o reuso de processos.
- O Modelo de Caracterização de Projetos, que realiza a comparação entre projetos de software com base nas suas características.

### **1.4 Estrutura da Dissertação**

A dissertação está estruturada da seguinte forma:

- O presente capítulo contém uma introdução ao trabalho realizado.

- O Capítulo 2 descreve a utilização de métodos de adaptação de processos de software na indústria de software, seja através de modelos de maturidade, de metodologias de desenvolvimento ou de iniciativas específicas de empresas desenvolvedoras de software.
- O Capítulo 3 descreve o RUP (Rational Unified Process), que será utilizado aqui como um exemplo de processo padrão.
- O Capítulo 4 apresenta uma visão geral do MAPS (Modelo de Adaptação de Processos de Software), aqui proposto, explicando seus principais conceitos.
- O Capítulo 5 explica como o MAPS realiza a comparação de projetos e o reuso de processos de software, descrevendo os componentes responsáveis por essa tarefa: o Modelo de Caracterização de Projetos e a Base de Processos.
- O Capítulo 6 apresenta um outro componente do MAPS, PConfig, que é um processo para configuração de processos de software com base nos artefatos do processo padrão. Neste trabalho, a implementação do PConfig utiliza o RUP como processo padrão.
- O Capítulo 7 descreve o estudo de caso realizado para validar o MAPS.
- O Capítulo 8 apresenta as conclusões do trabalho e uma descrição de possíveis trabalhos futuros.





---

## Capítulo 2

# Adaptação de Processos de Software

---

Este capítulo trata da utilização de métodos de adaptação de processos, seja como parte de modelos de maturidade, como atividades presentes em metodologias de desenvolvimento de software ou como experiência prática em organizações desenvolvedoras de software.

O capítulo está organizado da seguinte forma:

- A Seção 2.1 trata dos aspectos relacionados à definição e adaptação de processos de software contidos no CMM, focando nos aspectos relacionados ao nível 3 do CMM, nível que trata da adaptação de processos.
- A Seção 2.2 descreve como a adaptação de processos é tratada em algumas metodologias de desenvolvimento de software.
- A Seção 2.3 descreve algumas experiências práticas de empresas desenvolvedoras de software que realizam adaptação de processos.
- A Seção 2.4 apresenta uma breve conclusão do capítulo.

## 2.1 Adaptação de Processos no CMM

Em novembro de 1986, o Software Engineering Institute (SEI), da Universidade de Carnegie Mellon, iniciou o desenvolvimento de um *framework* de maturidade de processos de software com objetivo de ajudar as organizações na melhoria de seus processos de software. Em 1987 foi apresentada uma breve descrição desse *framework* e, quatro anos mais tarde, em 1991, foi lançado um modelo de maturidade de processos de software, uma evolução do *framework*, chamado de Capability Maturity Model for Software 1.0 (CMM 1.0). Em 1993, como resultado do *feedback* da comunidade de software, foi lançada a versão 1.1 do CMM [15], que serve de base para este trabalho.

O CMM permite à organização avaliar o seu nível de maturidade em termos de processos de software, identificar possíveis pontos de melhoria, estabelecer estratégias para essa melhoria e gerenciar de forma eficiente os processos de software.

O CMM apresenta cinco níveis de maturidade:

1. **Inicial:** o processo de software é caracterizado como *ad hoc* ou caótico. Poucos processos estão definidos e o sucesso de um projeto depende do esforço e talento individual dos membros da equipe.
2. **Repetível:** processos básicos de gerenciamento de projetos estão instituídos. A disciplina do processo de software é suficiente para que o sucesso de um projeto possa ser repetido em um projeto semelhante.
3. **Definido:** o processo de software está documentado e padronizado, ou seja, existe um processo de software padrão da organização. Todos os projetos usam uma versão adaptada do processo padrão.
4. **Gerenciado:** medições detalhadas do processo de software e da qualidade do produto são realizadas. O processo de software e os produtos desenvolvidos são quantitativamente entendidos e controlados.
5. **Otimizando:** processo de melhoria contínua através da análise quantitativa do processo e de projetos-pilotos para novas tecnologias e idéias.

Neste trabalho, será abordado apenas o nível 3 do CMM, mais especificamente os tópicos referentes à definição e adaptação do processo de software, descritos em [15].

### 2.1.1 Definição e Adaptação de Processos de Software no CMM

De acordo com o CMM, uma organização, para atingir o nível 3 de maturidade, deve possuir um processo padrão, chamado de Processo de Software Padrão da Organização<sup>2</sup> (PSPO). Esse processo padrão descreve o processo básico que deve guiar o desenvolvimento de todos os projetos da organização. O PSPO define a arquitetura do processo (descrição de alto nível), os elementos do processo e os relacionamentos entre esses elementos.

Apesar de definir uma forma padronizada de desenvolvimento, o PSPO é

---

<sup>2</sup> No original, em inglês, *Organization's Standard Software Process*, originalmente traduzido em [30].

descrito de uma forma abstrata demais para ser usado diretamente em um projeto. Por esse motivo, fazem-se necessários critérios e diretrizes para adaptar o PSPO para cada projeto. Esses critérios e diretrizes devem garantir a coerência entre o processo padrão e os processos adaptados, informando que elementos do processo devem ser considerados na adaptação e que elementos não podem ser adaptados. O processo resultante dessa adaptação recebe o nome de Projeto de Software Definido do Projeto<sup>3</sup> (PSDP). O PSDP deve conter os estágios do ciclo de desenvolvimento, as atividades e tarefas que devem ser realizadas, os papéis (quem realiza cada tarefa), e os artefatos que serão produzidos.

A partir do PSDP, deve ser desenvolvido um Plano de Desenvolvimento de Software<sup>4</sup> (PDS). Esse plano identifica que indivíduos desempenharão os papéis previstos no processo, descreve de forma precisa os artefatos que serão produzidos e define o cronograma de execução das atividades e tarefas.

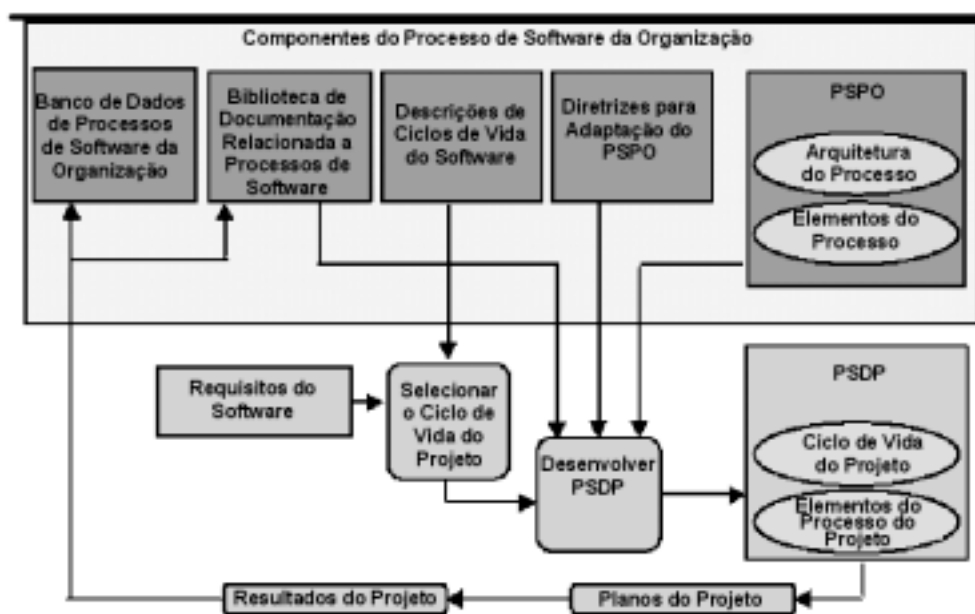


Figura 2-1: Estrutura de processos de software do CMM (traduzido de [15])

Para armazenar e disponibilizar informações relativas ao processo de software, existem o Banco de Dados de Processos de Software da Organização<sup>5</sup> e a Biblioteca de Documentação Relacionada a Processos de Software<sup>6</sup>. O Banco de Dados de Processos

<sup>3</sup> No original, em inglês, *Project's Defined Software Process*, originalmente traduzido em [30].

<sup>4</sup> No original, em inglês, *Software Development Plan*, originalmente traduzido em [30].

<sup>5</sup> No original, em inglês, *Organization's Software Process Database*, originalmente traduzido em [30].

<sup>6</sup> No original, em inglês, *Library of Software Process-related Documentation*, traduzido em [30].

de Software da Organização armazena processos e produtos de software como, por exemplo, dados estimados e dados reais, dados de produtividade e número de defeitos encontrados. A Biblioteca de Documentação Relacionada a Processos de Software armazena documentos como PSDPs, padrões, procedimentos, planos de desenvolvimento e material de treinamento.

A Figura 2-1 mostra a estrutura geral para processos de software sugerida pelo CMM. A Figura 2-2 mostra o modelo específico de adaptação de processos de software do CMM.

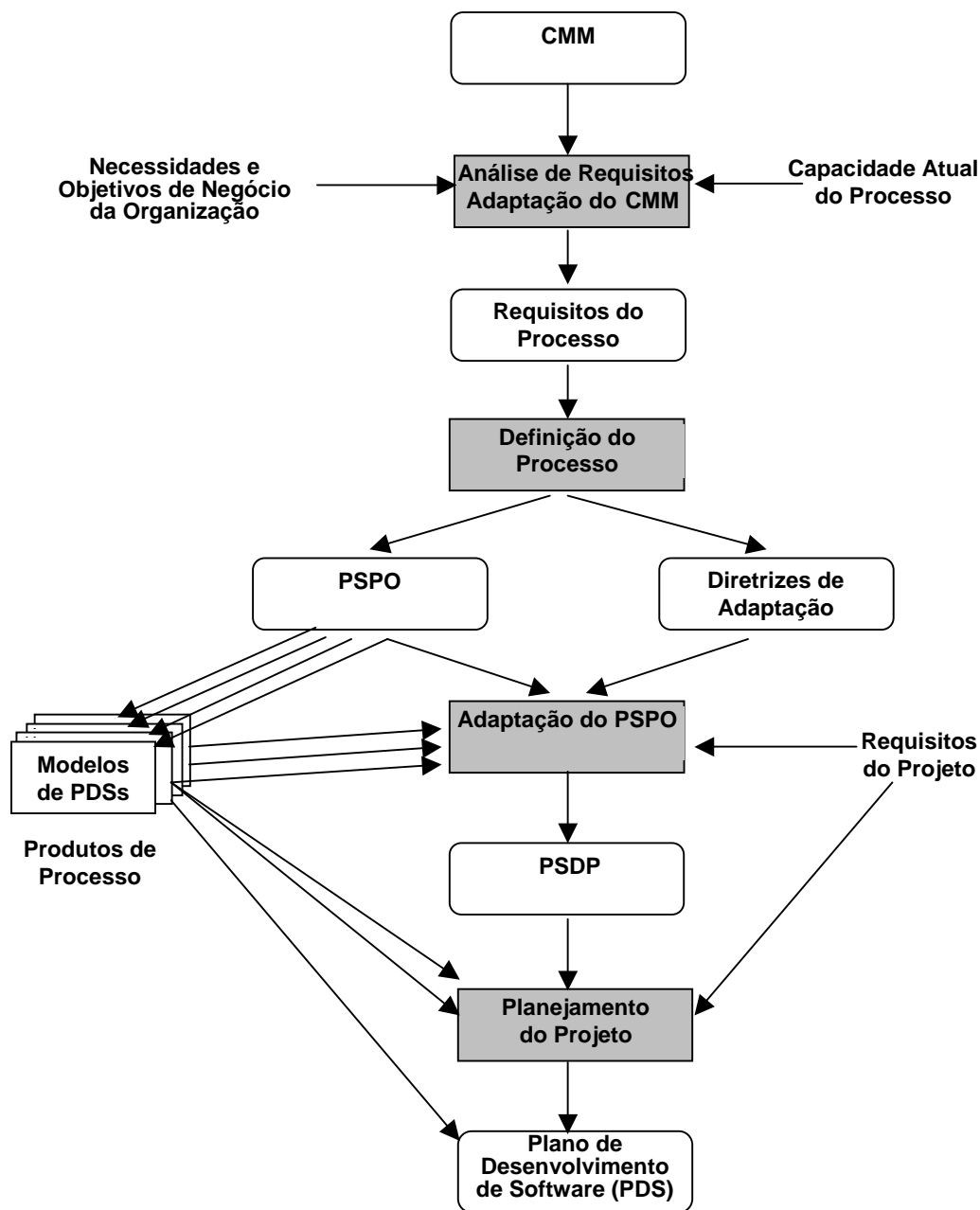


Figura 2-2: Processo de adaptação de processos de software do CMM (traduzido de [31])

## 2.1.2 CMM Nível 3: Áreas-chave para Definição e Adaptação de Processos de Software

Os objetivos que devem ser alcançados para se atingir determinado nível do CMM estão agrupados em áreas-chave, chamadas KPAs (*Key Process Areas*). Cada KPA contém práticas que devem ser obedecidas, capacidades necessárias, atividades que devem ser realizadas, medições e análises e verificações de implementação.

As KPAs relacionadas com processos de software que fazem parte do nível 3 do CMM são:

### **KPA Foco no Processo da Organização**

Objetivos:

- Coordenação das atividades de desenvolvimento e melhoria do processo de software.
- Os pontos fortes e fracos dos processos de software utilizados são identificados relativamente a um padrão de processo.
- As atividades de desenvolvimento e melhoria do processo de software a nível organizacional são planejadas.

Práticas:

- A organização segue uma política organizacional escrita para coordenar as atividades de desenvolvimento e melhoria do processo de software.
- A alta gerência apóia as atividades de desenvolvimento e melhoria do processo de software.
- A alta gerência acompanha as atividades de desenvolvimento e melhoria do processo de software.

Capacidades:

- Existe um grupo responsável pelas atividades relativas ao processo de software da organização.
- Recursos e financiamento adequados são destinados às atividades relativas ao processo de software da organização.

- Os membros do grupo responsável pelas atividades relativas ao processo de software da organização recebem treinamento adequado para realizar essas atividades.
- Os membros do grupo de engenharia de software e de outros grupos relacionados ao desenvolvimento de software recebem orientação sobre as atividades relativas ao processo de software da organização e sobre seus papéis nessas atividades.

#### Atividades

- O processo de software é avaliado periodicamente e planos de ação são desenvolvidos de acordo com os resultados da avaliação.
- A organização desenvolve e mantém um plano para as atividades de desenvolvimento e melhoria do seu processo de software.
- As atividades (da organização e de projeto) de desenvolvimento e melhoria do processo de software são coordenadas no nível organizacional.
- O uso do banco de dados de processos da organização é coordenado no nível organizacional.
- Novos processos, métodos e ferramentas de uso limitado na organização são monitorados, avaliados e, se preciso, transferidos para outra parte da organização.
- Os treinamentos para os processos organizacionais e de projeto são coordenados no nível organizacional.
- Os grupos envolvidos na implementação dos processos de software são informados das atividades (organizacionais e de projeto) de desenvolvimento e melhoria dos processos de software.

#### Medição e Análise:

- Medições são feitas e utilizadas para determinar o estado das atividades de desenvolvimento e melhoria dos processos de software.

#### Verificações:

- As atividades de desenvolvimento e melhoria dos processos de software são revisadas pela alta gerência de forma periódica.

## **KPA Definição do Processo da Organização**

### Objetivos:

- Um processo de software padrão da organização (PSPO) é desenvolvido e mantido.
- Informações relativas ao uso do processo padrão pelos projetos de software são coletadas, revisadas e disponibilizadas.

### Práticas:

- A organização segue políticas escritas para desenvolver e manter um processo padrão de software e produtos associados.

### Capacidades:

- Recursos e financiamento adequado são destinados ao desenvolvimento e manutenção do processo padrão da organização e produtos associados.
- Os indivíduos que desenvolvem e mantêm o processo padrão da organização e produtos associados recebem treinamento para realizar essas atividades.

### Atividades:

- O processo padrão da organização é desenvolvido e mantido de acordo com um procedimento documentado.
- O processo padrão da organização é documentado de acordo com padrões organizacionais estabelecidos.
- Descrições dos ciclos de vida de software que são aprovados para o uso em projetos são documentadas e mantidas.
- Diretrizes e critérios de adaptação do processo padrão da organização são desenvolvidos e mantidos.
- Um banco de dados de processos da organização é instituído e mantido.
- Uma biblioteca de documentação relativa a processos é instituída e mantida.

### Medição e Análise:

- Medições são feitas para determinar o estado das atividades de definição do processo padrão da organização.

### Verificação:

- O grupo de garantia da qualidade de software faz revisões e auditorias das atividades e produtos relacionados ao desenvolvimento e manutenção do

processo padrão da organização e produtos associados, reportando os resultados.

## **KPA Gerenciamento Integrado de Software**

Objetivos:

- O processo de software definido do projeto (PSDP) é uma versão adaptada do PSPO.
- O projeto é planejado e gerenciado de acordo com o PSDP.

Práticas:

- O projeto segue uma política organizacional escrita segundo a qual o projeto de software deve ser planejado e gerenciado utilizando o PSPO e produtos associados

Capacidades:

- Recursos e financiamento adequados são destinados para o gerenciamento do processo de software utilizando o PSDP.
- Os indivíduos responsáveis pelo desenvolvimento do PSDP recebem treinamento para adaptar o PSPO e produtos associados.
- Os gerentes recebem treinamento para gerenciar os aspectos técnicos, administrativos e pessoais do projeto de software baseado no PSDP.

Atividades:

- O PSDP é desenvolvido através da adaptação do PSPO de acordo com um procedimento documentado.
- Cada PSDP é revisado de acordo com um procedimento documentado.
- O plano de desenvolvimento de software do projeto, que descreve o uso do PSDP, é desenvolvido e revisado de acordo com um procedimento documentado.
- O projeto de software é gerenciado de acordo com o PSDP.
- O banco de dados de processos da organização é utilizado para o planejamento e estimativas dos projetos de software.

Medição e Análise:



- Medições são feitas e utilizadas para determinar a eficiência das atividades de gerenciamento integrado de software.

## 2.2 Adaptação de Processos nas Metodologias de Desenvolvimento de Software

A adaptação de processos de software é tratada de forma diferente nos vários processos de desenvolvimento de software. Essa seção apresenta uma classificação de processos de software de acordo com sua forma de adaptação e descreve como algumas metodologias tratam a adaptação de processos.

### 2.2.1 Classificação dos Processos

Martin Fowler [32] elaborou uma classificação de processos de desenvolvimento de software quanto à forma de adaptação.

#### **Processos Concretos**

**Definição:** provêm um conjunto fixo de práticas a serem seguidas, permitindo pouca ou nenhuma variação;

**Exemplo:** XP [10];

**Pontos Fortes:** mais simples de entender e aplicar, já que deixam claro o que deve ser feito;

**Pontos Fracos:** não podem ser mudados ou, pelo menos, não existe uma forma clara e pré-definida para realizar essa mudança.

#### **Processos Adaptáveis**

**Definição:** trazem diretrizes explícitas para realizar a adaptação do processo;

**Exemplo:** OPEN [7];

**Pontos Fortes:** fornecem meios para identificar as possíveis variações do processo e quando essas variações são apropriadas;

**Pontos Fracos:** normalmente é difícil entender como realizar essa adaptação de forma correta. É preciso entender o processo básico e todas as suas variações antes de decidir o que fazer.

## ***Frameworks* de Processo**

**Definição:** uma variação dos processos adaptáveis. Têm como filosofia apresentar um processo o mais abrangente possível e adaptá-lo através da remoção de elementos desnecessários para uma situação específica. A principal característica dos *frameworks* de processo é que eles tentam englobar todos os elementos (artefatos, atividades etc) que possam ser úteis a todos os tipos de projeto. Isso não é verdade para os demais processos adaptáveis;

**Exemplo:** RUP [6];

**Pontos Fortes:** existe um único processo que deve ser aprendido e que pode ser aplicado às mais variadas situações;

**Pontos Fracos:** se for preciso um processo pequeno, primeiro tem que se entender todo o *framework* para depois decidir o que deve ser feito, o que significa uma carga de trabalho bem maior do que simplesmente adotar um processo do tamanho desejado.

## **Processos Filosóficos**

**Definição:** não definem atividades que devem ser realizadas ou artefatos que devem ser produzidos, definindo apenas uma filosofia de desenvolvimento a ser adotada;

**Exemplo:** ASD [33];

**Pontos Fortes:** são flexíveis por natureza, podendo ser aplicados a qualquer tipo de projeto sem a necessidade de definir regras de adaptação;

**Pontos Fracos:** como não definem claramente o que deve ser feito, são processos difíceis de seguir.

## **Conjunto de Melhores Práticas**

**Definição:** não é propriamente um processo de desenvolvimento, mas um agrupamento de práticas independentes umas das outras que podem ser aplicadas em qualquer processo;

**Exemplo:** PSP [8];

**Pontos Fortes:** diferentemente dos processos filosóficos, constituem formas concretas de como realizar tarefas e produzir artefatos;

**Pontos Fracos:** não são suficientes para guiar o desenvolvimento.

## 2.2.2 Adaptação de Processos no OPEN

O OPEN (Object-oriented Process, Environment and Notation) [7] foi desenvolvido pelo OPEN Consortium, uma organização que reúne desenvolvedores de software, pesquisadores, fabricantes de ferramentas *case* etc.

O OPEN consiste basicamente de um *framework* de processos, chamado de OPF (OPEN Process Framework), que contém um metamodelo a partir do qual instâncias do OPEN podem ser criadas para cada organização. Essas instâncias são criadas escolhendo atividades, tarefas e técnicas específicas.

Os componentes do OPF são divididos em cinco classes:

- **Produtos:** são componentes desenvolvidos pelo projeto. Podem ser programas executáveis, diagramas, classes, modelos e documentos em geral.
- **Linguagens:** são componentes usados para documentar Produtos. Uma Linguagem pode ser uma linguagem natural, estruturada, de modelagem (UML, por exemplo) ou linguagens de programação (Java, SQL, C++ etc).
- **Produtores:** são os componentes que desenvolvem Produtos. Os Produtores podem ser diretos (pessoas, papéis desempenhados por pessoas e ferramentas utilizadas por pessoas) ou indiretos (equipes de desenvolvimento e organizações).
- **Unidades de Trabalho:** são componentes que modelam as operações realizadas pelos Produtores durante o desenvolvimento de um Produto. As Unidades de Trabalho estão divididas em quatro classes: Tarefas (o que deve ser feito), Técnicas (como as Tarefas devem ser realizadas), Realização de Tarefa (que Produtor irá realizar que Tarefa) e Atividades (conjuntos de tarefas que descrevem de forma geral o que deve ser feito, quem deve fazer e que Produtos serão produzidos).
- **Estágios:** são intervalos de tempo ou momentos específicos que servem para organizar as Unidades de Trabalho. Ex: ciclos, fases, workflows, marcos de referência.

De forma resumida, pode-se dizer que um sistema desenvolvido com o OPEN segue os seguintes passos (Figura 2-3):

- O desenvolvimento é organizado em Estágios, que possuem uma série de diretrizes que descrevem passo a passo como o sistema deve ser desenvolvido;

- Os Produtores realizam uma série de Unidades de Trabalho para desenvolver Produtos;
- O trabalho realizado em uma Unidade de Trabalho é agrupado em Atividades, que descrevem de maneira geral o que deve ser feito;
- As Atividades são subdivididas em Tarefas, que serão realizadas por Produtores através da utilização de Técnicas, que dizem como as Tarefas devem ser realizadas. A essa relação entre Produtores e Tarefas dá-se o nome de Realização de Tarefas;
- Finalmente, os Produtos desenvolvidos são documentados utilizando Linguagens.

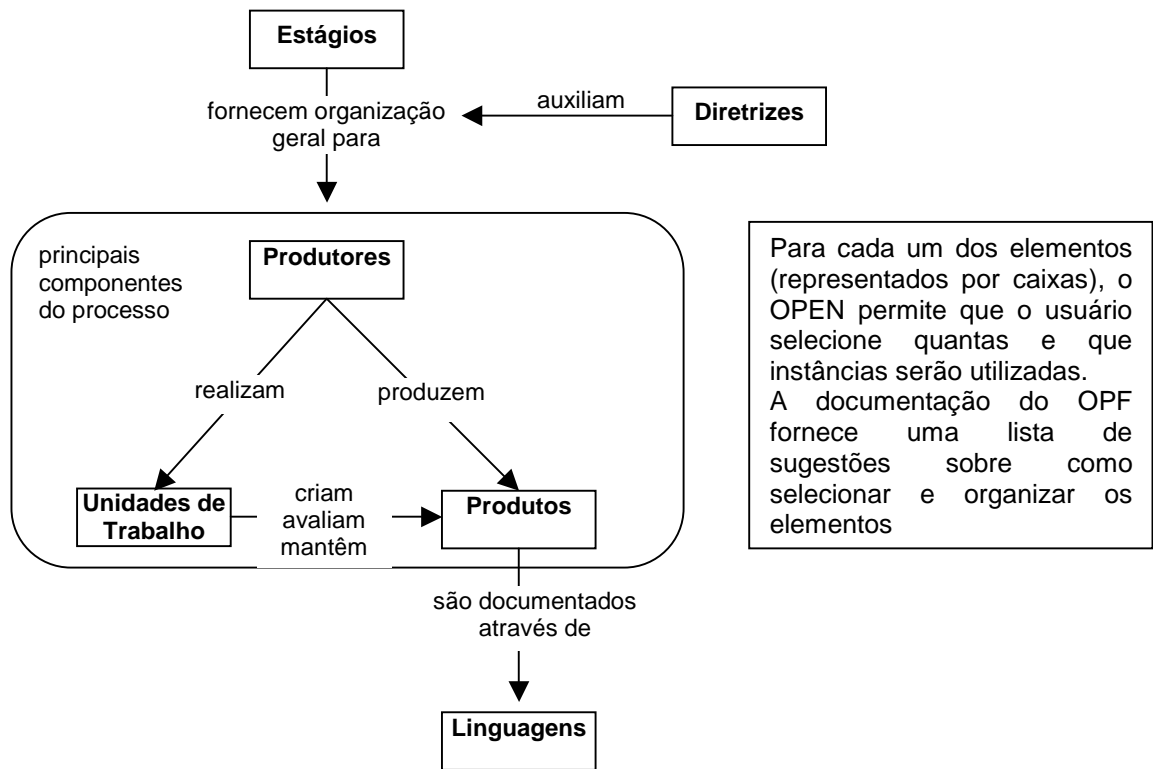
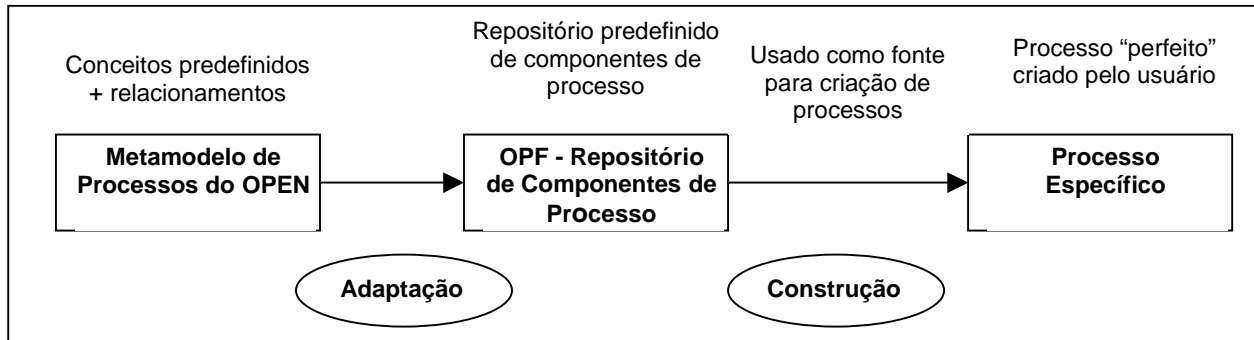


Figura 2-3: Funcionamento do OPEN (traduzido de [34])

O OPEN, ao contrário do RUP, descrito no Capítulo 3, não apresenta uma forma padrão de desenvolvimento. Os elementos do OPF devem ser selecionados de acordo com um metamodelo, composto de conceitos e relacionamentos predefinidos, e com as necessidades do projeto ou organização (Figura 2-4).



**Figura 2-4:** Adaptação do OPEN (traduzido de [34])

Para auxiliar a construção de processos, o OPEN fornece três tipos de diretrizes: construção, adaptação e extensão.

Diretrizes de construção servem para guiar os engenheiros de processos nas seguintes atividades:

- Selecionar os Produtos que serão desenvolvidos.
- Selecionar os Produtores (papéis, equipes e ferramentas) que irão desenvolver os Produtos.
- Selecionar as Unidades de Trabalho que serão realizadas.
- Alocar Tarefas, e Técnicas associadas, aos Produtores.
- Agrupar Tarefas em Atividades.
- Selecionar Estágios de desenvolvimento que darão uma visão geral das Unidades de Trabalho.

Após a construção do processo, durante o seu uso em um projeto real, pode ser necessário adaptar os elementos do processo. Isso significa realizar alterações nos elementos (incluir, excluir e alterar seções de um documento, excluir Produtos, mudar nomenclatura de elementos etc) de forma que eles se adequem às necessidades do projeto. Para auxiliar essa atividade, o OPEN fornece as diretrizes de adaptação. Vale observar que essa adaptação pode ocorrer em partes de elementos, como seções de um documento, ou seja, um Produto pode ser relevante e/ou adequado para um projeto sem que necessariamente todo o seu conteúdo seja relevante e/ou adequado.

As diretrizes de extensão são utilizadas para adicionar novos elementos ou novas classes de elementos ao OPF.

### 2.2.3 Adaptação de Processos nas Metodologias Ágeis

De forma geral, as metodologias ágeis não possuem diretrizes para adaptação. Isso se deve, sobretudo, ao fato de serem metodologias simples e pequenas, com um campo de aplicação mais restrito que as metodologias tradicionais. Assim, na maioria das metodologias ágeis, as variações ocorrem apenas de forma pontual. Pode-se realizar determinada atividade de uma maneira mais ou menos formal, sem, no entanto, modificar a estrutura da metodologia nem a seqüência de passos que deve ser seguida.

Existem trabalhos e discussões sobre propostas de variações de metodologias ágeis, especialmente XP [35, 36, 37, 38]. Essas propostas, entretanto, não estabelecem a criação de diretrizes de adaptação para essas metodologias, mas sim a criação de novas metodologias a partir das já existentes.

A metodologia Crystal [39], por suas características, constitui um caso a parte. Apesar de hoje ser mais referenciada como Crystal Clear, uma metodologia ágil, a idéia inicial da Crystal baseia-se no conceito de família de metodologias.

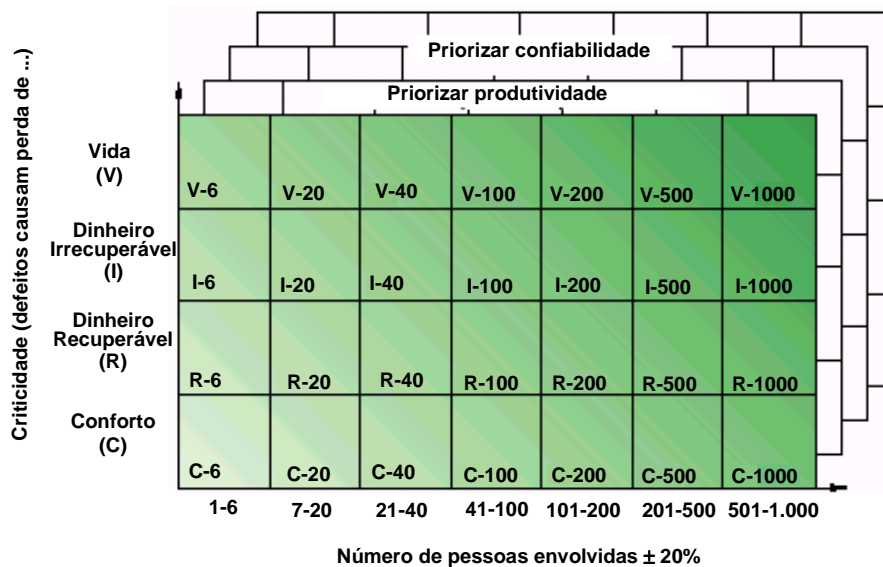


Figura 2-5: Classificação de projetos da metodologia Crystal (traduzido de [11])

Assim, para um dado conjunto de tipos de projeto, deve existir um membro da família Crystal adequado às necessidades específicas desses projetos. A complexidade da metodologia aumenta proporcionalmente ao aumento da complexidade do projeto, que é determinada em função do tamanho da equipe de desenvolvimento, da criticidade do software e das prioridades do projeto (Figura 2-5) [11]. A Crystal Clear é a

metodologia da família Crystal que foi desenvolvida para os projetos menos complexos. Aproveitando o movimento em prol das metodologias ágeis, a Crystal Clear foi desenvolvida e divulgada de maneira prioritária, em detrimento das outras metodologias da família. Isso acabou ofuscando a idéia original de “uma metodologia por projeto”.

## 2.3 Adaptação de Processos na Indústria de Software

As empresas cujas experiências são relatadas nesta Seção foram selecionadas tendo como critério a existência e disponibilidade de documentação sobre a estratégia de adaptação adotada, além de circunstâncias e métodos interessantes que pudessem enriquecer o relato das experiências.

### 2.3.1 Alcatel

No ano de 1998, a Alcatel<sup>7</sup>, após ter atingido o Nível 2 do CMM, iniciou um trabalho de reengenharia de seus processos de software [13]. A principal razão para essa reengenharia foi a necessidade de gerenciar a grande diversidade de processos de software da empresa.

A produção de software era feita em mais de 20 centros de desenvolvimento, por mais de 5000 pessoas. A base instalada de software se espalhava por mais de 60 países. As melhorias no processo de desenvolvimento eram frutos de iniciativas isoladas, que não chegavam a beneficiar a organização como um todo.

Assim, os objetivos da reengenharia de processos de software realizada eram:

- Facilitar a comunicação e interação das equipes virtuais através de um sistema de *workflow* integrado.
- Reduzir a sobrecarga causada pela replicação de atividades de definição e melhoria de processos.
- Estimular o aprendizado organizacional e evitar o isolamento de melhores práticas.
- Integrar *workflows* específicos de cada produto através de interfaces padronizadas e de uma gerência de configuração uniforme para elementos do processo e artefatos.
- Melhorar a eficiência do desenvolvimento através do uso de processos

---

<sup>7</sup> Site da Alcatel: [www.alcatel.com](http://www.alcatel.com).

padrões e das tecnologias e ferramentas que melhor suportem esses processos.

Devido à grande variedade de projetos de software existente na Alcatel, não era suficiente apenas definir processos padrões de desenvolvimento. Era necessário também estabelecer formas de adaptar esses processos para adequá-los às necessidades de cada projeto.

O primeiro passo para definir a política de adaptação de processos foi identificar quais elementos ou partes de processos deveriam variar de projeto para projeto e quais deveriam permanecer imutáveis. Em seguida foram identificados os seguintes critérios para a adoção de um processo específico:

- Tamanho do projeto, em termos de esforço (3 níveis).
- Tipo de produto (produto genérico, customização ou manutenção de produto, protótipo).
- Critérios específicos do produto (plataforma de desenvolvimento, linguagem de programação, parâmetros industriais).

Para garantir que os processos utilizados em todos os centros de desenvolvimento da Alcatel estivessem de acordo com o processo organizacional, foi desenvolvida uma ferramenta de apoio à adaptação. O gerente do projeto define valores para os critérios que caracterizam o projeto e a ferramenta gera um modelo do processo adaptado com links para os elementos do processo (descrição de papéis e atividades, modelos, ferramentas etc).

Os principais benefícios alcançados com a reengenharia dos processos de software na Alcatel foram:

- Menor redundância e maior facilidade de manutenção dos documentos de processo.
- Os processos tornaram-se mais acessíveis e compreensíveis para os desenvolvedores.
- Melhor coordenação entre as mudanças nos processos e as mudanças das ferramentas de apoio.
- Maior facilidade de treinamento especializado, de acordo com os papéis estabelecidos nos processos.
- Maior facilidade de mover um desenvolvedor de um projeto para outro devido ao alinhamento entre os processos.



### 2.3.2 Goddard Space Flight Center

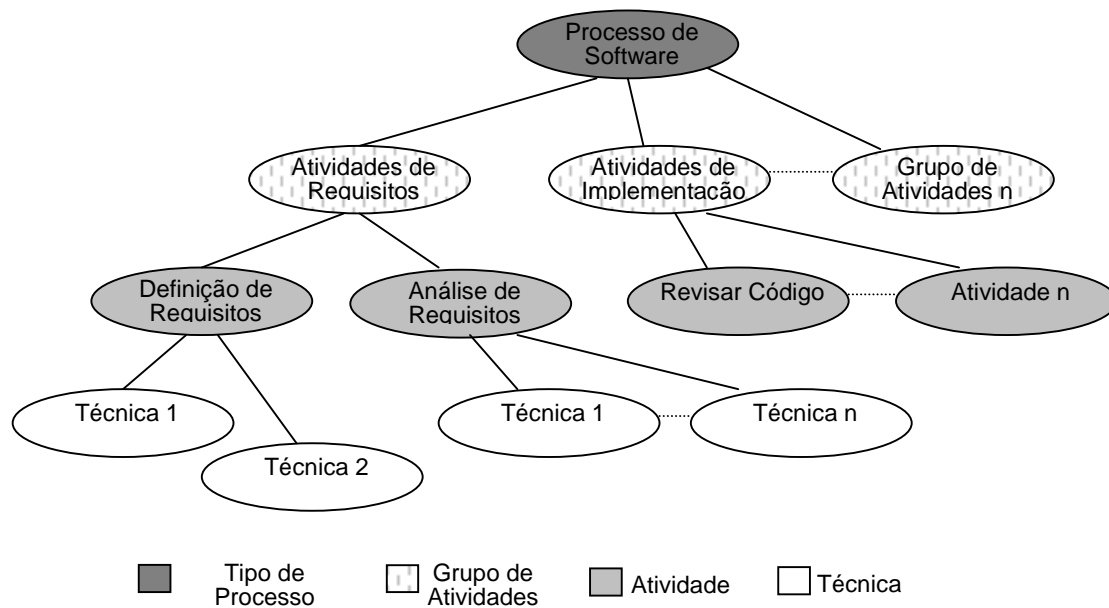
Em 1999 o Software Engineering Laboratory (SEL) iniciou um trabalho para o Information Systems Center (ISC) do Goddard Space Flight Center (GSFC)<sup>8</sup> [40]. Esse trabalho incluía, entre outras coisas, auxiliar a equipe de desenvolvimento de software do GFSC na seleção e adaptação de processos de software. Um dos objetivos era que os processos fossem compatíveis com os padrões ISO 9001 e CMM.

Os processos forma classificados em Novo Desenvolvimento, Manutenção, Alto Reuso e Prototipação.

Além desses tipos de projeto, foram definidos três critérios de adaptação:

- Tamanho da equipe de desenvolvimento: pequena, média, grande.
- Criticidade do sistema: crítico, não-crítico.
- Cronograma: folgado, normal, agressivo.

Os processos definidos são constituídos de Atividades, agrupadas em Grupos de Atividades. Cada Atividade emprega um conjunto de Técnicas (Figura 2-6).



**Figura 2-6:** Elementos do *framework* de processos do GSFC (traduzido de [40])

Para especificar em que condições certas atividades deveriam ou não ser realizadas, tentou-se utilizar uma linguagem estruturada, uma espécie de pseudocódigo. O problema é que cada especificação em pseudocódigo correspondia a somente uma

<sup>8</sup> Site do GSFC: [www.gfsc.nasa.gov](http://www.gfsc.nasa.gov).

instância do processo. Dado que existiam quatro tipos de processo e três critérios de adaptação, com dois desses critérios podendo assumir três valores distintos e o terceiro podendo assumir dois valores distintos, seriam necessárias setenta e duas especificações diferentes (4x3x3x2) para cobrir todas as instâncias possíveis do processo. Esse grande número de especificações tornava difícil o desenvolvimento e análise dos processos.

Para resolver esse problema, foi adotada uma abordagem baseada em uma matriz de representação de processos (Figura 2-7). A coluna esquerda da matriz identifica o pseudocódigo e os passos a serem seguidos. As outras colunas representam as várias combinações de critérios de adaptação. Cada atividade é identificada como obrigatória ou opcional e as revisões são identificadas como formais ou informais.

X.0 Grupo de Atividades	Software Crítico				Software Não Crítico											
	Cronograma Normal		Cronograma Agressivo		Cronograma Normal		Cronograma Agressivo									
	Equipe pequena	Equipe grande	Equipe pequena	Equipe grande	Equipe pequena	Equipe grande	Equipe pequena	Equipe grande								
X.X Atividades Principais	Atividades															
<b>2.0 Projeto</b>																
<b>2.1 Prototipação</b>																
SE existirem riscos técnicos significativos ENTÃO																
Fazer prototipação para reduzir riscos	Realizar para todos os tipos de projetos															
Documentar o esforço de prototipação	X	X	X	X	X	X	O	X								
<b>2.1 Projeto Preliminar</b>																
SE a arquitetura do sistema não existe ainda ENTÃO																
Preparar diagramas de arquitetura alto nível									Realizar para todos os tipos de projeto							
PARA todo novo software customizado FAÇA																
Projetar funções ou especificações de alto nível									Sempre realizar essa atividade							
Conduzir inspeção das funções ou especificações de alto nível									X	X	X	X	O	O	O	O
SE o software é parte de um sistema maior ENTÃO																
Conduzir uma revisão do projeto preliminar									I	F			O	O		
Conduzir revisão conjunta das especificações do software e do projeto preliminar											I	F			O	O
<b>SENÃO</b>																
Conduzir uma revisão do projeto preliminar									I	F						
Conduzir, em conjunto, revisão do projeto preliminar e revisão crítica do projeto											I	F	O	O	O	O
<b>FIM SE</b>																

LEGENDA: X = realizar atividade  
 O = atividade opcional mas recomendada  
 F = realizar revisão formalmente  
 I = realizar revisão informalmente

Figura 2-7: Matriz para adaptação de processos (retirada de [40])

A abordagem baseada em matriz simplificou o desenvolvimento, comparação, apresentação e revisão dos processos. Além disso, abriu caminho para uma futura automatização que possibilite, a partir da seleção do tipo de processo e de valores para os critérios de adaptação, gerar a lista de atividades para o projeto.

### 2.3.3 Raytheon

A Raytheon<sup>9</sup> é uma companhia de alta tecnologia que atua nos ramos de eletrônica, engenharia e aviação, entre outros. Desde 1988, a Raytheon tem acompanhado o trabalho do Software Engineering Institute (SEI) na área de processos de software. Um dos motivos para esse interesse foi a percepção de uma tendência que indicava que os clientes da Raytheon estavam cada vez mais propensos a utilizar o modelo de maturidade do SEI, o CMM, como critério de seleção de fornecedores.

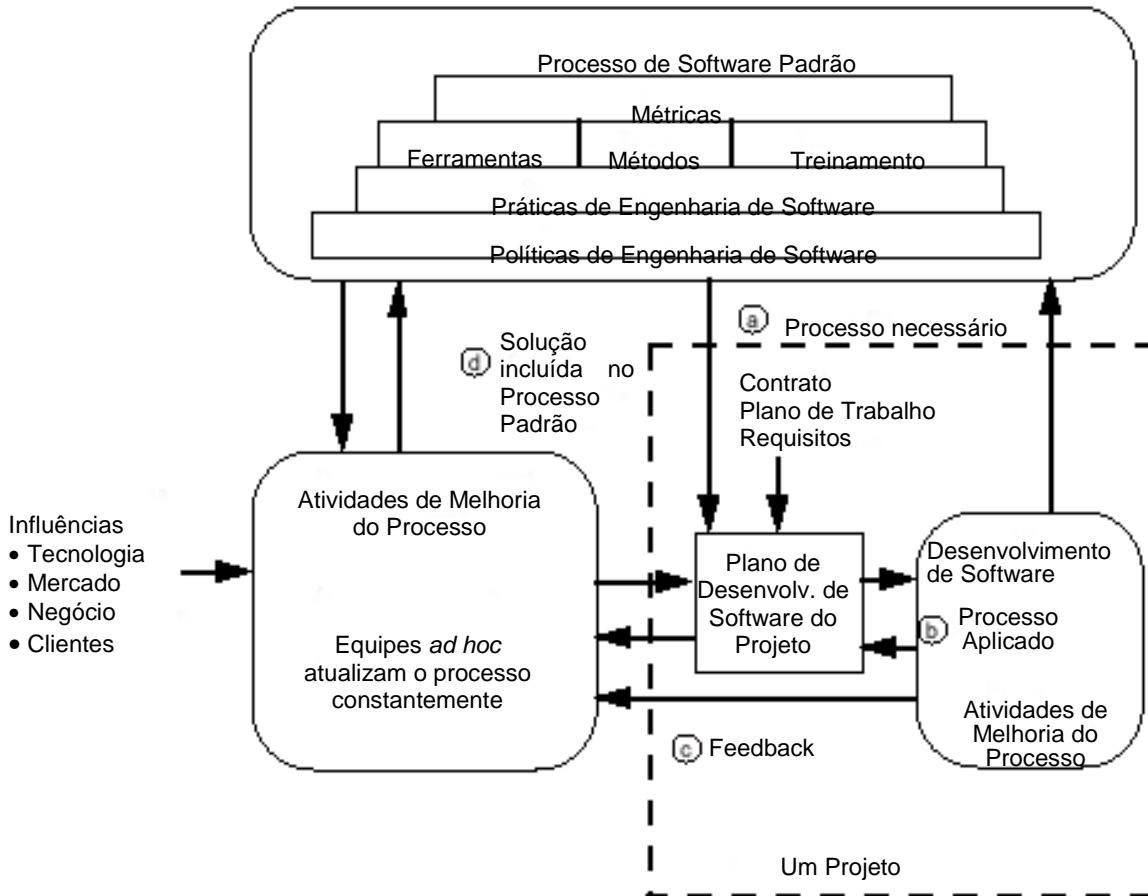
A Raytheon iniciou então um projeto de melhoria do seu processo de software baseado na institucionalização do conhecimento sobre métodos e tecnologias de engenharia de software [14]. O projeto se mostrou bem sucedido, trazendo ganhos em termos de aumento de produtividade, qualidade e previsibilidade do processo.

Nas áreas de definição, melhoria e adaptação de processos, foi desenvolvido um modelo de processos (Figura 8), que representa a estratégia adotada pela empresa.

O projeto padrão da organização é definido por uma política que descreve um conjunto de práticas comuns de engenharia de software. Esse conjunto de práticas é criado através da seleção de melhores práticas utilizadas nos projetos da companhia. As práticas são compostas por procedimentos, que descrevem aspectos críticos do desenvolvimento, ferramentas e treinamento necessários para incrementar a produtividade dos desenvolvedores. Um banco de dados de processos armazena métricas e *baselines*, que podem ser usadas para comparações entre projetos ou dentro do mesmo projeto, e produtos relacionados a projetos, como, por exemplo, lições aprendidas, que podem ser utilizados em desenvolvimentos futuros.

---

<sup>9</sup> Site da Raytheon: [www.raytheon.com](http://www.raytheon.com).



**Figura 2-8:** Estrutura para processos de software da Raytheon (traduzido de [14])

Cada projeto utiliza uma versão do processo padrão adaptado para suas necessidades específicas (Figura 2-8, passo a). Além disso, cada projeto possui seu próprio plano de desenvolvimento, que é um documento que faz a ligação entre o projeto e o processo. O contrato, o plano de trabalho e os requisitos do sistema funcionam como restrições ao plano de desenvolvimento.

No decorrer da aplicação do processo ao projeto (Figura 2-8, passo b), ocorrem dois tipos de *feedback* (Figura 2-8, passo c): no nível de projeto, o plano de desenvolvimento é alterado para refletir lições aprendidas em estágios anteriores do desenvolvimento; no nível organizacional, as lições aprendidas servirão de insumo às atividades de melhoria de processo e podem, inclusive, gerar soluções genéricas que serão adicionadas ao processo padrão da organização.

### 2.3.4 SPAWAR

O SPAWAR (Space and Naval Warfare)<sup>10</sup>, em San Diego, Estados Unidos, através de seu Centro de Sistemas, tem desenvolvido um esforço no sentido de melhorar seu processo de software [17]. O primeiro passo para realizar essa melhoria foi a criação de um repositório de informações de processos. Esse repositório tem como objetivos:

- Difundir melhores práticas para toda a organização.
- Fornecer informações para estimativas de projeto e melhoria do processo de software.
- Servir como fonte de informações para o desenvolvimento de processos para projetos específicos.
- Adequar a organização ao nível 3 do CMM.

As informações contidas no repositório compreendem:

- Uma descrição dos ciclos de vida de software utilizados pela organização.
- Uma descrição do processo padrão da organização, incluindo sua arquitetura e seus elementos.
- Diretrizes para a adaptação do processo padrão.
- Uma descrição de bancos de dados e métricas de projetos e processos.
- Uma descrição de uma biblioteca de documentos.

Os elementos que compõem o repositório são a base para o desenvolvimento de processos para projetos específicos a partir do processo padrão. Entretanto, alguns fatores que impactam o projeto devem ser considerados no momento da adaptação:

- Ambiente de programação.
- Padrões adotados ou impostos.
- Restrições financeiras.
- Obrigações contratuais.
- Cronograma.
- Políticas de desenvolvimento impostas pelo contratante.
- Tecnologia a ser utilizada.
- Nível de experiência dos desenvolvedores com o processo.
- Quantidade de mudanças que o projeto pode sofrer.

---

<sup>10</sup> Site do SPAWAR San Diego: [www.spawar.navy.mil/sandiego](http://www.spawar.navy.mil/sandiego).

Considerados esses fatores, pode ser realizado o processo de adaptação, que é composto de três passos. Cada passo contém atividades, que indicam como realizá-lo, critérios de entrada, saídas e critérios de saída.

### **Passo 1: Selecionar e adaptar processos**

Critérios de Entrada:

- Estratégia de desenvolvimento definida.
- Os membros da equipe responsáveis por definir e documentar os processos de software receberam treinamento em disciplinas de gerenciamento e possuem conhecimento dos requisitos das diretrizes de adaptação.

Atividades:

- Identificar os processos de software requeridos para dar suporte às atividades previstas na estratégia de desenvolvimento escolhida.
- Selecionar processos e modelos da biblioteca de processos.
- Adaptar a estratégia de desenvolvimento de acordo com procedimentos de adaptação documentados.
- Adaptar os processos e os modelos de acordo com os procedimentos documentados.

Saídas:

- Estratégia de desenvolvimento adaptada.
- Processo de software definido e documentado formalmente, através dos planos do projeto (Plano de Desenvolvimento, Plano de Gerência de Configuração e Plano de Garantia da Qualidade).

Critérios de Saída:

- Planos do projeto formalmente aprovados.

### **Passo 2: Documentar as decisões de adaptação**

Critérios de Entrada:

- Planos do projeto formalmente aprovados

Atividades:

- Documentar decisões de adaptação e histórico do projeto para posterior inclusão no banco de dados de processos.
- Documentar alterações nos planos do projeto e submetê-las ao

responsável por suas aprovações.

- Submeter sugestões de melhorias do processo padrão.
- Armazenar medições do esforço necessário para realizar a adaptação.

Saídas:

- Dados de decisões de adaptação documentados.
- Requerimentos aprovados de alterações dos planos do projeto.
- Medições de adaptação.

Critérios de Saída:

- Planos de projeto formalmente aprovados.

### **Passo 3: Aplicar e monitorar os processos adaptados**

Critérios de Entrada:

- Planos do projeto formalmente aprovados.

Atividades:

- Os processos adaptados são aplicados ao projeto e monitorados através das atividades de garantia da qualidade.
- Medições são utilizadas para identificar potenciais modificações no processo.
- Realização de uma análise do projeto, após o seu término, e documentação das informações obtidas.
- Processos identificados como reusáveis são disponibilizados como entradas para o Passo 1.

Saídas:

- Informações sobre o projeto.

Critérios de Saída:

- Projeto finalizado.

## **2.3.5 Bompreço**

O Bompreço<sup>11</sup> é uma grande empresa de varejo que conta com mais de 100 lojas de supermercados e magazines espalhados por nove Estados da Região Nordeste do Brasil.

---

<sup>11</sup> Site do Bompreço: [www.bompreco.com.br](http://www.bompreco.com.br)

Em 2000, o Bompreço, com o objetivo de melhorar o controle gerencial e aumentar a qualidade do seu processo de desenvolvimento, distribuição e instalação de software, contratou a Qualiiti<sup>12</sup>, empresa que atua na área de processos de software, para realizar uma reengenharia do seu processo de desenvolvimento.

A Qualiiti definiu, então, um processo de desenvolvimento de software padrão para o Bompreço. Notou-se, porém, que o processo estava sobrecarregando alguns projetos de menor porte. Nesses projetos, a produtividade estava sendo prejudicada sem que houvesse, em contrapartida, uma melhora significativa em termos de qualidade.

Para solucionar o problema, a Qualiiti desenvolveu um método de adaptação do processo padrão do Bompreço para adequá-lo às características dos projetos. Esse método baseia-se na existência de versões simplificadas, mais leves, do processo padrão. A versão a ser utilizada depende das características de cada projeto.

Os projetos foram divididos em cinco categorias, de acordo com a complexidade e tipo de desenvolvimento:

- Projetos;
- Manutenções;
- Pequenas Solicitações;
- Projetos de pacotes; e
- Projetos para a Internet.

A identificação de um projeto como Projeto, Manutenção ou Pequena Solicitação é feita com base em um sistema de pontuação. São atribuídos pontos ao projeto de acordo com características como:

- Esforço de desenvolvimento;
- Abrangência, em termos de áreas da organização afetadas;
- Interação com outros sistemas; e
- Complexidade na implantação.

Para cada tipo de projeto, são definidos os artefatos obrigatórios e opcionais. A produção de um artefato opcional fica a cargo do Gerente do Projeto. Entretanto, a decisão de não produzir um artefato deve vir sempre acompanhada de uma justificativa.

---

<sup>12</sup> Site da Qualiiti: [www.qualiti.com.br](http://www.qualiti.com.br)



## 2.4 Considerações

Este capítulo apresentou uma visão geral sobre como a adaptação de processos tem sido utilizada na indústria de software. Foram mostradas a forma como a adaptação de processos está inserida no CMM, a relação entre as metodologias de desenvolvimento e a adaptação de processos e alguns casos reais de utilização da adaptação de processos na indústria de software.

Pode-se perceber que, apesar da importância da adaptação de processos, essa é uma área que ainda é negligenciada ou tratada de forma apenas superficial por grande parte das metodologias de desenvolvimento de software. A falta de uma melhor definição, nas metodologias, sobre a forma como a adaptação de processos deve ser realizada acaba dificultando seu uso nas organizações desenvolvedoras. Por conta disso, muitas organizações acabam tomando como base um modelo mais genérico, como o CMM, e sendo obrigadas a desenvolver métodos de adaptação próprios que possam ser utilizados para seus processos específicos.



---

## Capítulo 3

# Rational Unified Process

---

Neste capítulo será descrito o Rational Unified Process (RUP), que, neste trabalho, está sendo utilizado como exemplo de processo padrão a ser adaptado.

O capítulo está estruturado da seguinte forma:

- A Seção 3.1 apresenta uma visão geral do RUP, com seus principais conceitos.
- A Seção 3.2 descreve a disciplina Ambiente do RUP, em especial os aspectos relacionados à configuração do RUP para projetos específicos.
- A Seção 3.3 trata da disciplina Planejamento e Gerenciamento, que será constantemente utilizada nos demais capítulos deste trabalho. Essa seção tem como foco o fluxo Planejamento e Gerenciamento, explicando como essa disciplina é executada ao longo do projeto. Os artefatos de Planejamento e Gerenciamento serão analisados detalhadamente no Capítulo 6.
- A Seção 3.4 tece algumas considerações sobre o capítulo.

### 3.1 Visão Geral do RUP

O Rational Unified Process (RUP) é uma metodologia de desenvolvimento de software criada pela Rational Software Corporation<sup>13</sup>. Teve como base o Objectory Process de Ivar Jacobson, cuja empresa, a Objectory, foi fundida com a Rational. Trata-se de um *framework* genérico que pode ser especializado para se adequar ao desenvolvimento de sistemas para diversas áreas. Aqui, estará sendo utilizado o RUP 2002 [16], versão mais recente da metodologia.

---

<sup>13</sup> Site da Rational Software Corporation: [www.rational.com](http://www.rational.com)

### 3.1.1 Elementos do RUP

O RUP é um *framework* composto por vários tipos de elementos que, em conjunto, formam o processo completo. A seguir, serão descritos os principais tipos de elementos do RUP.

#### Atividades

Definem o trabalho que será feito. Uma atividade é constituída por um conjunto de passos, que definem como a atividade deve ser realizada. Cada atividade é realizada por um ou mais papéis e possui artefatos de entrada e saída.

#### Papéis

São os responsáveis por realizar as atividades. Um papel não está associado a uma pessoa específica. Um papel pode ser desempenhado por mais de uma pessoa e uma pessoa pode desempenhar mais de um papel.

#### Artefatos

São os produtos gerados pelas atividades. Podem ser documentos, modelos ou elementos de software, como programas ou bibliotecas de componentes.

#### Disciplinas

São agrupamentos de atividades interrelacionadas. Até o RUP 2000, as disciplinas eram chamadas de fluxos de trabalho. Uma disciplina determina a ordem em que as atividades serão executadas. As disciplinas do RUP [16] são:

- **Modelagem de Negócio:** disciplina responsável por entender a organização onde o software será implantado e identificar problemas e oportunidades de melhoria.
- **Requisitos:** disciplina responsável por determinar o escopo do sistema a ser desenvolvido e comunicar de forma clara o que o sistema deve fazer.
- **Análise e Projeto:** disciplina responsável pela modelagem do sistema, por transformar requisitos em projeto e por desenvolver uma arquitetura adequada.
- **Implementação:** disciplina responsável pela codificação do software (classes, objetos, subsistemas, camadas etc) e por testar os componentes implementados (teste de unidade).

- **Testes:** disciplina responsável por validar as funções do sistema, por verificar se os requisitos foram implementados corretamente e por encontrar e documentar defeitos.
- **Implantação:** disciplina responsável por disponibilizar o software para os usuários finais.
- **Gerência de Configuração e Mudanças:** disciplina responsável por identificar itens de configuração, restringindo e verificando mudanças feitas nesses itens e gerenciando versões.
- **Planejamento e Gerenciamento do Projeto:** disciplina responsável por planejar e controlar a execução de um projeto, alocando equipes, acompanhando a realização de tarefas e gerenciando riscos.
- **Ambiente:** disciplina responsável pelo ambiente de desenvolvimento (processos e ferramentas) que irá dar suporte à equipe de desenvolvimento.

As três últimas disciplinas citadas são classificadas como disciplinas de suporte, já que não são diretamente responsáveis por produzir o produto final.

## Fases

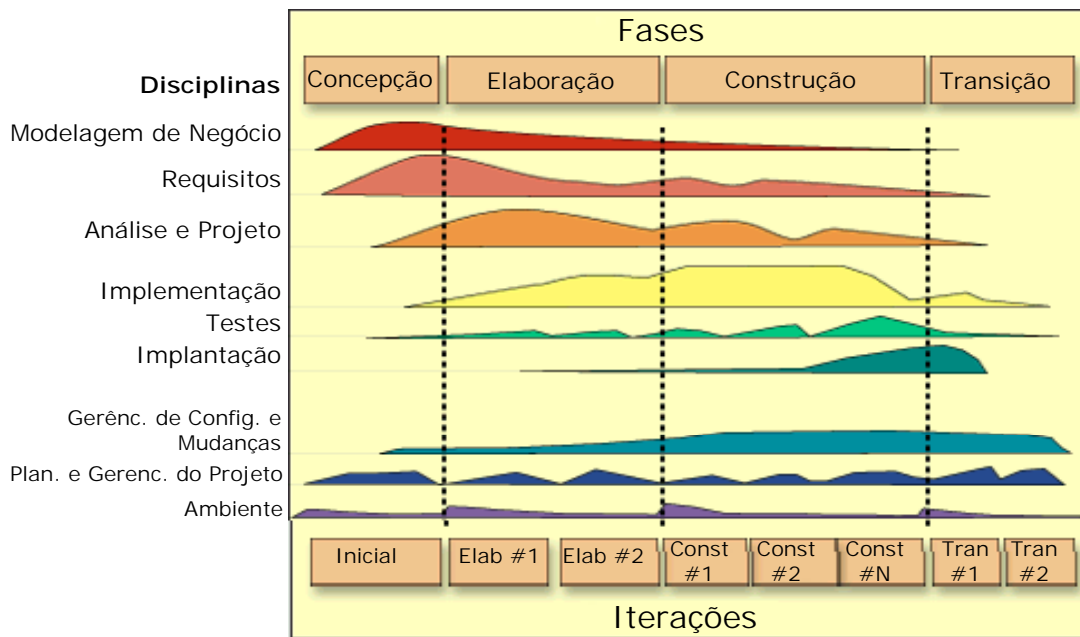
O desenvolvimento utilizando o RUP é feito em fases, que indicam a ênfase do projeto em um dado instante. O RUP possui quatro fases:

- **Concepção:** ênfase na definição do escopo do sistema.
- **Elaboração:** ênfase na arquitetura e na análise e modelagem do sistema.
- **Construção:** ênfase no desenvolvimento do sistema, produção de código.
- **Transição:** ênfase na implantação do sistema e produção e entrega de produtos associados (programa de instalação, manual do usuário etc).

Uma fase é dividida em **iterações**, que são intervalos de tempo definidos onde uma parte do sistema é desenvolvida.

A Figura 3-1 apresenta a relação entre fases, iterações e disciplinas, mostrando a concentração das atividades de cada disciplina ao longo das fases do processo. Pode-se ver, por exemplo, que as atividades da disciplina Análise e Projeto recebem maior ênfase na fase de Elaboração enquanto as atividades disciplina Implementação estão concentradas na fase de Construção e recebem pouca ênfase na fase de Concepção. As atividades da disciplina Planejamento e Gerenciamento estão, em comparação com as

atividades das demais disciplinas, distribuídas de forma mais homogênea ao longo de todo o projeto.



**Figura 3-1:** Visão geral do RUP (traduzido de [16])

### 3.1.2 Boas Práticas de Desenvolvimento do RUP

O RUP é baseado em boas práticas de desenvolvimento utilizadas com sucesso na indústria de software. Essas boas práticas serão descritas a seguir.

#### Desenvolvimento Iterativo

O RUP utiliza o modelo de ciclo de vida iterativo, o que implica que o desenvolvimento do software é feito incrementalmente, através de iterações que contemplam atividades de todas as disciplinas. As principais vantagens do modelo iterativo sobre o modelo cascata são [5]:

- O modelo iterativo leva em consideração que os requisitos podem mudar, ao contrário do modelo cascata, que concentra todas as atividades de requisitos no início do desenvolvimento.
- No modelo iterativo a integração é feita progressivamente, ao longo do projeto, e não somente no final do projeto, o que diminui a complexidade e o esforço necessário para essa atividade.

- Como atividades de implementação e integração são realizadas desde as primeiras iterações do projeto, riscos e problemas tendem a ser descobertos mais cedo, diminuindo o retrabalho causado por eles.
- O modelo iterativo permite uma maior flexibilidade na data de lançamento ou implantação do produto, já que logo nas primeiras iterações já existe um produto funcional, ainda que com um número reduzido de funcionalidades.
- No modelo iterativo, o trabalho da equipe de desenvolvimento é melhor distribuído ao longo do projeto. Analistas, programadores e testadores possuem atividades a serem realizadas logo no início do projeto.
- Como, no modelo iterativo, as atividades de análise e projeto são realizadas no início do projeto, as oportunidades de reuso podem ser identificadas com maior antecedência.
- No modelo iterativo, o processo pode ser melhorado ao longo das iterações.

### **Gerenciamento de Requisitos**

O gerenciamento de requisitos permite levantar, organizar e controlar mudanças nos requisitos de forma sistemática. Um bom gerenciamento de requisitos permite um melhor controle de projetos complexos, melhora a qualidade do produto e a satisfação do usuário, reduz custos e atrasos e melhora a comunicação entre os membros da equipe de desenvolvimento. No RUP, o gerenciamento é feito através de casos de uso. Os casos de uso são utilizados durante todo o desenvolvimento, afetando praticamente todas as disciplinas do processo e fazendo a ligação entre elas. Isso permite que seja mantida uma maior consistência entre os requisitos e o sistema desenvolvido. Casos de uso podem ser utilizados, também, para avaliar o progresso de um projeto [41, 42]. Por utilizar os casos de uso como guia para o desenvolvimento, diz-se que o RUP é guiado por casos de uso.

### **Arquitetura Baseada em Componentes**

No RUP, o foco das primeiras iterações é produzir e validar a arquitetura do sistema através do desenvolvimento de um protótipo executável da arquitetura. Por conta dessa preocupação com a arquitetura, pode-se dizer que o RUP é centrado na arquitetura. A definição da arquitetura no início do projeto e o seu refinamento nas iterações subseqüentes permite a identificação progressiva de que componentes dessa

arquitetura serão desenvolvidos e quais serão reusados, já que a estrutura e a interação entre esses componentes já estão definidas na arquitetura.

### **Modelagem Visual**

Modelos são simplificações da realidade que permitem entender mais facilmente problemas complexos. O RUP utiliza a UML (Unified Modeling Language) [43], que é uma linguagem gráfica que fornece uma série de diagramas para a modelagem de sistemas. No RUP, estão definidos os diagramas que devem ser utilizados e a forma de utilização desses diagramas.

### **Verificação Contínua da Qualidade**

O RUP divide qualidade em duas áreas distintas: qualidade do produto e qualidade do processo. Qualidade do processo é a implementação adequada do processo durante o desenvolvimento do produto, estando relacionada com a qualidade dos artefatos produzidos (planos, modelos etc). A qualidade do produto refere-se à qualidade do produto final e dos elementos que o compõe, como componentes, subsistemas e arquitetura. O RUP possui atividades de verificação e avaliação para garantir a qualidade do produto. Essas atividades estão concentradas, principalmente, na disciplina Testes.

### **Controle de Mudanças**

No modelo iterativo, é comum que artefatos sejam modificados. A evolução dos artefatos ao longo das iterações faz parte da natureza do modelo. Por isso, existe a necessidade de controlar as mudanças feitas em requisitos, projeto, implementação etc. Além disso, é preciso acompanhar os defeitos e problemas identificados durante o projeto. O RUP contempla todas essas atividades através da disciplina Gerência de Configuração e Mudanças.

## **3.2 Configuração do RUP: a Disciplina Ambiente**

O RUP é um *framework* para processos em geral, o que permite que seja usado em praticamente todo tipo de projeto. Como consequência dessa abrangência, há a necessidade de configurar o RUP para que ele possa ser utilizado de forma eficiente.

A configuração do RUP é descrita na disciplina Ambiente. A disciplina Ambiente, entretanto, não define claramente como o RUP deve ser adaptado de acordo



com as características dos projetos, limitando-se a comentários genéricos que, embora sejam úteis, não são suficientes para guiar a adaptação.

Além das informações presentes na disciplina Ambiente, na documentação de cada artefato do RUP existe uma seção (“*Tailoring*”) que descreve quando esse artefato deve ser modificado. Essas informações, entretanto, estão longe do nível de detalhamento recomendável e nem sempre estão associadas às características que o RUP define como importantes para a sua configuração.

A configuração do RUP é feita em duas etapas. Primeiro deve-se configurar o RUP para a organização que irá utilizá-lo, criando, assim, um processo padrão para a organização<sup>14</sup>. Esse processo pode, inclusive, ser o próprio RUP. Na segunda etapa da configuração, o processo padrão deve ser configurado para cada projeto, levando em conta suas características. Esse processo específico do projeto<sup>15</sup> é descrito por um documento chamado Caso de Desenvolvimento<sup>16</sup>.

No nível organizacional, devem ser produzidos Casos de Desenvolvimento, modelos e diretrizes que possam ser reutilizados em vários projetos. No nível de projeto, é criado um Caso de Desenvolvimento descrevendo o processo específico para o projeto. Esse Caso de Desenvolvimento pode ser uma especialização de um Caso de Desenvolvimento já desenvolvido no nível organizacional. Além disso, no nível de projeto, os modelos e diretrizes desenvolvidos no nível organizacional devem ser especializados para atender às necessidades específicas do projeto específico.

Para adaptar o RUP para projetos específicos deve-se escolher, com base nas características do projeto, que artefatos devem ser produzidos, que atividades devem ser realizadas, que papéis necessitam ser desempenhados etc. Como o RUP, na sua forma padrão, contém todos os elementos do processo (artefatos, atividades etc), a forma mais comum de adaptação é a exclusão de elementos que não sejam relevantes para uma circunstância específica.

De acordo com a disciplina Ambiente do RUP, vários fatores exercem influência na adaptação do processo. Esses fatores serão brevemente comentados a seguir.

---

<sup>14</sup> No original, em inglês, *organization-wide process*.

<sup>15</sup> No original, em inglês, *project-specific process*.

<sup>16</sup> No original, em inglês, *Development Case*.

### 3.2.1 Tipo de Desenvolvimento

O tipo de desenvolvimento refere-se ao contexto no qual o software está sendo desenvolvido. Os tipos de desenvolvimento mais comuns são:

- *Desenvolvimento por Contrato*: o software é produzido para um cliente específico. Nesse tipo de desenvolvimento, existem, normalmente, mais *stakeholders*, tornando necessária a produção de mais documentos, protótipos etc para tornar claro o progresso do desenvolvimento. Além disso, os documentos têm que ser escritos de forma que os clientes, normalmente pessoas com formações diferentes, tenham facilidade de entender.
- *Desenvolvimento Especulativo ou Comercial*: o produto é desenvolvido para o mercado (software de prateleira). O tempo para lançar o produto no mercado é, normalmente, mais importante que suas funcionalidades.
- *Desenvolvimento Interno*: o produto é desenvolvido para o uso da própria companhia. Nesse tipo de desenvolvimento, existe maior participação do usuário final. Os documentos podem ser escritos de forma mais “técnica”.

### 3.2.2 Tamanho do Projeto ou Esforço de Desenvolvimento

O tamanho de um projeto pode ser descrito por algumas métricas, como LOC (linhas de código) e FP (pontos de função). Quanto maior o projeto, maior o tamanho da equipe e, conseqüentemente, maior o nível de formalismo necessário. Mais pessoas na equipe implica a necessidade de uma forma de comunicação mais formal e, por isso, um processo mais pesado. A comunicação entre os membros da equipe pode ser bastante afetada se as pessoas estiverem geograficamente dispersas.

### 3.2.3 Grau de Inovação

O grau de inovação refere-se à experiência da organização com o processo de desenvolvimento e com o tipo de produto a ser desenvolvido. Um novo projeto, sem semelhantes desenvolvidos anteriormente, exige maior atenção às fases de concepção e elaboração. Torna-se necessário dar maior ênfase à elicitação de requisitos, por exemplo. Se o projeto é uma nova versão de um sistema ou se o desenvolvimento estiver no segundo (ou subsequente) ciclo, a atenção volta-se para a gerência de

configuração. A utilização de um processo novo, por sua vez, pode, num primeiro projeto, diminuir a produtividade da equipe.

### **3.2.4 Tipo de Aplicação**

O tipo de aplicação afeta o processo de desenvolvimento porque cada tipo de sistema impõe restrições diferentes ao desenvolvimento. Essas restrições podem se referir à complexidade técnica do software, restrições de tempo, performance, necessidade de atender normas, criticidade do sistema etc. Uma aplicação crítica, por exemplo, exige um maior grau de formalismo no desenvolvimento, um processo mais pesado.

### **3.2.5 Processo de Desenvolvimento Atual**

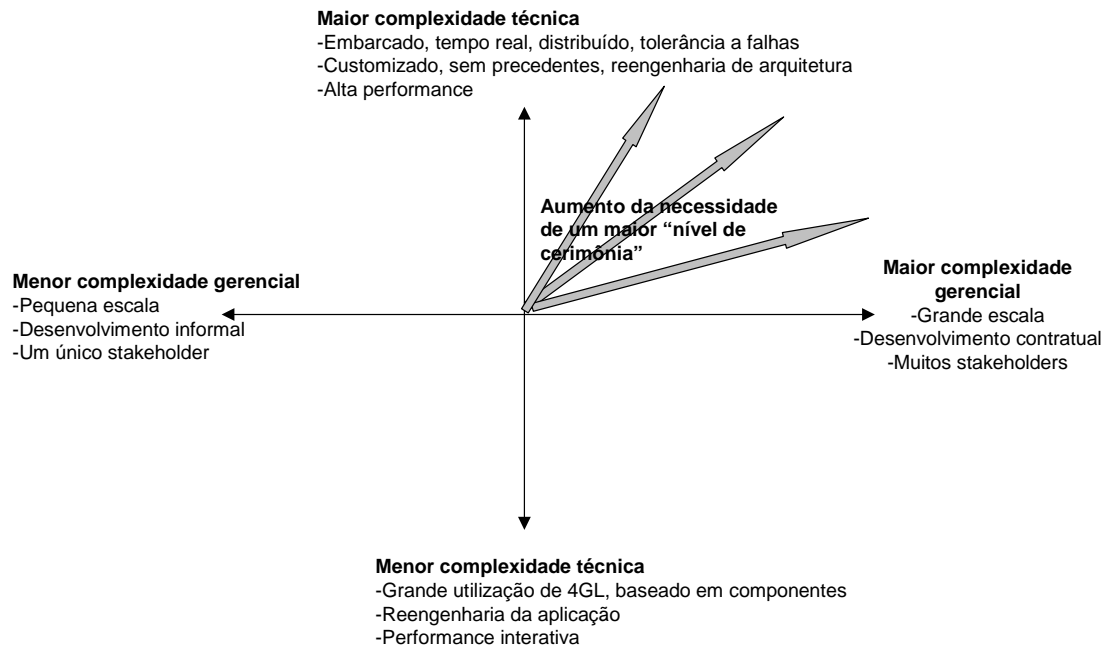
Se estiver havendo uma mudança no processo de desenvolvimento, algumas partes do processo antigo podem ter que ser usadas em conjunto com partes do processo novo.

### **3.2.6 Fatores Organizacionais**

Para implantar um processo em uma organização, deve-se levar em conta aspectos como a estrutura e cultura organizacionais, as habilidades e atitudes das pessoas envolvidas e experiências anteriores vividas pela organização.

### **3.2.7 Complexidade Técnica e Gerencial**

De uma forma geral, a necessidade de um processo mais pesado cresce de acordo com a complexidade técnica e gerencial do projeto. A complexidade gerencial afeta diretamente o nível de cerimônia do processo devido ao aumento de revisões formais, artefatos, marcos de referência etc. O aumento da complexidade técnica, por sua vez, aumenta o número de técnicas, ferramentas e papéis especializados, gerando um maior número de atividades (Figura 3-2).



**Figura 3-2:** Complexidade do projeto *versus* nível de cerimônia do processo (traduzido de [16])

### 3.3 Fluxo de Planejamento e Gerenciamento do RUP

Segundo o RUP, Planejamento e Gerenciamento de Software é “a arte de balancear objetivos discordantes, gerenciar riscos e superar restrições para entregar, com sucesso, um produto que atenda às necessidades dos clientes e usuários”. É considerada, juntamente com as disciplinas Gerência de Configuração e Mudanças e Ambiente, uma disciplina de suporte ao projeto. Muitos dos conceitos de Planejamento e Gerenciamento utilizados pelo RUP estão baseados no Project Management Body of Knowledge (PMBOK) [44], um conjunto de boas práticas de planejamento e gerenciamento organizado pelo Project Management Institute<sup>17</sup>.

Os principais aspectos tratados na disciplina Planejamento e Gerenciamento do RUP são o gerenciamento de riscos, o planejamento de um projeto iterativo e o acompanhamento do progresso através de métricas.

<sup>17</sup> Site do PMI: [www.pmi.org](http://www.pmi.org)

Alguns aspectos importantes, por outro lado, não são considerados pelo RUP, como gerenciamento de pessoal, controle de orçamento e gerenciamento de contratos com clientes e fornecedores.

A Figura 3-3 ilustra a disciplina Planejamento e Gerenciamento do RUP, subdividindo-o em vários subfluxos e organizando esses subfluxos na forma de um diagrama de atividades.

A iteração inicial começa com o subfluxo “Conceber Novo Projeto”. Esse subfluxo tem como objetivo obter uma visão geral do projeto através de uma versão inicial dos artefatos Visão, Estudo de Viabilidade e Lista de Riscos, que serão utilizados para produzir a primeira versão do Plano de Desenvolvimento de Software e o Plano de Iteração para essa iteração inicial.

Os planos iniciais produzidos em “Conceber Novo Projeto” servem de subsídio para o subfluxo “Avaliar Escopo e Risco do Projeto”, onde a Lista de Riscos será refinada e uma versão mais completa e atualizada do Estudo de Viabilidade será produzida.

A Lista de Riscos e o Estudo de Viabilidade produzidos, juntamente com o documento de Visão produzido na disciplina Requisitos, servirão de base para as atividades do subfluxo “Desenvolver Plano de Desenvolvimento de Software”. Nesse subfluxo, serão produzidos os planos necessários ao projeto, como Plano de Medições, Plano de Gerenciamento de Riscos, Plano de Aceitação do Produto, Plano de Resolução de Problemas e Plano de Garantia da Qualidade. O Plano de Desenvolvimento de Software é refinado e pode englobar os demais planos. Ao final do subfluxo “Desenvolver Plano de Desenvolvimento de Software”, o planejamento inicial do projeto deve fornecer informações suficientes para que se possa decidir se o projeto deve continuar ou ser abortado.

Caso os planos sejam aprovados e o projeto tenha prosseguimento, será realizado o subfluxo “Planejar Próxima Iteração”, que possui a mesma estrutura do subfluxo “Planejar Próxima Iteração” presente nas iterações subseqüentes do projeto e que será detalhado posteriormente.

A seqüência de subfluxos descrita até aqui se aplica apenas à iteração inicial ou preliminar do projeto, sendo essa iteração diferente das demais, que possuem uma estrutura comum entre si quanto à seqüência de subfluxos realizados. Assim, a

seqüência de subfluxos apresentada a seguir é repetida em todas as iterações do projeto, com exceção da iteração preliminar.

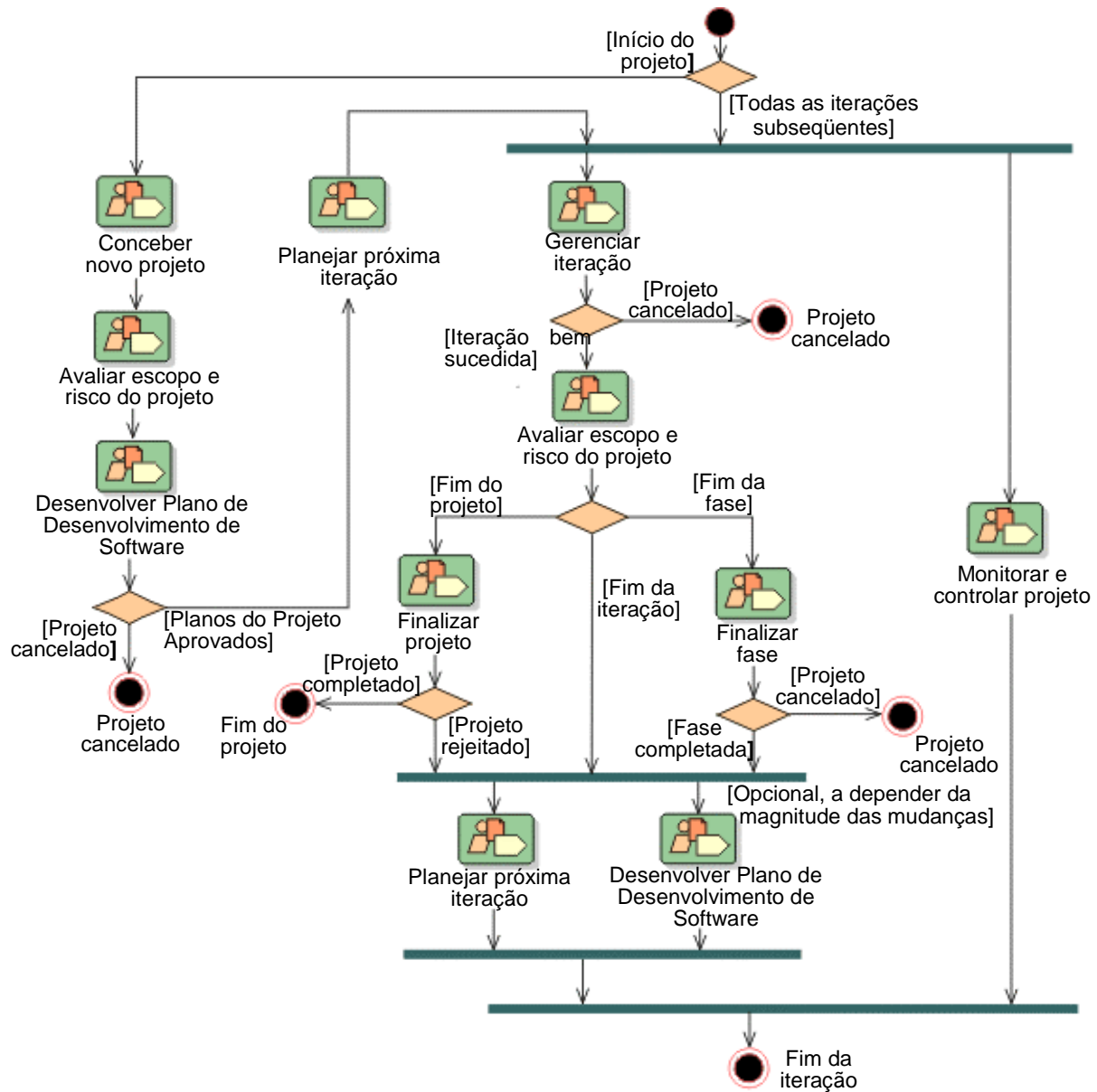


Figura 3-3: Fluxo de Planejamento e Gerenciamento (traduzido de [16])

Cada iteração começa com o subfluxo “Gerenciar Iteração”, onde a iteração é executada. Esse subfluxo engloba atividades para iniciar, finalizar e avaliar a iteração. Após a avaliação da iteração, deve-se decidir se o projeto irá continuar ou não. Em caso afirmativo, o subfluxo seguinte é “Avaliar Escopo e Risco do Projeto”, onde a Lista de Riscos e o Estudo de Viabilidade são atualizados.

Após o subfluxo “Avaliar Escopo e Risco do Projeto”, existem três caminhos a seguir. Se a iteração for a última de uma fase, segue o subfluxo “Finalizar Fase”, onde a

fase é avaliada e uma revisão do marco de referência principal associado a essa fase é realizada. Se a iteração for a última do projeto, segue o subfluxo “Finalizar Projeto”, onde, com base nas Avaliações de Status, na Avaliação da Iteração e no Plano de Desenvolvimento de Software, é decidida a aceitação ou não projeto. Se a iteração não marcar o final de uma fase nem do projeto, será realizado o subfluxo “Planejar Próxima Iteração”. Nesse subfluxo será produzido um Plano de Iteração, que servirá de base para a execução da iteração seguinte. O Plano de Iteração é desenvolvido a partir dos artefatos Visão (da disciplina Requisitos), Lista de Riscos, Plano de Desenvolvimento de Software, Atributos de Requisitos (da disciplina Requisitos) e Documento de Arquitetura de Software (da disciplina Análise e Projeto). Opcionalmente, caso a magnitude das mudanças necessárias ao Plano de Desenvolvimento de Software seja alta, pode ser realizado, em paralelo ao desenvolvimento do Plano de Iteração, o subfluxo “Desenvolver Plano de Desenvolvimento de Software”, onde um novo plano será produzido.

Finalmente, ao longo de toda a iteração, é realizado o subfluxo “Monitorar e Controlar Projeto”, que abrange o trabalho contínuo de acompanhamento realizado pelo Gerente do Projeto. Nesse subfluxo, estão contidas atividades como acompanhar e reportar o status do projeto, alocar e atribuir tarefas e gerenciar exceções e problemas.

### 3.4 Considerações

Neste capítulo, foram discutidos alguns aspectos importantes do RUP, Rational Unified Process. Além de uma visão geral do processo, foram detalhadas as disciplinas Ambiente, que trata da adaptação do RUP, e Planejamento e Gerenciamento, que será bastante utilizada no decorrer do trabalho.

O RUP, por ser um *framework* que tenta englobar todos os elementos necessários para o desenvolvimento dos mais variados tipos de software, precisa ser adaptado às características de cada projeto. Entretanto, a disciplina Ambiente, que é responsável por definir como essa adaptação deve ser realizada, é muito vaga e superficial, não sendo suficiente para guiar a adaptação.





---

## Capítulo 4

# Modelo de Adaptação de Processos de Software

---

Este capítulo tem como objetivo apresentar a arquitetura e o funcionamento geral do Modelo de Adaptação de Processos de Software (MAPS). O capítulo segue a seguinte estrutura:

- A Seção 4.1 apresenta uma visão geral do modelo, seus objetivos, estrutura e funcionamento.
- A Seção 4.2 identifica que aspectos do MAPS serão detalhados nesse trabalho e que circunstâncias de utilização do Modelo serão consideradas.
- A Seção 4.3 apresenta algumas estratégias que podem ser utilizadas para implantar o MAPS em uma organização.
- A Seção 4.4 identifica oportunidades de automação do MAPS.
- A Seção 4.5 apresenta uma breve conclusão do capítulo.

### 4.1 Visão Geral do Modelo

Neste trabalho, pretende-se analisar a influência das características do projeto no processo de desenvolvimento e desenvolver um modelo para a adaptação de um processo padrão para um projeto específico com base nessas características.

Para auxiliar a escolha de um processo adequado para cada projeto, será apresentado um Modelo de Adaptação de Processos de Software (MAPS) baseado nas características dos projetos.

São objetivos do MAPS:

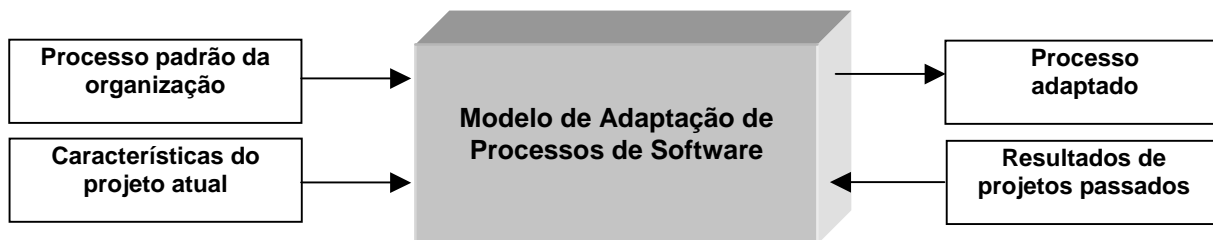
- Estabelecer um método de adaptação de um processo padrão da organização para projetos específicos.

- Fornecer mecanismos para diminuir o esforço necessário para realizar a adaptação de processos.
- Permitir a melhoria contínua dos processos utilizados pela organização.
- Estar em conformidade com o nível 3 do CMM [15], que define uma estrutura para utilização e adaptação dos processos da organização.

Dado que a empresa possui um processo padrão, a adaptação deste será feita com base nas características do projeto, que devem ser fornecidas ao MAPS. O projeto será comparado com projetos passados para um possível reuso de partes de processos anteriormente definidos. Por fim, o MAPS deve fornecer diretrizes para a adaptação do processo de forma que no final seja obtido um processo adequado às características do projeto.

Deve ficar claro que as informações contidas no MAPS são apenas um ponto de partida, podendo não se aplicar totalmente a uma organização específica. Ao longo de cada projeto, devem ser colhidas informações e resultados que irão calibrar o MAPS, adequando-o às características de cada organização.

A Figura 4-1 representa o funcionamento básico do MAPS.



**Figura 4-1:** Modelo de Adaptação de Processos de Software: entradas e saída

A partir do processo padrão da organização, o MAPS deve gerar um processo específico para cada projeto. Para isso, são utilizados, além do próprio processo padrão, as características do projeto atual, para o qual deseja-se um processo adaptado, e resultados de projetos passados que possuam semelhanças com o projeto atual.

O Modelo é constituído de três módulos (Figura 4-2): Modelo de Caracterização de Projetos, PConfig e Base de Processos.

A utilização do Modelo de Caracterização de Projetos tem como objetivo realizar uma comparação entre os projetos. A partir dessa comparação, deseja-se permitir que um processo (ou partes de um processo) que tenha sido utilizado com

sucesso em um projeto seja reutilizado em projetos com características semelhantes. Existe ainda a possibilidade de melhorar esse processo de forma que, ao longo do tempo, a organização adquira um portfólio de processos bem sucedidos para os vários tipos de projeto.

Para caracterizar os projetos de software, serão analisadas as principais características do projeto que impactam o processo. Seguindo a idéia defendida por Cameron [45] de que “modularidade é um pré-requisito para reuso e para adaptabilidade”, a caracterização dos projetos será feita por disciplinas do processo (Planejamento e Gerenciamento, Requisitos etc) e não através do processo completo. Assim, o conjunto de características do projeto que influencia uma determinada disciplina é distinto do conjunto de características que influenciam outra disciplina, embora esses conjuntos possam ter alguma interseção. Um projeto pode, portanto, ser considerado, ao mesmo tempo, semelhante a outro com relação a uma determinada disciplina e diferente com relação a uma outra disciplina. Essa abordagem busca facilitar a comparação entre projetos e permitir um maior reuso de partes dos processos de software, atenuando um problema recorrente na analogia entre projetos de software que é encontrar projetos realmente semelhantes. O conceito de divisão dos processos em disciplinas, semelhante ao conceito de blocos de construção descritos por Budlong [12], também será utilizado nos outros componentes do MAPS. Uma vantagem adicional dessa estratégia é que ela torna o MAPS mais compatível com o CMMI (Capability Maturity Model Integrated) [46], modelo de maturidade que deve, no futuro, substituir o CMM. O CMMI, através do seu modelo contínuo [47], permite que uma organização atinja determinado nível de maturidade para apenas uma parte do processo, por exemplo, Gerenciamento de Projeto. Essa melhoria de processo por áreas do processo está em concordância com a idéia aqui utilizada de caracterizar, adaptar e melhorar o processo por disciplinas.

PConfig é formado por diretrizes para configurar o processo padrão para um projeto específico, de acordo com as características do projeto. Para essa configuração será tomado, como processo padrão, o Rational Unified Process (RUP), na sua versão 2002 [16].

A importância de armazenar informações sobre os processos de software já utilizados na organização, bem como sobre sua utilização em projetos, é discutida em [48] e [49]. Um repositório de informações permite a identificação de situações

recorrentes, diminuindo a quantidade de esforço duplicado e difundindo as lições aprendidas da organização. No Modelo de Adaptação de Processos de Software, a função de repositório de informações é exercida pela Base de Processos. A Base de Processos armazena informações/resultados dos projetos já realizados e os respectivos processos utilizados. A Base de Processos corresponde, no Modelo de Adaptação de Processos de Software, ao Banco de Dados de Processos da Organização e à Biblioteca de Documentação Relacionada a Processos de Software descritos no CMM [15]. A Base de Processos armazena e organiza as informações necessárias para o reuso de processos, funcionando em conjunto com o Modelo de Caracterização de Projetos. Esse reuso pode vir acompanhado da melhoria do processo armazenado, utilizando-se, para isso, uma avaliação do processo feita em projetos anteriores. Essa avaliação também estará armazenada na Base de Processos. Espera-se que, com a constante utilização do Modelo de Adaptação de Processos de Software e o conseqüente armazenamento de informações da Base de Processos, a organização adquira uma “família” de processos, todos baseados no seu processo padrão, que constituam um conjunto de soluções já testadas para as situações mais comuns na organização.

Os componentes do Modelo contribuem da seguinte forma para a consecução dos objetivos estabelecidos no início do capítulo:

- A utilização, em conjunto, do Modelo de Caracterização de Projetos e de PConfig constituem o método de adaptação utilizado.
- A reutilização de processos anteriores, através da interação entre o Modelo de Caracterização de Projetos e a Base de Processos, contribui para diminuir o esforço necessário para adaptar o processo.
- O armazenamento, na Base de Processos, das avaliações acerca dos processos utilizados permite que o reuso de processos venha acompanhado de uma melhoria de processos, resultando em uma maior qualidade dos processos que serão reusados no futuro.
- A estrutura do Modelo de Adaptação de Processos de Software é bastante semelhante à proposta pelo CMM, com a Base de Processos fazendo o papel do Banco de Dados de Processos de Software da Organização e da Biblioteca de Documentação Relacionada a Processos de Software e o PConfig, em conjunto com o Modelo de Caracterização de Projetos, correspondendo aos critérios e diretrizes para adaptação [15]. Deve ficar

claro que não é objetivo desse trabalho fornecer um método ou uma estratégia para atingir o nível 3 do CMM. O CMM foi utilizado apenas como referência e o Modelo foi estruturado de forma a diminuir o retrabalho e a quantidade de alterações necessárias no caso da organização ter como objetivo atingir o nível 3 do CMM. Ou seja, o Modelo de Adaptação de Processos de Software é compatível com o CMM e pode ser considerado como um passo em direção ao CMM nível 3, mas não é suficiente para atingir esse nível.

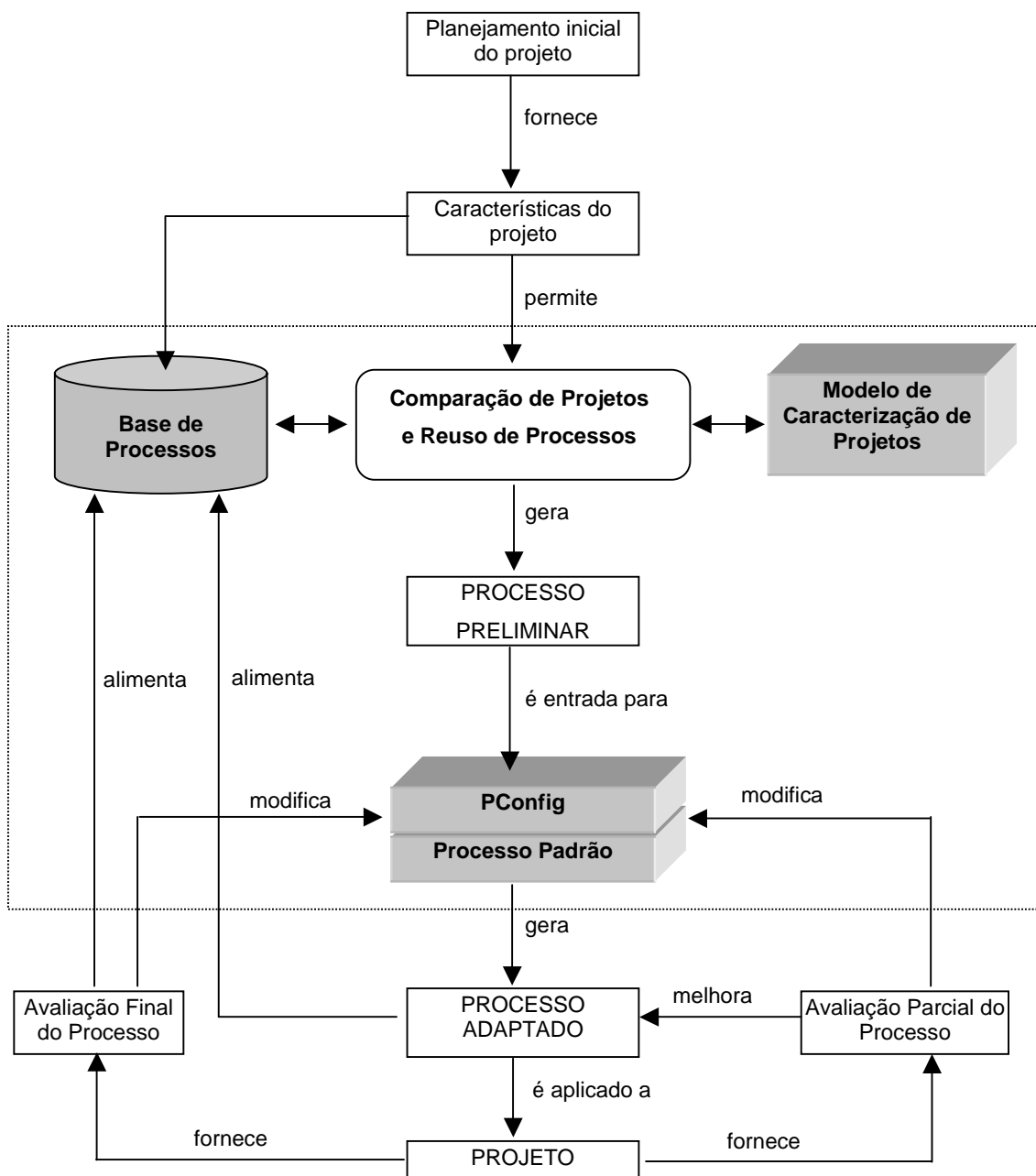


Figura 4-2: Modelo de Adaptação de Processos de Software (MAPS)

Em resumo, a adaptação de processos de software através do MAPS segue os seguintes passos:

1. **Identificação das Características.** A partir de um planejamento inicial do processo, são identificadas as características do projeto.
2. **Comparação de Projetos.** As características do projeto servem como base para a identificação de projetos semelhantes, armazenados na Base de Processos, utilizando o método de comparação estabelecido no Modelo de Caracterização de Projetos. Esse método de comparação analisa os projetos disciplina a disciplina, e não através do projeto como um todo.
3. **Reuso de Processos.** Para cada disciplina, uma lista com os projetos identificados como semelhantes ao projeto atual, os processos que foram utilizados nesses projetos e uma avaliação desses processos é retornada para que o Engenheiro de Processos escolha os mais adequados, ou seja, aqueles que serão reusados.
4. **Complementação do Processo.** O reuso de disciplinas de processos utilizados em projetos anteriores dá origem a um processo preliminar. Esse processo preliminar pode estar incompleto, ou seja, pode conter apenas algumas disciplinas, já que é possível que, para algumas disciplinas, não exista um projeto semelhante anterior. Para completar o projeto preliminar, é utilizado o PConfig. No PConfig, a partir do processo padrão da organização e das características do projeto, serão adaptadas as disciplinas que não puderam ser reusadas de processos de projetos anteriores.
5. **Utilização do Processo.** A aplicação do PConfig resulta em um processo adaptado, completo, que será utilizado no projeto.
6. **Avaliação do Processo.** Periodicamente, no decorrer do projeto, são feitas avaliações parciais do processo. Essas avaliações podem ser feitas, por exemplo, ao final de cada iteração. As avaliações parciais podem implicar em melhorias do processo adaptado e mudanças no PConfig. As mudanças no PConfig podem ocorrer de duas formas: modificando o processo padrão, no caso de identificação de melhorias que sejam úteis para os vários processos da organização, ou calibrando o método de adaptação de PConfig para que ele reflita as reais necessidades da organização. Além da avaliação parcial, também é realizada, ao final do projeto, uma avaliação final do

projeto que, assim como a avaliação parcial, também pode implicar em modificações de PConfig.

7. **Alimentação da Base de Processos.** A avaliação final do processo e o próprio processo adaptado, já com as melhorias introduzidas, são armazenados, juntamente com as características do projeto, na Base de Processos, para que o processo possa ser reutilizado futuramente.

## 4.2 Escopo do Trabalho

Inicialmente, no escopo deste trabalho, serão introduzidos os elementos do PConfig e do Modelo de Caracterização de Projetos necessários para a adaptação da disciplina Planejamento e Gerenciamento.

Vários trabalhos mostram a importância e o impacto do Planejamento e Gerenciamento de projetos no processo de software [50, 51, 52]. Esses trabalhos indicam a falta de um planejamento de qualidade como causa de grande parte dos fracassos em projetos de software.

Apesar das atividades de Planejamento e Gerenciamento serem de suma importância para a qualidade do produto e para o bom andamento do projeto, elas também podem acrescentar uma sobrecarga desnecessária ao processo. As atividades de Planejamento e Gerenciamento não resultam em um progresso tangível do projeto, sendo, por isso, referenciadas como parte das “atividades de *overhead*” do processo [16, 53]. Exemplos de atividades de *overhead* são apresentados por Royce [53]: preparação de planos, documentação, acompanhamento de progresso, avaliação de riscos, avaliação financeira, gerência de configuração, avaliação de qualidade, integração, teste, retrabalho, gerenciamento, treinamento, administração do negócio etc. Pode-se notar que grande parte dessas atividades faz parte do escopo da disciplina Planejamento e Gerenciamento. Por isso, pode-se atribuir boa parte do peso de um processo à quantidade e à complexidade das atividades e artefatos ligados ao Planejamento e Gerenciamento do projeto. Ainda segundo Royce “atividades de *overhead* incluem muitos esforços que agregam valor ao projeto, mas, em geral, quanto menos esforço for despendido nessas atividades, mais esforço pode ser gasto nas atividades produtivas. O objetivo da melhoria de processos é maximizar a alocação de recursos para atividades produtivas e minimizar o impacto de atividades de *overhead* em recursos como pessoal, computadores e cronograma”. Segundo Boehm [24], gastar um tempo excessivo no

planejamento gera uma alta probabilidade de fracasso por causa do tempo gasto nessa atividade e porque mudanças tornarão os planos obsoletos, causando atrasos. Esses atrasos podem causar perda de mercado para competidores mais fortes e ágeis.

Como está ligado tanto à qualidade do produto como ao peso do processo, adaptar a disciplina Planejamento e Gerenciamento às características de um projeto específico é essencial para a obtenção de um processo adequado a esse projeto, ou seja, um processo que garanta a qualidade desejada sem penalizar o projeto com um processo excessivamente pesado.

A adição de novas disciplinas, além de Planejamento e Gerenciamento, não deve trazer maiores conseqüências para o Modelo de Adaptação de Processos de Software, já que tanto a análise das características do Modelo de Caracterização de Projetos como a escolha, em PConfig, dos artefatos que farão parte do processo adaptado são feitas por disciplinas, de forma modular. O único ponto de preocupação é a interdependência entre artefatos de disciplinas diferentes, ou seja, quando o artefato de uma disciplina servir como entrada para uma atividade de uma disciplina diferente. Esse problema pode ser resolvido pela identificação de que artefatos de outras disciplinas são entradas ou saídas de atividades da disciplina Planejamento e Gerenciamento. Uma lista desses artefatos será fornecida no Capítulo 6.

O Modelo de Adaptação de Processos de Software parte do pressuposto que a organização já possui um processo padrão de desenvolvimento. Por esse motivo, faz-se necessário adotar um processo padrão como base para a descrição do funcionamento do modelo. Para servir como exemplo de processo padrão, foi escolhido o Rational Unified Process 2002 (RUP 2002) [16].

O RUP foi escolhido porque, além de ser largamente utilizado e possuir farto material de consulta disponível, possui duas características importantes: é, ao mesmo tempo, um processo abrangente e detalhado.

Por se tratar de um *framework* de processos bastante completo, o RUP pode ser utilizado em vários tipos de projeto. Muitos elementos do RUP, entretanto, podem, e devem, ser descartados ou simplificados em projetos mais simples. Essa necessidade de adaptar um processo complexo para ser utilizado em projetos mais simples assemelha-se ao problema que está sendo discutido neste trabalho: a adaptação de um processo padrão para um projeto específico. Em geral, na definição de um processo padrão, busca-se um processo que possa ser aplicado de forma adequada a projetos considerados



grandes ou complexos para a organização [12]. Esse processo padrão, é claro, deve ser adaptado para projetos mais simples. Assim, sob o aspecto da abrangência do processo, adotar o RUP como exemplo de processo padrão é uma escolha justificável.

A outra característica importante que motivou a escolha do RUP foi o alto grau de detalhamento de seus elementos. Durante a definição e desenvolvimento de um processo padrão, adquire-se um conhecimento aprofundado do processo. Como, no caso deste trabalho, optou-se por um processo já definido para servir como processo padrão, o conhecimento sobre o processo adotado teria que ser obtido, basicamente, da documentação disponível sobre o mesmo. Esse requisito também é atendido satisfatoriamente pelo RUP, que possui uma documentação bastante completa.

Um outro fator relevante para a escolha do RUP é que grande parte dos processos de software são, ou podem ser, descritos através de seus artefatos, atividades e papéis, assim como o RUP. Isso torna o Modelo de Adaptação de Processos de Software mais facilmente adaptável a outros processos de software.

### **4.3 Estratégias de Implantação do MAPS**

Um aspecto que é importante avaliar antes de implementar o MAPS em uma organização é a relação entre os benefícios esperados e o custo de implantação do MAPS. Deve-se ter sempre em mente que o MAPS pode não ter um desempenho muito bom nos primeiros projetos, quando a Base de Processos conterà poucas informações e os próprios componentes do Modelo, o PConfig em especial, podem não estar totalmente ajustados à organização.

O principal fator a ser considerado para determinar a relação custo/benefício da implantação do MAPS é a situação atual da organização, ou seja, se a organização possui um processo de adaptação e se está tendo problemas relacionados com a adaptação de processos de software. Diferentes situações sugerem estratégias de implantação diferentes, conforme é apresentado na Tabela 4-1.

Situação	Estratégia
1. A organização possui um processo de adaptação já estabelecido e que funciona satisfatoriamente, mas não contempla reuso e melhoria de processos.	1. Utilizar o processo de adaptação de processos da organização no lugar do PConfig.
2. A organização possui um processo de adaptação que funciona de forma parcialmente satisfatória, ou seja, que gera processos adaptados com algumas inadequações ou que exige um esforço grande de adaptação.	2. Realizar, em um ou mais projetos, de forma paralela, a adaptação utilizando o MAPS e avaliar se os resultados obtidos com o MAPS são melhores ou não, fazendo uso, por exemplo, da abordagem utilizada no Capítulo 7 para comparar a aderência de dois processos diferentes ao projeto. Ao longo dos projetos, pode-se ajustar o MAPS à organização até que ele atinja um nível satisfatório de eficiência para só então passar a utilizá-lo no lugar do processo de adaptação anterior.
3. A organização não adapta processos (utiliza sempre o processo padrão) ou utiliza um processo de adaptação que não funciona satisfatoriamente.	3. Se os problemas causados pela falta de um processo adaptado forem considerados graves, deve-se adotar o MAPS de imediato e melhorá-lo durante sua aplicação nos projetos. Se os problemas forem menos graves, pode-se adotar a estratégia 2.

**Tabela 4-1:** Estratégias para implantação do MAPS em uma organização

## 4.4 Automação do Processo

Para tornar a aplicação do Modelo de Adaptação de Processos de Software mais eficiente e menos custosa é essencial a utilização de ferramentas de apoio que automatizem parte do processo de adaptação.

Algumas atividades importantes para a adaptação de processos que podem ser automatizadas são a construção, publicação e armazenamento de processos. Algumas ferramentas já desenvolvidas, tanto na área acadêmica quanto na comercial, fornecem todas ou parte dessas funcionalidades. Exemplos dessas ferramentas são:

- Methodology Explorer [54];
- ProKnowHow [48];
- DefPro e AssistPro [1];
- Rational Process Workbench (RPW) [55].

Apesar da existência dessas ferramentas, seria importante o desenvolvimento de uma ferramenta específica para o MAPS, que automatizasse outras tarefas do processo de adaptação. Essa nova ferramenta poderia funcionar de forma integrada com uma das outras ferramentas citadas, adicionando, por exemplo, a capacidade de lidar com as

informações armazenadas na Base de Processos e a utilização do Modelo de Caracterização de Projetos para identificar projetos semelhantes.

## **4.5 Considerações**

Este capítulo apresentou uma visão geral do MAPS, Modelo de Adaptação de Processos de Software, descrevendo seus objetivos, estrutura e funcionamento.

Também foi definido o escopo do trabalho, além de outros aspectos relevantes relativos ao MAPS, como estratégias para sua adoção em organizações e oportunidades de utilização de ferramentas já existentes para automatizar parte do Modelo.

Os capítulos seguintes apresentam, de forma mais detalhada, os principais componentes do MAPS.



---

## Capítulo 5

# Comparação de Projetos e Reuso de Processos

---

Neste capítulo será descrita a forma como o Modelo de Adaptação de Processos de Software (MAPS) realiza a comparação de projetos e o reuso de processos. Para isso, serão apresentados os dois componentes do MAPS que, em conjunto, são responsáveis por essas atividades: o Modelo de Caracterização de Projetos e a Base de Processos. A estrutura do capítulo é a seguinte:

- A Seção 5.1 apresenta o Modelo de Caracterização de Projetos, um dos componentes do MAPS. As características consideradas para caracterizar projetos (quanto à disciplina Planejamento e Gerenciamento) são definidas e analisadas, incluindo níveis de classificação para cada característica (Seções 5.1.1 e 5.1.2), é apresentado um método para comparação de projetos com base nas características (Seção 5.1.3) e sugerida uma estratégia de reuso de processos (Seção 5.1.4). Além disso, são identificadas possibilidades de extensão e adaptação do Modelo de Caracterização de Projetos (Seção 5.1.5).
- A Seção 5.2 apresenta a Base de Processos, outro componente do MAPS. Além da estrutura da Base de Processos (Seção 5.2.1), é descrito o método de reuso de processos utilizando o Modelo de Caracterização de Projetos e a Base de Processos em conjunto (Seção 5.2.2) e a forma como os processos adaptados são avaliados (Seção 5.2.3).
- A Seção 5.3 conclui o capítulo.

## 5.1 Modelo de Caracterização de Projetos

O Modelo de Caracterização de Projetos tem por finalidade comparar projetos de desenvolvimento de software, com base em suas características, identificando em que aspectos eles diferem e em que aspectos eles se assemelham. A identificação de pontos em comum entre os projetos permite que decisões acertadas de um projeto sejam repetidas em projetos semelhantes. Da mesma forma, ações mal sucedidas podem ser reavaliadas, evitando que erros sejam repetidos.

No contexto do Modelo de Adaptação de Processos de Software (MAPS), o Modelo de Caracterização de Projetos permite que projetos façam uso de processos ou partes de processos utilizados com sucesso em projetos semelhantes.

Como foi inicialmente concebido como parte do MAPS, o Modelo de Caracterização de Projetos analisa os projetos com base nas características que causam maior impacto no processo de software. Nada impede, porém, que o Modelo de Caracterização de Projetos seja adaptado para outros propósitos, como, por exemplo, para realizar estimativas de projeto com base em projetos semelhantes. Também é possível estender o Modelo de Caracterização de Projetos de forma a melhorá-lo ou adequá-lo a uma determinada organização.

Para facilitar a comparação entre projetos e permitir um maior reuso de partes dos processos de software, o Modelo de Caracterização de Projetos faz uma análise dos projetos por disciplinas do processo e não comparando os projetos com base no processo completo, com todas as características consideradas ao mesmo tempo. Assim, o processo foi dividido nas seguintes disciplinas:

- Modelagem de Negócio;
- Requisitos;
- Análise e Projeto;
- Implementação;
- Testes;
- Implantação;
- Gerência de Configuração;
- Planejamento e Gerenciamento do Projeto.

Essas disciplinas, é claro, podem ser redefinidas dependendo da estrutura do processo padrão.

Como já foi dito, essa abordagem, além de facilitar a comparação, permite que um projeto reuse partes do processo utilizado em um outro projeto mesmo que esses dois projetos não sejam totalmente semelhantes.

### 5.1.1 Definição das Características

No Modelo de Caracterização de Projetos, as características do projeto estão divididas em três tipos: características de desenvolvimento, características restritivas e prioridades do projeto.

As características de desenvolvimento caracterizam o projeto levando em consideração apenas os aspectos ligados ao desenvolvimento do software, sem considerar fatores organizacionais, contratuais ou de mercado. As características de desenvolvimento que serão analisadas serão:

- Tamanho da equipe;
- Distribuição geográfica da equipe;
- Experiência da equipe;
- Criticidade do software;
- Tamanho do projeto.

As características restritivas relacionam-se a aspectos que restringem a livre utilização de um processo em um projeto, sejam esses aspectos de ordem técnica, gerencial ou de negócio. Exemplos de características restritivas são:

- Padrões adotados;
- Exigências contratuais;
- Ferramentas disponíveis;
- Cronograma;
- Orçamento.

As prioridades do projeto se referem a aspectos organizacionais e de mercado e refletem a expectativa da organização em relação ao projeto. A análise das prioridades do projeto será feita considerando a relação entre a qualidade do produto e o tempo de desenvolvimento necessário.

## 5.1.2 Análise das Características

Para cada característica, serão feitas algumas considerações gerais, seguidas da análise do impacto da característica na disciplina Planejamento e Gerenciamento, um comentário sobre outras disciplinas afetadas e uma classificação proposta do projeto quanto à característica. Vale salientar que as características apresentadas são aquelas que causam maior impacto no Planejamento e Gerenciamento do projeto, ou seja, são as características necessárias para caracterizar o projeto em relação à sua disciplina Planejamento e Gerenciamento. Outras características devem ser consideradas para caracterizar o projeto quanto às demais disciplinas, mas sempre agrupadas nos três tipos definidos: características de desenvolvimento, características restritivas e prioridades do projeto.

### Tamanho da Equipe

Uma importante função dos processos de software é coordenar as pessoas envolvidas no desenvolvimento. Portanto, é natural que equipes de tamanhos diferentes necessitem de processos diferentes.

O tamanho da equipe de desenvolvimento tem impacto direto na forma de comunicação entre os membros da equipe. Em equipes pequenas, as formas de comunicação informais são suficientes para uma boa coordenação da equipe. Quanto maior a equipe, porém, menor a eficácia desse tipo de comunicação e maior a necessidade de comunicações formais [25].

Comunicações informais são, em geral, menos custosas e mais efetivas que as formais [11]. Assim, o aumento do tamanho da equipe, e o conseqüente aumento do número de comunicações formais, causa um aumento no custo de coordenação do projeto. Esse é um importante motivo que leva muitas das metodologias ágeis a colocar uma equipe pequena como pré-requisito para que a metodologia possa ser utilizada de forma adequada [25].

Como principais exemplos de comunicações formais temos reuniões formais e documentos em geral. Reuniões e conversas informais, documentação em rascunhos ou quadros são exemplos de comunicações informais.

O tamanho da equipe afeta, principalmente, as disciplinas Planejamento e Gerenciamento e Gerência de Configuração, já que essas disciplinas são responsáveis



pela coordenação entre os membros da equipe. Indiretamente, o tamanho da equipe acaba influenciando também as outras disciplinas do processo.

### ***Planejamento e Gerenciamento***

O Planejamento e Gerenciamento do projeto é afetado pois o tamanho da equipe indica o número de pessoas que devem ser treinadas, assessoradas e monitoradas. Equipes pequenas podem permitir, por exemplo, que a atribuição de responsabilidades, ou seja, quem irá realizar cada atividade, seja feita de maneira informal, o que é inviável em equipes grandes.

Em [53], Royce descreve a necessidade de aumento do *overhead* de gerenciamento (planejamento, comunicação, coordenação, avaliação de progresso, revisões, administração) à medida que o tamanho da equipe aumenta (Tabela 5-1). Em equipes pequenas, a necessidade de documentar artefatos intermediários é baixa, o foco está nos artefatos técnicos e as atividades de planejamento e gerenciamento são realizadas de maneira informal. O aumento do tamanho da equipe causa uma maior necessidade de documentar os artefatos intermediários, uma maior ênfase nos artefatos de gerenciamento e um maior formalismo nas atividades de planejamento e gerenciamento.

<b>Aspecto</b>	<b>Equipe Pequena</b>	<b>Equipe Grande</b>
Fases do ciclo de vida	Fronteiras pouco claras	Transições entre fases bem definidas para sincronizar o progresso entre atividades concorrentes
Artefatos	Foco em artefatos técnicos Poucos artefatos de gerenciamento	Gerenciamento de mudanças de artefatos técnicos Maior importância dos artefatos de gerenciamento
Alocação de esforço	Maior necessidade de generalistas	Maior porcentagem de especialistas
Checkpoints	Muitos eventos informais para manter a consistência técnica	Alguns eventos formais Sincronização entre equipes pode durar alguns dias
Formalismo do gerenciamento	Planejamento, controle e organização do projeto informais	Planejamento, controle e organização do projeto formais
Automação	Mais ambientes <i>ad hoc</i> , coordenados por indivíduos	Infra-estrutura para manter ambiente consistente para toda a equipe Ferramentas para controle de configuração e mudanças

**Tabela 5-1:** Características de P&G de acordo com o tamanho da equipe (traduzido de [53])

### ***Outras Disciplinas***

Equipes grandes precisam de mecanismos mais complexos de Gerência de Configuração, já que se torna mais difícil controlar as alterações feitas em artefatos do projeto e a possibilidade de alteração simultânea de artefatos é maior.

De forma geral, como o tamanho da equipe influi na forma de comunicação da equipe, todas as disciplinas acabam sendo afetadas, já que os documentos do projeto (especificações, modelos, casos de teste, o próprio código-fonte etc.) terão que ser desenvolvidos de forma mais completa e precisa.

### ***Caracterização***

Alguns trabalhos [11, 57] apresentam classificações para tamanhos de equipes. Porém, como essas classificações foram feitas com base nas características da indústria de software norte-americana, teve que ser feita uma adequação dos números à realidade brasileira que, em geral, apresenta equipes de desenvolvimento menos numerosas.

A classificação proposta de projetos quanto ao tamanho da equipe, em número de pessoas, é:

1. Muito pequena: 1-6 pessoas
2. Pequena: 7-20 pessoas
3. Média: 21-50 pessoas
4. Grande: 51-100 pessoas
5. Muito grande: +100 pessoas

### **Distribuição Geográfica da Equipe**

O acelerado ritmo de globalização das empresas que temos presenciado tem efeitos na indústria de software. Empresas com sedes em várias cidades, e até mesmo em vários países, estão se tornando a norma. Também é cada vez mais comum que um projeto seja desenvolvido, em parceria, por mais de uma organização. Dessa forma, a existência de equipes de desenvolvimento distribuídas tende a ser cada vez mais comum, o que causa impacto direto no processo de desenvolvimento de software e afeta a produtividade da equipe [50].

Segundo Cockburn [11], as formas de comunicação mais efetivas são aquelas que enfatizam o contato pessoal, face a face, entre os membros da equipe. De fato, equipes geograficamente distribuídas, onde o contato pessoal é menos freqüente, tendem a sofrer mais com problemas de coordenação. Exemplos desses problemas são o

maior tempo necessário para resolver dúvidas e a dificuldade de detectar problemas causados por falta de entendimento ou má interpretação [56]. Por isso, equipes de desenvolvimento distribuídas, assim como acontece com equipes grandes, necessitam de formas de comunicação mais formais.

É comum, em equipes geograficamente distribuídas, que informações que poderiam ser transmitidas de maneira informal em equipes centralizadas sejam transmitidas através de documentos. A qualidade da documentação produzida, em termos de clareza, correção e riqueza de detalhes, deve ser maior, já que o esclarecimento de dúvidas e a percepção de mal-entendidos são mais difíceis.

### ***Planejamento e Gerenciamento***

O Planejamento e Gerenciamento do projeto é afetado, já que coordenar pessoas em diferentes locais apresenta maiores dificuldades, principalmente devido aos problemas de comunicação já mencionados. Serão necessários, para gerenciar o trabalho da equipe, um maior número de reuniões formais e uma maior dedicação ao controle de qualidade da documentação produzida.

### ***Outras Disciplinas***

A Gerência de Configuração também é afetada, já que garantir a consistência dos artefatos e controlar as mudanças realizadas em um ambiente de trabalho distribuído é uma tarefa complexa.

Indiretamente, todas as disciplinas são afetadas, já que a qualidade dos artefatos, em termos de clareza e correção, deve ser maior. Essa qualidade, como já foi dito, será monitorada através das atividades de Gerenciamento do Projeto.

### ***Caracterização***

A classificação proposta para a distribuição geográfica da equipe, em termos de localização dos membros da equipe, é (adaptado de [57]):

1. Mesma sala;
2. Mesmo prédio, salas diferentes;
3. Mesma cidade, mesma empresa, prédios diferentes;
4. Mesma cidade, empresas diferentes;
5. Cidades diferentes.

## **Experiência da Equipe**

A experiência da equipe de desenvolvimento é um fator importante na definição de um processo de desenvolvimento de software. Por se tratar de uma característica complexa, a experiência da equipe foi decomposta em três características distintas: experiência técnica, experiência no domínio da aplicação e experiência no processo de desenvolvimento. Espera-se, com essa divisão, facilitar a análise dessas características e determinar de forma mais precisa as disciplinas que elas afetam.

### ***Planejamento e Gerenciamento***

A experiência no processo de desenvolvimento afeta, principalmente, o Gerenciamento do Projeto. Uma equipe inexperiente no processo precisa ser acompanhada mais atentamente, possivelmente com mais mecanismos de controle. Esses mecanismos, no caso de uma equipe mais experiente, podem ser dispensáveis, tornando o processo mais ágil.

A experiência no domínio da aplicação e a experiência técnica, apesar de não terem impacto significativo na disciplina de Planejamento e Gerenciamento, impactam diretamente o trabalho do gerente do projeto. Esses fatores influenciam, por exemplo, o tempo alocado a cada tipo de atividade (requisitos, análise e projeto, implementação etc) e a duração das iterações [53]. A inexperiência da equipe pode, inclusive, fazer parte de uma eventual lista de riscos do projeto. Além disso, esses fatores exercem grande impacto em outras disciplinas do processo. A experiência no domínio da aplicação tem grande influência, por exemplo, na disciplina Requisitos, enquanto a experiência técnica influencia sobremaneira as disciplinas Análise e Projeto e Implementação. Assim, apesar de não serem consideradas para a caracterização do projeto quanto à disciplina Planejamento e Gerenciamento, a experiência no domínio da aplicação e a experiência técnica também serão classificadas, já pensando em um futuro melhoramento, para a disciplina de Planejamento e Gerenciamento, ou extensão, para outras disciplinas, do Modelo de Caracterização de Projetos.

### ***Outras Disciplinas***

A experiência no domínio da aplicação tem impacto relevante na disciplina Requisitos, já que espera-se que, para domínios conhecidos, os requisitos sejam elicitados de forma mais completa e correta, possibilitando um acompanhamento mais informal desses requisitos. A Análise e Projeto também é afetada, já que para domínios desconhecidos fazem-se necessários um maior número de modelos para garantir o

entendimento do problema.

A experiência técnica refere-se à experiência dos desenvolvedores com a tecnologia, com as ferramentas e com os paradigmas adotados. A principal disciplina afetada pela experiência técnica é a disciplina Implementação, já que as funcionalidades e as possibilidades de integração das ferramentas e as facilidades e dificuldades decorrentes da tecnologia utilizada já serão mais conhecidas em equipes experientes, exigindo menor esforço de implementação. A disciplina Análise e Projeto também pode ser afetada, já que a utilização de novas tecnologias e novos paradigmas pode mudar a forma de modelar o sistema (a transição de análise estruturada para análise orientada a objetos, por exemplo).

### ***Caracterização***

As classificações para as experiências no processo, no domínio da aplicação e técnica foram definidas da seguinte forma:

**Experiência no Processo** (número médio de projetos que utilizaram o processo de que os membros da equipe participaram):

1. Nenhum projeto
2. 1 projeto
3. 2 a 3 projetos
4. 4 a 5 projetos
5. Mais de 5 projetos

**Experiência no Domínio da Aplicação** (número médio de projetos no mesmo domínio da aplicação de que os membros da equipe participaram):

1. Nenhum projeto
2. 1 projeto
3. 2 a 3 projetos
4. 4 a 5 projetos
5. Mais de 5 projetos

**Experiência Técnica** (tempo médio de experiência dos membros da equipe com as principais tecnologias a serem utilizadas no projeto):

1. Nenhum projeto
2. 1 projeto
3. 2 a 3 projetos
4. 4 a 5 projetos

## 5. Mais de 5 projetos

### **Criticidade do Projeto**

A criticidade de um sistema define a gravidade das conseqüências de uma possível falha desse sistema. Sistemas críticos são aqueles que não podem falhar, sob pena de causar danos irreparáveis.

A criticidade do sistema exerce influência forte e abrangente no processo de desenvolvimento, já que está diretamente associada à qualidade do produto: quanto mais crítico, maior deve ser a qualidade do sistema.

Sistemas críticos requerem mais mecanismos de controle no processo de desenvolvimento ou, em outras palavras, requerem processos mais pesados. Esses mecanismos de controle, embora constituam um fator que afeta negativamente a produtividade, fazem-se necessários nesse tipo de desenvolvimento, tendo em vista que a qualidade do produto deve ser a prioridade do projeto.

Embora afete, de uma maneira ou de outra, todas as disciplinas do processo de desenvolvimento, a criticidade do projeto causa maior impacto no Planejamento e Gerenciamento do Projeto, nos Requisitos e nos Testes.

### ***Planejamento e Gerenciamento***

O Planejamento e Gerenciamento do Projeto é afetado na sua tarefa de realizar o controle de qualidade dos artefatos. A documentação de um sistema crítico, por exemplo, deve permitir uma perfeita compreensão do sistema, dando maior segurança no desenvolvimento e na manutenção. Para isso a documentação deve ser detalhada e precisa, além de estar sempre atualizada. Existe, também, a necessidade de um maior número de *checkpoints* formais para assegurar que o processo está sendo corretamente seguido e que o projeto está correndo de acordo com o planejado.

### ***Outras Disciplinas***

A disciplina Requisitos é impactada em termos de requisitos não funcionais, como segurança, confiabilidade, robustez etc. Esses requisitos devem ser especificados de forma precisa e é necessário assegurar que eles sejam satisfeitos.

A disciplina Testes de um sistema crítico deve ser a mais completa possível. Casos de teste bem documentados, realização de testes de regressão e alto grau de cobertura dos testes são alguns fatores que podem contribuir para que o sistema atinja o nível de confiabilidade desejado.

### ***Caracterização***

A classificação proposta de projetos quanto à criticidade, em termos de conseqüências de uma possível falha do sistema, é (adaptado de [57]):

1. Perda de conforto;
2. Prejuízos baixos, perdas facilmente recuperáveis;
3. Prejuízos moderados, perdas recuperáveis;
4. Prejuízos altos, perdas irrecuperáveis;
5. Risco de vida.

### **Tamanho do Projeto**

O tamanho do projeto é uma característica que afeta o processo de desenvolvimento de forma bastante abrangente, impactando diversas disciplinas do processo.

### ***Planejamento e Gerenciamento***

Estatísticas mostram que projetos de grande porte têm menores chances de sucesso que projetos pequenos [51, 52]. Uma solução indicada por essas pesquisas para aumentar as chances de sucesso de um projeto grande é o estabelecimento de um Planejamento e Gerenciamento de Projeto adequado. De modo geral, quanto maior o projeto, maior o número de atividades a serem realizadas, artefatos a serem produzidos e recursos a serem administrados. Isso implica que o Planejamento e Gerenciamento do Projeto deve propiciar maior controle sobre o projeto à medida que o tamanho do projeto aumenta.

Segundo pesquisa do Standish Group [52], um processo de gerenciamento de projetos formal é a única forma de obtenção de sucesso em grandes projetos. Entretanto, segundo a mesma pesquisa, um gerenciamento formal não agrega valor a projetos pequenos e simples. Ao contrário, para esse tipo de projeto, utilizar um processo formal de gerenciamento pode causar o fracasso do projeto devido ao tempo gasto nessa atividade.

### ***Outras Disciplinas***

Outra disciplina do processo afetada pelo tamanho do projeto é a Gerência de Configuração. Como, em geral, o número de artefatos produzidos é maior em projetos grandes, o formalismo da Gerência de Configuração também precisa ser maior nesses projetos.

O tamanho do projeto também afeta as disciplinas Requisitos, Análise e Projeto, Implementação e Testes, já que a tendência é que, em projetos maiores, existam mais requisitos, classes, componentes, casos de teste etc. A existência de muitas classes, por exemplo, pode inviabilizar técnicas mais informais de modelagem, como colocar o diagrama em um quadro ao invés de colocá-lo em um documento, conforme sugerido em algumas metodologias ágeis [58, 59].

### ***Caracterização***

Há várias formas de medir o tamanho de um projeto. Pode-se medir, por exemplo, o tamanho do produto através de métricas como pontos de função ou linhas de código. Pode-se medir o esforço de desenvolvimento através, por exemplo, de pessoas-mês. Neste trabalho, a medida adotada para medir o tamanho do projeto será o custo do projeto.

Apesar de não indicar de forma tão precisa o esforço de desenvolvimento e o tamanho do produto, embora esteja intrinsecamente relacionado a esses aspectos, o custo do projeto captura uma variável importantíssima, que é o investimento feito pela organização no projeto. Em projetos de maior investimento, é provável que a organização prefira adotar um processo mais preditivo, que possibilite um maior controle do projeto, ainda que para isso seja necessário sacrificar a produtividade.

Existem algumas classificações de projetos quanto ao custo, como a apresentada no relatório CHAOS [52]. Porém, esses números estão distantes da realidade brasileira. Por isso, foram utilizadas faixas com valores mais baixos, tornando a classificação mais adequada aos valores de projetos de software comumente encontrados na indústria de software nacional.

A classificação de projetos quanto ao seu tamanho foi definida da seguinte forma:

1. Até R\$ 50.000,00
2. Entre R\$ 50.000,00 e R\$ 150.000,00
3. Entre R\$ 150.000,00 e R\$ 1.000.000,00
4. Entre R\$ 1.000.000,00 e R\$ 3.000.000,00
5. Mais de R\$ 3.000.000,00

### **Características Restritivas**

Algumas características têm a propriedade de limitar a livre definição de um



processo de software na medida em que impõem restrições a esse processo. Por esse motivo, essas características serão denominadas características restritivas.

As características restritivas podem ser de ordem técnica, organizacional ou de negócio como, por exemplo, ferramentas de apoio utilizadas, padrões de qualidade adotados e exigências contratuais, respectivamente.

A seguir, serão apresentadas algumas características restritivas e a forma como elas interferem no processo de desenvolvimento de software:

- **Ferramentas:** o uso de ferramentas de apoio que automatizem parte do desenvolvimento pode viabilizar a realização de algumas atividades, mesmo que essas não sejam estritamente necessárias ao processo de desenvolvimento, devido ao baixo custo que esta atividade passará a ter. Por outro lado, a ausência de ferramentas de apoio podem fazer com que atividades que seriam recomendáveis do ponto de vista do processo não sejam realizadas por serem inviáveis do ponto de vista do projeto devido ao custo ou tempo necessário para realizá-las. A utilização de ferramentas de apoio também pode restringir a definição do processo na medida em que o processo tenha que se adequar à forma de trabalho das ferramentas.
- **Padrões adotados:** a utilização de padrões de qualidade, como CMM [15] e ISO [60], exigem que determinadas atividades sejam realizadas para manter o processo em conformidade com esses padrões. Uma organização que desenvolva software seguindo o padrão CMM nível 2, por exemplo, não pode abrir mão de ter uma Gerência de Requisitos nos moldes exigidos pela norma, ainda que algumas atividades não sejam realmente necessárias a um projeto em particular.
- **Exigências contratuais:** o contrato de desenvolvimento pode exigir a produção de determinados artefatos que, analisando o projeto apenas sob os aspectos técnico e gerencial, não seriam realmente necessários. No que se refere à disciplina Planejamento e Gerenciamento, exigências contratuais aumentam a complexidade gerencial [53]. Se não existir um contrato, ou se o contrato for maleável, o processo tende a ser mais flexível, já que o conjunto de funcionalidades do software, o cronograma, o orçamento e o nível de qualidade do produto podem ser alterados com menor esforço do que no caso de existir um contrato rigoroso.

- **Cronograma:** um cronograma agressivo pode impedir que um processo mais formal, e por consequência mais custoso, seja adotado de forma integral. Segundo Yourdon [61], a melhor forma de adequar um processo a esse tipo de cronograma é estabelecer a formalização de algumas atividades críticas e deixar que as demais atividades sejam realizadas de maneira informal. Em suma, o projeto ideal do ponto de vista das características de desenvolvimento pode não ser viável por conta de restrições do cronograma.
- **Orçamento:** projetos que trabalham com um orçamento limitado podem sofrer sérias limitações com relação ao seu processo de desenvolvimento. Limitações orçamentárias podem impedir que atividades importantes do processo sejam realizadas por falta de recursos para contratar profissionais ou consultoria, adquirir hardware e software necessários, impossibilidade de prolongar o desenvolvimento por falta de recursos para manter o projeto etc.

### **Prioridades do Projeto**

As características analisadas anteriormente nos permitem determinar adaptações necessárias ao processo de software para que ele se adeqüe ao projeto e restrições a essas adaptações. Um fator importante, contudo, não pode ser analisado com base apenas nas características de desenvolvimento e restritivas: qual a expectativa da organização desenvolvedora em relação ao projeto? Em outras palavras, é preciso analisar quais são as prioridades do projeto.

Neste trabalho, as prioridades do projeto serão analisadas com base na relação entre a qualidade desejada do produto e o tempo necessário para que esse produto chegue ao cliente, seja ele uma empresa contratante, a própria organização desenvolvedora ou qualquer outro tipo de mercado consumidor.

Embora a relação entre qualidade e tempo de desenvolvimento seja, em parte, determinada por características como criticidade do sistema, cronograma e exigências contratuais, ela pode ser produto de uma estratégia organizacional, sendo afetada por fatores do mercado e da própria organização. Pode-se requerer que um software, mesmo que ele não seja crítico e que não existam exigências contratuais, possua o maior nível de qualidade possível. Isso pode se dever a inúmeros fatores como, por exemplo:

- a organização pode querer associar a sua imagem a produtos de alta qualidade;

- o nicho de mercado do software pode ser muito concorrido, exigindo produtos de alta qualidade para manter a competitividade;
- o desenvolvimento de um produto de qualidade pode gerar projetos futuros com o contratante.

Por outro lado, pode-se desejar que um projeto seja finalizado o mais rápido possível, sacrificando a qualidade, mesmo que não exista um cronograma agressivo ou, até mesmo, um prazo definido para o término do projeto:

- a organização desenvolvedora pode querer ganhar uma fatia de um mercado em expansão ou ser pioneira em um novo nicho de mercado;
- o produto pode precisar de adaptações urgentes devido a mudanças no ambiente onde ele está inserido;
- a organização pode necessitar urgentemente dos recursos financeiros que serão gerados com o término do projeto;
- os recursos alocados ao projeto devem ser remanejados para projetos mais importantes.

Como pode ser visto por alguns dos exemplos acima, o tipo de desenvolvimento é um fator importante na definição das prioridades do projeto. Enquanto softwares desenvolvidos por contrato exigem maior qualidade, softwares desenvolvidos de forma especulativa (software de prateleira) são frequentemente desenvolvidos rapidamente, em detrimento da qualidade do produto [6].

As prioridades do projeto impõem adaptações ao processo de desenvolvimento. Em projetos cuja prioridade é a entrega rápida do produto, processos ágeis podem ser a melhor opção, já que focam mais no desenvolvimento e menos em atividades de controle e garantia da qualidade. Em projetos onde a qualidade seja priorizada, é importante contar com mecanismos de controle, documentação mais completa, processos mais preditivos.

Uma característica peculiar das prioridades do projeto é que, ao contrário das características de desenvolvimento, elas não afetam apenas disciplinas específicas do processo, mas sim o processo como um todo. Tanto a qualidade do produto quanto o tempo de desenvolvimento afetam o rigor dos testes e da gerência de configuração, a qualidade dos modelos de análise e projeto e das definições de requisitos, as decisões de implementação, o nível de controle em termos de gerenciamento do projeto etc.

Por estarem associadas a todo o processo de desenvolvimento e por darem uma

visão mais geral do projeto, as prioridades do projeto podem ser usadas como diretrizes na adaptação do processo, funcionando como fator de decisão no caso de características conflitantes, conforme será visto no Capítulo 6.

A classificação proposta para a prioridade do projeto foi elaborada com base em uma relação entre a qualidade e o tempo de desenvolvimento. Foram distribuídos 100 pontos entre qualidade e tempo de desenvolvimento, de acordo com aquilo que for mais relevante para o projeto. Os extremos foram desconsiderados porque, na prática, é extremamente raro que em um projeto o tempo ou a qualidade não tenham relevância nenhuma. O projeto deve ser classificado de acordo com a situação que mais se aproxime da sua realidade. A classificação proposta é a seguinte:

1. Qualidade = 90, Tempo = 10
2. Qualidade = 70, Tempo = 30
3. Qualidade = 50, Tempo = 50
4. Qualidade = 30, Tempo = 70
5. Qualidade = 10, Tempo = 90

### 5.1.3 Método de Comparação

A comparação entre projetos de software, no Modelo de Caracterização de Projetos, é feita analisando os níveis de classificação das características que impactam cada disciplina do processo.

A comparação é feita disciplina a disciplina e não entre projetos como um todo. Assim, dois projetos podem ser semelhantes em relação a uma disciplina e diferentes em relação a outra.

Para a comparação de projetos, serão consideradas apenas as características de desenvolvimento. As características restritivas e as prioridades do projeto serão consideradas na fase de reuso dos processos, conforme será detalhado na Seção 5.2.

O grau de semelhança  $S_D$  entre dois projetos P e P', em relação a uma disciplina D, é dado por:

$$S_D = 1 - \frac{\sum_{i=1}^m |n_i - n_i'|}{m}$$

onde:

- $m$  é o número total de características que impactam D;

- $n_i$  e  $n_i'$  são os níveis de classificação dessas características em P e P', respectivamente;
- $|n_i - n_i'|$  representa a diferença de níveis para cada característica de P e P'.

Na fórmula acima, o somatório representa a diferença entre os projetos, sendo zero quando os projetos forem totalmente semelhantes. Esse somatório nos dá a diferença total, expressa em número de níveis, entre dois projetos em relação a determinada disciplina do processo, levando em conta todas as características que impactam essa disciplina.

A divisão por  $m$  faz-se necessária porque um mesmo valor do somatório pode significar graus de diferença bastante distintos, dependendo do número de características que estiverem sendo consideradas. Tome-se, por exemplo, uma disciplina que seja impactada por apenas uma característica. Se o valor do somatório for “2”, por exemplo, isso significa que os projetos comparados são muito diferentes em relação à disciplina considerada, já que, para a única característica que impacta a disciplina, existe uma diferença de dois níveis entre os projetos. Por outro lado, se o valor “2” do somatório tivesse sido encontrado para uma disciplina que fosse impactada por quatro características, os projetos comparados seriam relativamente semelhantes, já que a diferença média entre os projetos seria de apenas 0,5 nível/característica. Para corrigir essa distorção, optou-se por dividir o somatório pelo número  $m$  de características analisadas, tornando relativo o valor do somatório.

Analisando os possíveis valores de  $S_D$ , temos as seguintes situações:

- Caso 1:  $S_D = 1$  → projetos totalmente semelhantes em relação a D
- Caso 2:  $0,5 \leq S_D < 1$  → projetos muito semelhantes em relação a D
- Caso 3:  $0 \leq S_D < 0,5$  → projetos pouco semelhantes em relação a D
- Caso 4:  $S_D < 0$  → projetos não semelhantes em relação a D

#### 5.1.4 Estratégia Recomendada para Reuso de Processos

A comparação entre projetos tem como principal objetivo permitir que processos utilizados em projetos anteriores sejam reutilizados em projetos semelhantes. Para decidir o processo que será reusado deve-se determinar, através do método de comparação, o projeto mais semelhante ao projeto atual em relação a cada disciplina.

Vale salientar que tanto as informações sobre os projetos quanto as descrições dos processos utilizados estão contidas na Base de Processos, descrita na Seção 5.2, juntamente com avaliações do processo e sugestões de melhoria que devem, sempre que possível, ser incorporadas ao novo processo.

A abordagem recomendada de reuso dos processos para cada caso de semelhança descrito na seção anterior é:

- **Caso 1:** reuso direto das partes do processo relativas à disciplina em questão.
- **Caso 2:** reuso das partes do processo relativas à disciplina em questão com adaptações. As adaptações devem ser definidas com base nas diferenças entre os projetos percebidas na comparação e no sentido de tornar o processo mais ou menos formal de acordo com essas diferenças.
- **Caso 3:** a definição do novo processo deve ser feita a partir da comparação entre o processo já utilizado, que está sendo reusado, e o processo padrão da organização. Apesar da pouca semelhança entre os projetos, algumas decisões tomadas no projeto anterior podem se aplicar ao novo projeto.
- **Caso 4:** a definição do novo processo deve ser feita a partir do processo padrão da organização. As possibilidades de reuso são pequenas.

Deve ficar claro que a comparação de projetos, por considerar apenas as características de desenvolvimento, leva ao reuso de processos adequados apenas do ponto de vista de desenvolvimento. Dessa forma, o reuso dos processos pode sofrer restrições das características restritivas e das prioridades do projeto. Um processo, mesmo tendo se mostrado eficiente em situações semelhantes anteriores, talvez não possa ser reusado livremente por conta de restrições orçamentárias, contratuais ou de mercado.

### 5.1.5 Extensão e Adaptação do Modelo

O Modelo de Caracterização de Projetos pode ser estendido ou adaptado para propósitos diferentes da adaptação de processos ou de acordo com as características da organização. As modificações do Modelo de Caracterização de Projetos podem ser feitas incluindo e retirando características ou alterando os limites entre os níveis de classificação das características.

A escolha das características dos projetos que causam maior impacto no processo de software, em especial a disciplina Planejamento e Gerenciamento, foi feita

através do estudo de diversos trabalhos, idéias, relatos e opiniões presentes na literatura de Engenharia de Software. Deve-se reconhecer, entretanto, que muitas outras características também impactam o processo de software. A opção por um número limitado de características analisadas deve-se à tentativa de tornar a caracterização de projetos menos complexa, tanto para o desenvolvimento quanto para a aplicação do Modelo de Caracterização de Projetos. A expectativa é que, com a aplicação do Modelo de Caracterização de Projetos em um número razoável de projetos, sejam identificadas oportunidades de inclusão e remoção de algumas características.

A inclusão e remoção de características pode ser necessária para que o Modelo de Caracterização de Projetos reflita os aspectos que são importantes para o desenvolvimento na organização. Exemplificando, uma organização pode não precisar analisar a distribuição geográfica da equipe de desenvolvimento por concentrar todas as suas atividades em um mesmo local. Por outro lado, uma organização que trabalha muito com subcontratação de serviços pode desejar incluir uma ou mais características relacionadas a esse aspecto do desenvolvimento. Na inclusão de características, devem ser informados as disciplinas afetadas e a forma de classificação de projetos de acordo com a característica.

A alteração dos limites dos níveis de classificação pode ser necessária para adequar o Modelo de Caracterização de Projetos aos projetos e características da organização. Em uma empresa pequena, por exemplo, pode não ser necessário classificar equipes de tamanho maior que 30 pessoas. Os níveis de classificação para o tamanho da equipe podem ser redefinidos para refletir a realidade da organização.

## 5.2 Base de Processos

Adaptar processos de software para projetos específicos, analisando os vários aspectos do processo e do projeto, é uma tarefa complexa, exigindo grande esforço e conhecimento do Engenheiro de Processos. Por isso, os modelos, métodos e ferramentas para a adaptação de processos devem apresentar formas de lidar com essa complexidade, tornando a tarefa de adaptar processos mais simples e menos custosa.

O MAPS, Modelo de Adaptação de Processos de Software, busca um melhor aproveitamento do conhecimento adquirido durante a adaptação de um processo padrão para um projeto específico com o objetivo de tornar menos complexa a adaptação para projetos subsequentes. No MAPS, o conhecimento adquirido durante a adaptação do

processo padrão para um projeto é armazenado na Base de Processos e pode ser reusado em projetos semelhantes, diminuindo bastante o esforço de adaptação do processo.

### 5.2.1 Estrutura da Base de Processos

As informações armazenadas na Base de Processos são:

- Informações sobre os projetos passados e suas características;
- O processo de software utilizado em cada projeto, especificando as seguintes informações:
  - **Artefatos Selecionados:** são artefatos do processo padrão que, a partir da aplicação do MAPS, foram selecionados e utilizados no processo adaptado. Devem estar contidas na Base de Processos, também, qualquer modificação no artefato original (como o desenvolvimento de uma versão simplificada) sugerida pelo MAPS;
  - **Artefatos Não Selecionados:** são artefatos do processo padrão que não serão utilizados no processo adaptado por terem sido definidos como dispensáveis pelo MAPS;
  - **Artefatos Incluídos:** são artefatos que, apesar de não fazerem parte do processo originalmente sugerido pelo MAPS, precisam ser incluídos no processo devido a alguma característica restritiva ou à análise do Engenheiro de Processos. O motivo da inclusão do artefato deve estar contido na Base de Processos;
  - **Artefatos Excluídos:** são artefatos que, apesar de fazerem parte do processo originalmente sugerido pelo MAPS, precisam ser excluídos do processo devido a alguma característica restritiva ou à análise do Engenheiro de Processos. O motivo da exclusão do artefato deve estar contido na Base de Processos.

Os artefatos utilizados no processo adaptado são obtidos da seguinte forma:

***Artefatos Utilizados = Artefatos Selecionados + Artefatos Incluídos***

Já os artefatos do processo originalmente sugerido pelo MAPS, sem levar em consideração as características restritivas, são obtidos da seguinte forma:

***Artefatos MAPS = Artefatos Selecionados + Artefatos Excluídos***

Os artefatos do processo padrão não utilizados no processo adaptado podem ser



obtidos da seguinte maneira:

$$\text{Artefatos Não Utilizados} = \text{Artefatos Não Selecionados} + \text{Artefatos Excluídos} - \text{Artefatos Incluídos}$$

A estrutura da Base de Processos está representada na Figura 5-1 em forma de Diagrama de Classes UML.

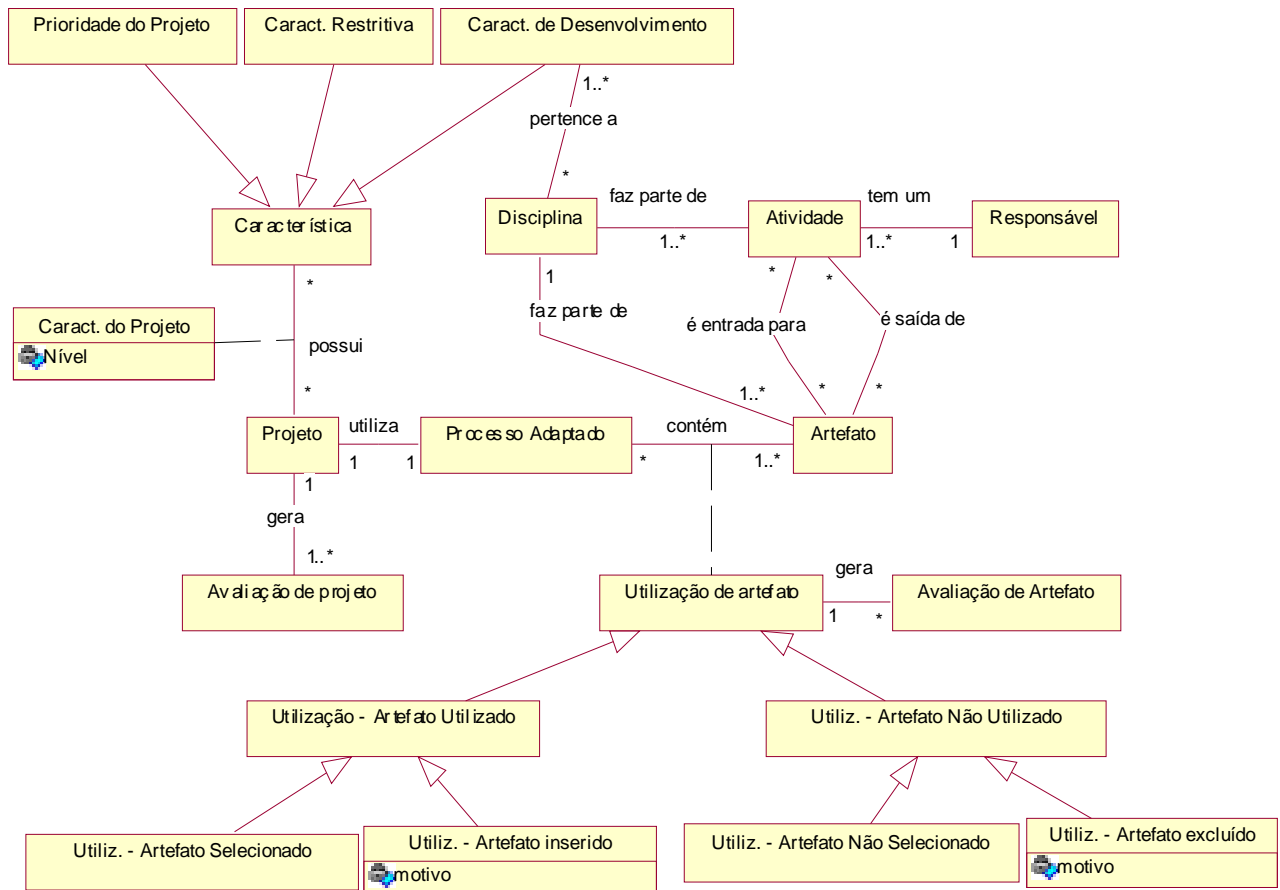


Figura 5-1: Modelo de classes da Base de Processos

A utilização de ferramentas para automatizar o processo de adaptação definido pelo MAPS pode alterar a estrutura do Modelo, especialmente da Base de Processos. Como exemplo, tomemos o Rational Process Workbench (RPW) [55] como ferramenta a ser utilizada.

O Rational Process Workbench é uma ferramenta que visa auxiliar a tarefa de customização do Rational Unified Process. Desenvolvido pela Rational Software Corporation, o RPW fornece mecanismos para a modelagem de processos através de extensões da UML (Unified Modeling Language) [43], implementadas através de estereótipos. A modelagem de processos deve ser feita através do Rational Rose, ferramenta para modelagem visual à qual o RPW se integra sob a forma de *add-in*. O

RPW também é capaz de analisar o processo modelado, verificando a existência de inconsistências como, por exemplo, a ausência de um artefato necessário para realizar uma atividade do processo. Além disso, o RPW permite que os processos customizados sejam publicados na forma de páginas HTML.

Como cada processo modelado no RPW contém todas as informações sobre sua estrutura (artefatos, atividades etc) e é armazenado em um arquivo separado (arquivo .cat), a Base de Processos poderia ser implementada de forma mais simples, armazenando informações apenas sobre o projeto e os artefatos, e suas respectivas avaliações, e fazendo referência ao arquivo com o modelo do processo utilizado e ao local onde as páginas HTML correspondentes estão armazenado. Para isso, é claro, teria que ser estabelecido um critério onde os locais de armazenamento do modelo e das páginas HTML do processo já estejam pré-estabelecidos.

Nesse caso, a estrutura da Base de Processos seria a mostrada na Figura 5-2. Pode-se notar que a única diferença em relação à figura anterior é a ausência das classes Disciplina, Atividade e Responsável que, neste caso, seriam gerenciadas pela ferramenta, e não pela Base de Processos.

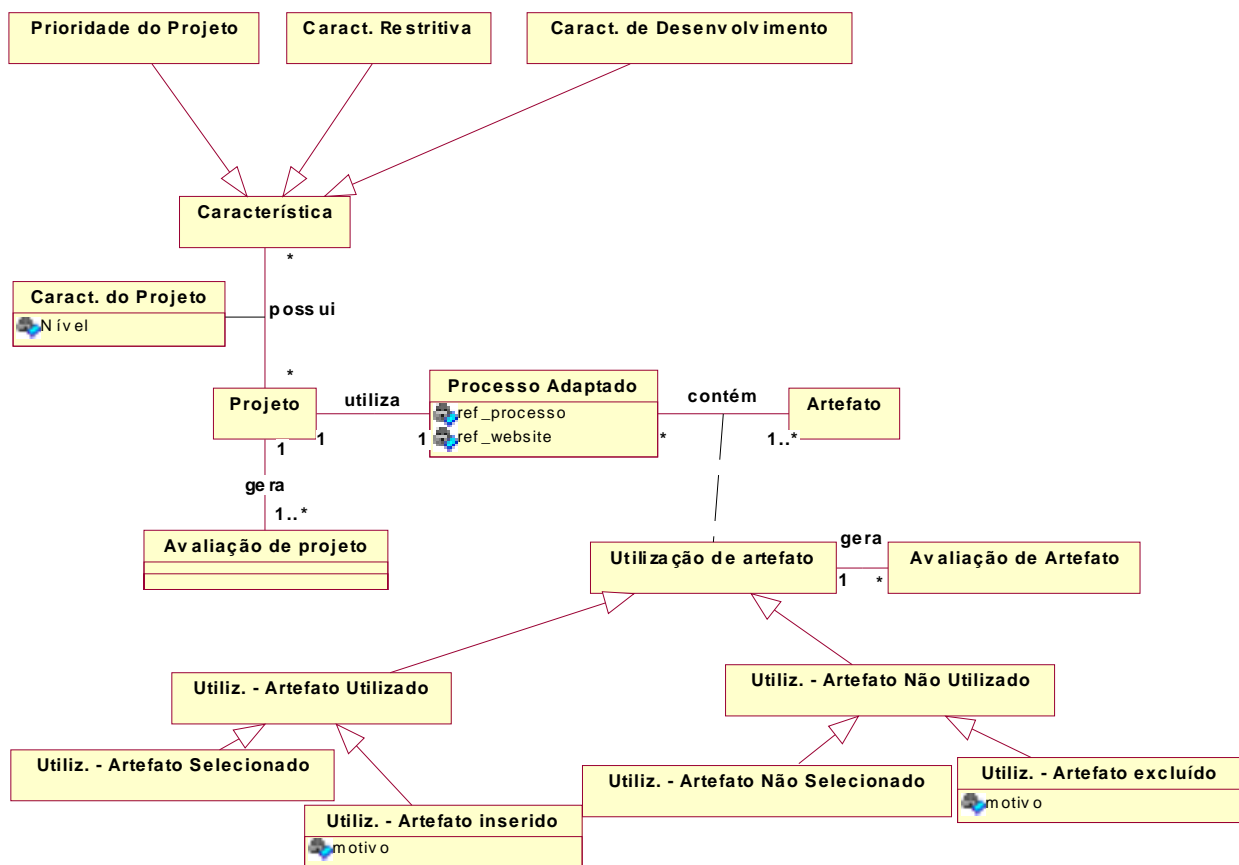


Figura 5-2: Modelo de classes modificado da Base de Processos

## 5.2.2 Reuso de Processos

O reuso de processos é feito através da integração entre a Base de Processos e o Modelo de Caracterização de Projetos. A Base de Processos é o repositório de conhecimentos, onde informações sobre projetos passados, juntamente com os processos utilizados e as avaliações desses processos, estão disponíveis. Já o Modelo de Caracterização de Projetos provê um método para selecionar, dentro da Base de Processos, as informações relevantes para um projeto específico.

Quando surge um novo projeto a ser desenvolvido, são selecionadas, na Base de Processos, as informações sobre os projetos passados que possuem maior semelhança com o projeto atual de acordo com o método de comparação previsto no Modelo de Caracterização de Projetos (Seção 5.1.3). Essas informações são submetidas à avaliação do Engenheiro de Processos para que ele decida que processos devem ser reusados, podendo utilizar a estratégia de reuso proposta pelo MAPS (Seção 5.1.4).

Alguns aspectos devem ser levados em consideração pelo Engenheiro de Processos na avaliação dos processos:

- A comparação entre projetos leva em consideração apenas as características de desenvolvimento dos projetos. Assim, cabe ao Engenheiro de Processos observar as prioridades dos projetos e as características restritivas que ele julgar relevantes para escolher o melhor processo a ser reusado.
- Devem ser observados os artefatos incluídos e excluídos. Como esses artefatos são adicionados ou retirados do processo por conta das características restritivas do processo, o Engenheiro de Processos deve avaliar se essas restrições aplicam-se ou não ao projeto atual.

## 5.2.3 Avaliação do Processo

A avaliação dos processos de software utilizados é de suma importância para uma maior eficiência do MAPS. As avaliações dos processos estão divididas em dois grupos: as avaliações parciais do processo e as avaliações finais do processo.

As avaliações parciais do processo podem ocorrer diversas vezes durante o projeto. Podem ser realizadas, por exemplo, ao final de cada iteração. O principal objetivo das avaliações parciais do processo são:

- Melhorar o processo que está sendo utilizado no projeto. O processo adaptado pode não estar funcionando a contento, requerendo melhorias imediatas.
- Melhorar o processo de adaptação de processos e o processo padrão da organização. Avaliações intermediárias do processo podem identificar problemas na adaptação de processos ou no processo padrão da organização que podem ser resolvidos antes que outros projetos passem pelas mesmas dificuldades.

Para que seja útil tanto para o projeto atual como para o processo de adaptação, as avaliações parciais serão realizadas, basicamente, avaliando os artefatos do processo. A avaliação deve englobar tanto os artefatos que estão sendo utilizados no projeto quanto aqueles que fazem parte do processo padrão mas não estão presentes no processo adaptado. Um modelo de avaliação parcial pode ser visto no Apêndice B.

A avaliação final do processo é feita somente uma vez, ao final do projeto. Ela não traz benefícios diretos para o projeto, mas serve como uma última análise do processo utilizado. A avaliação final do processo é bastante semelhante às avaliações parciais. Também é feita através da avaliação dos artefatos do processo e pode gerar melhorias para o processo padrão da organização e para o processo de adaptação de processos. Além disso, a avaliação final do processo será armazenada na Base de Processos, juntamente com as características do projeto e com o processo ultimamente utilizado (já com as melhorias feitas durante o projeto). O Apêndice C apresenta um modelo de avaliação final do projeto.

## 5.3 Considerações

Este capítulo apresentou dois dos principais componentes do MAPS: o Modelo de Caracterização de Projetos e a Base de Processos. No contexto do Modelo de Caracterização de Projetos, foram identificadas as principais características dos projetos de software que impactam o processo de planejamento e gerenciamento do projeto.

Foi descrito, também, como o Modelo de Caracterização de Projetos e a Base de Processos contribuem para as atividades de comparação de projetos e reuso de processos. Realizadas de forma integrada, essas duas atividades tornam a adaptação de processos mais eficaz, através da obtenção de um melhor processo

adaptado, e eficiente, através da diminuição do esforço necessário para realizar a adaptação.

A comparação de projetos e o reuso de processos, apesar de serem atividades extremamente úteis, não são suficientes para realizar, de forma completa, a adaptação de processos. Por isso, para completar o MAPS, será apresentado, no próximo capítulo, o PConfig, outro componente essencial do MAPS.



---

## Capítulo 6

# PConfig: Um Processo para Configuração de Processos

---

Neste capítulo, será descrito o PConfig, que é um processo para configuração de processos baseado em artefatos. Este capítulo está organizado da seguinte forma:

- Na Seção 6.1 será apresentada uma visão geral do PConfig, explicando sua função e seus objetivos.
- A Seção 6.2 fala sobre alguns conceitos e métodos presentes na literatura e utilizados no PConfig.
- A Seção 6.3 apresenta o processo de configuração propriamente dito, descrevendo cada um dos seus passos.
- A Seção 6.4 analisa os artefatos de Planejamento e Gerenciamento do RUP [16], utilizado como exemplo de processo padrão. Para cada artefato é feita uma breve descrição e uma análise de que características o influenciam.
- A Seção 6.5 mostra, em forma de tabelas, a relação entre as características do projeto e a necessidade de produzir cada artefato do processo.
- A Seção 6.6 apresenta alguns cuidados e dificuldades esperadas na implementação de PConfig.
- A Seção 6.7 apresenta uma breve conclusão do capítulo.

## 6.1 Visão Geral

A utilização da Base de Processos e do Modelo de Caracterização de Projetos, apesar de promover o reuso de processos, não é suficiente para realizar a adaptação de processos.

As primeiras adaptações terão como dificuldade adicional a falta de informações armazenadas. Somente com a utilização do Modelo de Adaptação de Processos de

Software (MAPS) em alguns projetos da organização é que o reuso de processos passará a ser realmente efetivo.

Mesmo após a organização haver construído uma base de conhecimento razoável, ainda é provável que projetos com características diferentes dos já realizados e catalogados pela organização venham a surgir.

Por isso, faz-se necessário um mecanismo para realizar a adaptação de processos para projetos sem similares anteriores e para completar processos onde apenas algumas disciplinas puderem ser reusadas de projetos anteriores. Essa é a função do processo configurador de processos: PConfig.

PConfig é um processo de configuração de processos de software para projetos específicos. Baseia-se na escolha dos artefatos do processo padrão que farão parte do processo adaptado de acordo com as características do projeto. A partir desses artefatos, serão derivados os papéis e atividades necessários.

Diferentemente do Modelo de Caracterização de Projetos, o PConfig é dependente do processo padrão que a organização utiliza. Apesar de a idéia geral poder ser aplicada a um conjunto maior de processos, uma instância específica do PConfig só pode ser aplicada a um único processo padrão. Essa limitação deve-se ao fato do PConfig estar intrinsecamente relacionado com os artefatos do processo. Assim, para um processo diferente, com artefatos diferentes, uma nova versão do PConfig deve ser produzida. É claro que, para processos relativamente semelhantes, duas versões do PConfig podem ser extremamente parecidas, exigindo um menor esforço de adaptação. Como o MAPS, a princípio, será aplicado em organizações com um processo padrão definido, a dificuldade de produzir uma nova versão do PConfig deve ocorrer apenas na implantação do MAPS ou no caso de uma grande mudança no processo padrão da organização.

A existência de processos especializados, ou seja, processos derivados do processo padrão que possuem adaptações específicas para determinados grupos de projetos (aplicações Web, sistemas embarcados, software para dispositivos portáteis etc), modifica a aplicação do PConfig. Nesse tipo de situação, deve existir uma versão do PConfig para cada um dos processos especializados, já que esses processos são diferentes entre si. Como, em geral, todos os processos especializados são gerados a partir do processo padrão, uma versão de PConfig produzida para um processo especializado pode, em grande parte, ser reutilizada para desenvolver as versões para os demais processos.



Como estamos considerando o RUP como exemplo de processo padrão, a versão do PConfig apresentada aqui contemplará os elementos definidos no RUP, mais especificamente na sua disciplina Planejamento e Gerenciamento.

## 6.2 Conceitos e Métodos Utilizados

Conforme já foi dito, o PConfig baseia-se na escolha de artefatos do processo padrão que farão parte do processo adaptado e dos papéis e atividades necessários para a produção desses artefatos. A idéia de adaptar processos a partir dos artefatos produzidos não é nova. Além do próprio RUP [16], na sua disciplina de Ambiente, alguns outros trabalhos utilizam-se desse método de adaptação [34, 45]. A principal vantagem dessa forma de adaptação é que, em um primeiro momento, o trabalho de adaptação fica resumido à definição dos artefatos necessários ao projeto, sem haver preocupação com os demais elementos do processo.

Segundo Leffingwell [62], um artefato só deve ser utilizado se atender a pelo menos um dos seguintes critérios:

- O artefato comunica um acordo ou conhecimento importante em situações onde um tipo de comunicação mais simples, como a verbal, é impraticável, como no caso de equipes grandes ou distribuídas, ou criaria um risco muito grande para o projeto.
- O artefato permite que pessoas que entrem na equipe no decorrer do projeto adquiram o conhecimento necessário de forma mais fácil e rápida.
- O investimento no artefato trará um retorno a longo prazo, na manutenção do sistema, ou o artefato será utilizado em vários momentos durante o desenvolvimento.
- O artefato é imposto pela organização, pelo cliente ou por algum padrão adotado.

Leffingwell defende ainda que deve ser avaliado qual o menor grau possível de complexidade do artefato de forma que as necessidades do projeto, identificadas através dos critérios citados, sejam atendidas.

Está sendo considerado, como processo padrão, um *framework* de processos, o RUP. Ou seja, o processo padrão é estabelecido de forma que possa ser aplicado a projetos considerados grandes ou complexos para a organização. A utilização desse processo, sem uma prévia adaptação, em todos os projetos da organização é uma alternativa altamente

ineficiente do ponto de vista de custo e tempo, como já foi discutido no Capítulo 1. Nesse caso, Royce [53] sugere três abordagens para aumentar a eficiência do processo para uma situação específica:

1. Considerar um processo com  $n$  passos e melhorar a eficiência de cada um desses passos.
2. Considerar um processo com  $n$  passos e eliminar alguns passos desnecessários, transformando o processo em um processo com  $m$  passos, onde  $m < n$ .
3. Considerar um processo com  $n$  passos e utilizar maior concorrência nas atividades realizadas ou nos recursos aplicados em cada passo.

PConfig utiliza as duas primeiras abordagens para adaptar processos. A segunda abordagem é utilizada quando, ao se decidir pela simplificação ou pela não inclusão de um artefato do processo padrão em um processo específico, um conjunto de atividades é eliminado do processo. Em muitas ocasiões, entretanto, as atividades continuarão existindo, embora de forma simplificada. Nesse caso, estará sendo utilizada a primeira abordagem.

Para exemplificar a utilização das duas abordagens, tome-se, como exemplo, o artefato Avaliação da Iteração. Esse artefato é produto da atividade Avaliar Iteração. Caso seja decidido, em um determinado projeto, que não é necessária uma Avaliação da Iteração, a atividade Avaliar Iteração pode ser suprimida. Estará sendo utilizada a segunda abordagem, já que o número de elementos do processo estará sendo reduzido. Por outro lado, pode ser decidido que a Avaliação da Iteração continuará existindo, porém será apenas uma reunião informal, sem que um documento seja produzido. Nesse caso, a atividade Avaliar Iteração continuará existindo, mas será realizada de uma outra forma, mais simples. Estará sendo utilizada a primeira abordagem, já que a eficiência de alguns elementos do processo estará sendo melhorada para uma situação específica.

## 6.3 O Processo de Configuração de Processos

PConfig utiliza os conceitos e métodos descritos anteriormente para adaptar processos através da realização de uma série de passos:

**Passo 1:** Identificar os artefatos do processo padrão que são adaptáveis. Alguns artefatos podem não ser adaptáveis por precisarem atender a algum tipo de padrão (CMM, por exemplo) ou por serem considerados essenciais ao processo.

**Passo 2:** Para cada característica, fazer uma matriz com os níveis de classificação da característica e os artefatos da disciplina impactada, identificando, para cada nível, os artefatos que:

S – serão produzidos

N – não serão produzidos

R – serão produzidos com restrições (informalmente, apenas algumas seções do documento, apenas se houver necessidade durante o projeto etc)

I – indiferente. A característica não influi nesse artefato

A Tabela 6-1 exemplifica a matriz resultante para a característica Tamanho da Equipe:

Tamanho da Equipe					
Artefatos	1-6 pessoas	7-20 pessoas	21-50 pessoas	51-100 pessoas	+100 pessoas
Artefato 1	N	N	R	S	S
Artefato 2	S	S	S	S	S
Artefato 3	I	I	I	I	I
Artefato 4	N	N	N	N	S

**Tabela 6-1:** Níveis da característica tamanho da equipe versus artefatos

Os artefatos já incorporados ao processo preliminar, oriundos do reuso de processos anteriores, não precisam ser incluídos nesse passo, já que a prioridade é reaproveitar partes do processo que já tenham se mostrado efetivas na prática, evitando buscar novas soluções para problemas já solucionados.

**Passo 3:** Para cada característica, selecionar a coluna da matriz correspondente à característica do projeto. Com base na Tabela 6-1, caso o número de pessoas da equipe do projeto estivesse entre 7 e 20, a coluna selecionada seria a mostrada na Tabela 6-2.

	7-20 pessoas
Artefato 1	N
Artefato 2	S
Artefato 3	I
Artefato 4	N

**Tabela 6-2:** Coluna da matriz correspondente à característica tamanho da equipe do projeto

**Passo 4:** Para cada disciplina, fazer uma sobreposição das colunas selecionadas de cada característica que impacta a disciplina. Caso haja conflito em um artefato, dois caminhos podem ser tomados:

1. adotar uma solução intermediária (indicada pelo Engenheiro de Processos)
2. decidir com base nas características restritivas (padrões, orçamento, cronograma, ferramentas, contratos) e na prioridade do projeto. Essa decisão é responsabilidade do Engenheiro de Processos

Pelas descrições acima, pode-se notar a importância da intervenção do Engenheiro de Processos para decidir o caminho a ser seguido em relação ao artefato.

**Passo 5:** Completar a lista de artefatos. Caso algum artefato seja indiferente a todas as características, decidir se ele será ou não produzido.

**Passo 6:** Realizar adaptação do processo. De posse da lista de artefatos que serão produzidos, identificar as atividades necessárias para a produção desses artefatos e os papéis responsáveis pelas atividades.

**Passo 7:** Análise do Engenheiro de Processos. O Modelo de Adaptação de Processos de Software tem como objetivo auxiliar o Engenheiro de Processos e não substituí-lo. O conhecimento, experiência e talento do Engenheiro de Processos continua sendo um importante diferencial na adaptação de processos. Assim, depois do processo de adaptação concluído, o Engenheiro de Processos tem liberdade para modificar o processo adaptado, adicionando, modificando e removendo artefatos, atividades e papéis. A análise do Engenheiro de Processos cria a possibilidade de inovação na definição do processo, tornando mais equilibrada a relação entre o uso de conhecimentos passados e a produção de novos conhecimentos, conforme sugere Henninger [27].

## 6.4 Artefatos do RUP

Nesta seção, serão analisados os artefatos da disciplina Planejamento e Gerenciamento do RUP. Será feita uma descrição de cada artefato e algumas considerações sobre como e com base em que características do projeto esses artefatos podem ser adaptados ou eliminados.



**Figura 6-1:** Artefatos de Planejamento e Gerenciamento do RUP (traduzido de [16])

### 6.4.1 Plano de Desenvolvimento de Software

O Plano de Desenvolvimento de Software é um documento cujo objetivo é aglutinar todas as informações necessárias para o gerenciamento do projeto. No Plano de Desenvolvimento de Software podem estar contidos vários outros artefatos, inclusive de disciplinas diferentes de Planejamento e Gerenciamento:

- Plano de Iteração;
- Plano de Gerenciamento de Requisitos (da disciplina Requisitos);
- Plano de Medições;
- Plano de Aceitação do Produto;
- Plano de Gerenciamento de Configuração (da disciplina Gerenciamento de Configuração);
- Plano de Garantia da Qualidade;
- Plano de Resolução de Problemas;
- Plano de Gerenciamento de Riscos;
- Caso de Desenvolvimento (da disciplina de Ambiente);
- Diretrizes de Modelagem de Negócio, Interface com o Usuário, Modelagem de Casos de Uso, Projeto, Programação e Testes (da disciplina de Ambiente);
- Guia de Estilo de Manuais (da disciplina de Ambiente);

- Plano de Infra-estrutura, Plano de Documentação, Plano de Gerenciamento de Subcontratação, Plano de Melhoria do Processo (citados, mas não definidos pelo RUP);

Percebe-se que alguns artefatos são apenas citados pelo RUP, mas não são definidos. Esses artefatos não serão analisados. Os artefatos de Planejamento e Gerenciamento serão analisados separadamente e os artefatos de outras disciplinas não serão analisados por estarem fora do contexto deste trabalho. Dessa forma, o Plano de Desenvolvimento de Software que será analisado aqui fica resumido à seguinte estrutura:

- Visão Geral do Projeto
- Organização do Projeto
- Processo de Gerenciamento
  - Estimativas do Projeto
  - Plano de Projeto
  - Acompanhamento e Controle do Projeto

A seção Visão Geral do Projeto trata do escopo, objetivos e restrições do projeto, além de descrever os artefatos que serão disponibilizados externamente e a estratégia para evolução do Plano de Desenvolvimento de Software, ou seja, versões e critérios para revisão do documento.

A seção Organização do Projeto apresenta o organograma do projeto, grupos externos à equipe de desenvolvimento que desempenharão algum papel no projeto e papéis e responsabilidades associados ao processo de desenvolvimento do projeto.

A seção Processo de Gerenciamento inclui as estimativas de tempo e custo do projeto, um plano de projeto, composto de um planejamento das fases e iterações do projeto, objetivos das iterações, liberação de versões do produto, cronograma, orçamento e recursos do projeto e o processo de acompanhamento e controle do projeto (qualidade, cronograma, orçamento e relatórios).

O Plano de Desenvolvimento de Software é um artefato essencial, devendo estar presente em todos os projetos. É claro que, em projetos mais simples, existirá um número menor de informações contidas no Plano de Desenvolvimento de Software, o que tornará o artefato mais simples.

## 6.4.2 Estudo de Viabilidade

O objetivo do Estudo de Viabilidade é fornecer informações sobre o projeto para que se possa determinar sua viabilidade em termos de retorno de investimento.

O Estudo de Viabilidade descreve:

- O produto a ser desenvolvido;
- O contexto de negócio onde ele se insere;
- Os objetivos do produto;
- Previsões de retorno de investimento;
- Restrições impostas ao produto como, por exemplo, padrões, leis e relação com outros produtos.

No caso de softwares desenvolvidos por contrato, o contrato entre as partes pode ser visto como parte do Estudo de Viabilidade.

Segundo o próprio RUP [16], o formalismo recomendado para o Estudo de Viabilidade é diretamente proporcional ao investimento que será feito no projeto. O Estudo de Viabilidade pode variar de um documento com grande quantidade de detalhes, no caso de projetos mais complexos e caros, até uma mera comparação entre as estimativas do investimento a ser feito e do retorno a ser obtido, no caso de projetos simples. Essa análise do retorno de investimento pode, no caso de desenvolvimento por contrato, representar a diferença entre o custo do projeto e o valor previsto no contrato, ou seja, pode indicar o lucro da empresa desenvolvedora.

## 6.4.3 Plano de Iteração

O objetivo do Plano de Iteração é fornecer uma lista das atividades e tarefas a serem realizadas e dos artefatos a serem produzidos em uma dada iteração, incluindo o estabelecimento de marcos de referência. Essa lista descreve os recursos (tempo, recursos financeiros e recursos humanos) alocados a cada atividade ou tarefa, a ordem de realização e a dependência entre atividades e tarefas. Além disso, o Plano de Iteração contém critérios de avaliação da iteração e os casos de uso envolvidos na iteração.

O Plano de Iteração deve estar presente em todos os projetos. Em projetos mais simples, entretanto, a granularidade das atividades pode ser maior, tornando o plano menos complexo.

## 6.4.4 Avaliação da Iteração

O objetivo da Avaliação da Iteração é capturar os resultados da iteração, compará-los aos critérios de avaliação estabelecidos e avaliar possíveis melhorias e mudanças a serem feitas no desenvolvimento. A Avaliação da Iteração apresenta os objetivos alcançados na iteração, a conformidade com o plano estabelecido, os casos de uso e cenários implementados, uma análise dos resultados a partir dos critérios de avaliação estabelecidos, resultados dos testes, ocorrências externas (mudança de requisitos do usuário, mudança no ambiente de negócio, fatos relativos à organização desenvolvedora que afetem o projeto etc) e retrabalho que necessita ser feito nas iterações seguintes.

O artefato Avaliação da Iteração pode variar bastante, dependendo das características do projeto. Projetos com equipes grandes, geograficamente distribuídas ou com pouca experiência no processo de desenvolvimento, além de softwares críticos, precisarão de um documento estruturado para difundir mais fácil e corretamente os resultados da iteração, as mudanças a serem feitas e o retrabalho necessário nas iterações subsequentes. Uma equipe pequena e experiente, por sua vez, pode necessitar apenas de uma reunião informal ou de um documento bastante simples para uma avaliação consistente.

## 6.4.5 Avaliação de Status

O propósito da Avaliação de Status é fornecer informações sobre o progresso do projeto. São apresentados dados sobre recursos utilizados, riscos do projeto, progresso técnico, avaliação dos marcos de referência em relação às datas previstas e ações realizadas ou em andamento.

A necessidade de produzir Avaliações de Status é maior em projetos com alto investimento, onde o interesse em acompanhar o progresso do desenvolvimento e os custos do projeto é, em geral, maior. Avaliações de Status também são necessárias quando a equipe não tem experiência no processo de desenvolvimento, já que as Avaliações de Status podem detectar erros, falta de entendimento e possibilidades de melhoria do processo. Equipes grandes ou geograficamente distribuídas também podem se beneficiar de Avaliações de Status, pois, nesses casos, é mais difícil ter uma idéia geral do andamento do projeto sem que exista um artefato que aglutine todas as informações necessárias. Em projetos simples, com equipes pequenas e experientes, o progresso do projeto pode ser



feito apenas através das Avaliações de Iterações, sem a necessidade de Avaliações de Status.

O tamanho das iterações, apesar de não ser uma característica intrínseca do processo, mas uma consequência do planejamento do projeto, também tem relação com as Avaliações de Status. É recomendável, no caso de iterações longas, que existam avaliações intermediárias, além das Avaliações de Iteração. Essas avaliações intermediárias podem ser implementadas através de Avaliações de Status.

## 6.4.6 Plano de Resolução de Problemas

O objetivo do Plano de Resolução de Problemas é estabelecer o processo utilizado para comunicar, analisar e resolver problemas do projeto. Esse artefato indica os responsáveis pela análise e resolução de cada tipo de problema, além das ferramentas, técnicas e formas de documentação, acompanhamento e resolução dos problemas.

O Plano de Resolução de Problemas é particularmente útil quando o produto a ser desenvolvido é crítico e, por isso, cada modificação no produto, no projeto ou no processo deve ser minuciosamente avaliada e os erros e problemas encontrados devem ser devidamente corrigidos. Em equipes grandes, geograficamente distribuídas ou inexperientes no processo de desenvolvimento, um Plano de Resolução de Problemas pode ser importante para facilitar a comunicação e coordenar a equipe, além de permitir uma melhor compreensão do processo, evitando que abordagens inapropriadas de correção de erros causem problemas ao projeto.

Em projetos de software de baixa criticidade e com equipes pequenas e centralizadas, a estratégia de resolução de problemas pode ser definida de maneira informal ou *ad-hoc*, sem a necessidade de desenvolver um plano para esse propósito. Outra opção para tornar o processo menos pesado é reduzir o Plano de Resolução de Problemas a uma seção do Plano de Desenvolvimento de Software, eliminando a necessidade de produzir um documento completo.

## 6.4.7 Plano de Gerenciamento de Riscos

O Plano de Gerenciamento de Riscos descreve a estratégia para identificação, análise, documentação, mitigação, acompanhamento e controle dos riscos de um projeto. O Plano de Gerenciamento de Riscos determina as atividades de gerenciamento de riscos que

devem ser executadas, bem como os responsáveis por essas atividades, os recursos alocados para as atividades e os riscos que devem ser monitorados.

Assim como o Plano de Resolução de Problemas, o Plano de Gerenciamento de Riscos pode melhorar a comunicação e a coordenação entre os membros da equipe de desenvolvimento e garantir conformidade com o processo utilizado, sendo, por isso, indicado para equipes grandes, distribuídas e inexperientes. Projetos de alto investimento ou críticos, onde a ausência de uma estratégia de tratamento de riscos pode causar muitos prejuízos, também se beneficiam de um Plano de Gerenciamento de Riscos. Projetos simples, por sua vez, podem optar por uma abordagem mais ágil, utilizando apenas a Lista de Riscos para fazer o gerenciamento de riscos. Assim como acontece com o Plano de Resolução de Problemas, existe a opção de tornar o processo menos pesado reduzindo o Plano de Gerenciamento de Riscos a uma seção do Plano de Desenvolvimento de Software, eliminando a necessidade de produzir um documento completo.

### **6.4.8 Lista de Riscos**

A Lista de Riscos é um artefato que descreve em ordem decrescente de magnitude os riscos de um projeto. A Lista de Riscos, além da magnitude e da descrição dos riscos, traz também uma descrição do impacto de cada risco no projeto, indicadores que serão utilizados para monitorar os riscos, estratégias de mitigação e planos de contingência.

A Lista de Riscos é um artefato essencial para um processo iterativo, devendo estar presente em todos os projetos. Em projetos complexos, a Lista de Riscos pode ser apenas uma parte do Plano de Gerenciamento de Riscos. Apesar de ser recomendado que todo projeto possua uma Lista de Riscos, o número de riscos que serão tratados pode variar de acordo com as necessidades de cada projeto.

### **6.4.9 Ordem de Trabalho**

O objetivo da Ordem de Trabalho é fornecer ao Gerente uma forma de comunicar à equipe do projeto o que deve ser feito e em que momento. A Ordem de Trabalho é composta de uma lista de atividades a serem realizadas, com seus respectivos responsáveis, tempo e recursos alocados, resultados esperados, data de realização e uma indicação de concordância por parte do responsável por cada tarefa.

A Ordem de Trabalho é um artefato útil para equipes grandes, distribuídas e inexperientes no processo de desenvolvimento, já que é uma forma de assegurar um

melhor entendimento sobre as tarefas que devem ser realizadas, contribuindo, portanto, para uma melhor comunicação entre o Gerente e os membros da equipe. Em equipes pequenas, por outro lado, a Ordem de Trabalho pode gerar um *overhead* desnecessário, sendo recomendável que seja realizada de maneira informal, através de breves reuniões, correio eletrônico ou conversas individuais com os membros da equipe.

### 6.4.10 Plano de Aceitação do Produto

O Plano de Aceitação do Produto determina os critérios para que um produto seja aceito, as atividades necessárias para a aceitação do produto, os responsáveis por essas atividades, ferramentas e técnicas utilizadas nessas atividades, artefatos que serão avaliados e estratégias para a resolução de problemas encontrados durante as atividades de aceitação do produto. As informações contidas no Plano de Aceitação do Produto podem estar contidas no contrato de desenvolvimento do software, no caso de desenvolvimento por contrato, tornando o artefato desnecessário.

O Plano de Aceitação do Produto é importante para o desenvolvimento de produtos críticos, já que é fundamental estabelecer de forma precisa alguns critérios de aceitação do produto. Exemplos desses critérios são: tempo entre falhas, taxa de cobertura dos testes, realização de testes específicos, tempo de utilização de protótipo. Projetos de alto investimento também possuem, normalmente, um Plano de Aceitação do Produto, para garantir que o investimento realizado gere o produto esperado.

### 6.4.11 Plano de Garantia da Qualidade

O Plano de Garantia da Qualidade descreve a estratégia para atingir os objetivos de qualidade do projeto. O Plano de Garantia da Qualidade apresenta:

- Referência aos objetivos de qualidade do projeto (Especificação de Requisitos, da disciplina Requisitos);
- As atividades de garantia da qualidade que serão realizadas e os responsáveis por cada atividade;
- Padrões e diretrizes de qualidade a serem seguidos;
- A documentação que deve ser produzida durante o projeto (Plano de Documentação, citado, mas não definido pelo RUP);
- Medições que devem ser realizadas (Plano de Medições);
- Revisões, auditorias, avaliações e testes necessários;

- Referência aos processos de resolução de problemas (Plano de Resolução de Problemas);
- Referência aos processos de gerência de configuração (Plano de Gerência de Configuração, da disciplina Gerência de Configuração);
- Referência a controles sobre fornecedores e subcontratados (Plano de Fornecimento e Subcontratação, citado, mas não definido pelo RUP);
- Referência a atividades de treinamento (Plano de Treinamento, citado, mas não definido pelo RUP);
- Referência a atividades de gerência de riscos (Plano de Gerenciamento de Riscos).

Pode-se perceber, pela descrição acima, que o Plano de Garantia da Qualidade pode referenciar vários outros artefatos, inclusive de disciplinas diferentes de Planejamento e Gerenciamento. Alguns desses artefatos que podem ser referenciados pelo Plano de Garantia da Qualidade não são definidos pelo RUP e, por esse motivo, não serão discutidos aqui. Os artefatos de outras disciplinas também não serão tratados, por estarem fora do escopo deste trabalho. Os artefatos definidos na disciplina de Planejamento e Gerenciamento serão tratados de forma individual, assim como foi feito para os artefatos que podem estar contidos no Plano de Desenvolvimento de Software. Assim, o conteúdo do Plano de Garantia da Qualidade que será analisado ficará restrito às atividades de garantia da qualidade e seus respectivos responsáveis, os padrões e diretrizes de qualidade a serem seguidos e as revisões, auditorias, avaliações e testes que devem ser realizados.

O Plano de Garantia da Qualidade é fundamental para o desenvolvimento de softwares críticos, já que a qualidade, nesse tipo de software, deve ser fator prioritário. Em projetos mais simples, com poucas restrições de qualidade, o Plano de Garantia da Qualidade pode ficar resumido a uma lista de objetivos de qualidade a serem alcançados e uma lista de revisões, auditorias, avaliações e testes, incluindo os responsáveis por essas atividades. No caso de um Plano de Garantia da Qualidade simples, pode-se resumir-lo a uma seção do Plano de Desenvolvimento de Software.

### **6.4.12 Plano de Medições**

O Plano de Medições tem como propósito definir os objetivos do programa de medições do projeto e as medições que precisam ser feitas para atingir esses objetivos. As medições definidas no Plano de Medições podem fornecer informações sobre

cumprimento do planejamento (cronograma, custo etc.), qualidade do produto e qualidade e oportunidades de melhoria do processo. A quantidade necessária de medições depende, portanto, da importância e complexidade desses fatores em cada projeto.

Um dos principais objetivos do Plano de Medições, segundo o RUP, é fornecer informações para a Avaliação de Status. Assim, em projetos onde exista a necessidade de se produzir Avaliações de Status, é recomendável a adoção de um Plano de Medições, mesmo que de forma simplificada. Projetos de softwares críticos também se beneficiam de um Plano de Medições, já que é necessário um acompanhamento minucioso das características de qualidade do produto, o que pode ser feito monitorando as medições de qualidade. Caso as medições a serem realizadas sejam de baixa complexidade, o Plano de Medições pode ser, simplesmente, uma seção do Plano de Desenvolvimento de Software.

### **6.4.13 Registro de Revisão**

O Registro de Revisão é um formulário utilizado para capturar os resultados de revisões realizadas durante o projeto. Contém o tipo de revisão realizada, os artefatos revisados e os objetivos da revisão, os participantes da revisão, cronograma da revisão, uma lista de problemas identificados e recomendações para solucioná-los, ações a serem tomadas com base nos resultados das revisões, itens a serem considerados pelo Gerente (caso o problema não possa ser resolvido pelos participantes da revisão), revisões futuras e o esforço despendido na revisão. É um artefato utilizado em várias disciplinas do Processo, não apenas em Planejamento e Gerenciamento. A adaptação sugerida aqui, entretanto, aplica-se apenas à utilização do Registro de Revisão como artefato de Planejamento e Gerenciamento.

Projetos de desenvolvimento de softwares críticos devem possuir Registros de Revisão para cada revisão realizada, já que os problemas encontrados e ações a serem tomadas para resolvê-los devem estar claras e bem controladas. Registros de Revisão também podem servir como forma de comunicação entre os membros da equipe de desenvolvimento, sendo, por isso, útil para equipes grandes, geograficamente distribuídas e inexperientes no processo de desenvolvimento. Os Registros de Revisão podem ser úteis, por exemplo, para que o Gerente tenha mais facilidade e segurança no acompanhamento do projeto sem precisar, necessariamente, participar de todas as revisões. Registros de Revisão também podem facilitar a comunicação com o cliente, já que permite que este tenha um melhor acompanhamento do projeto através do resultado das revisões. Assim, Registros de

Revisão também são úteis para projetos de alto investimento onde, em geral, o cliente tem um maior interesse no progresso do projeto.

O fato de ser um artefato recomendável para as situações descritas acima não implica que o Registro de Revisão não possa e não deva ser utilizado em outras situações. Na prática, o Registro de Revisão deve ser utilizado sempre que uma revisão tiver grande importância para o projeto.

#### **6.4.14 Lista de Pendências**

A Lista de Pendências é um artefato que possibilita, ao Gerente do Projeto, registrar e acompanhar problemas, anomalias, exceções e tarefas incompletas que não sejam tratadas pela Gerência de Configuração nem sejam tarefas contidas nos Planos de Iteração e Plano do Projeto. Segundo o RUP, a Lista de Pendências é um artefato de formato livre, dependendo apenas das necessidades do projeto.

A Lista de Pendências é especialmente útil como uma forma de melhorar a comunicação de equipes grandes ou geograficamente distribuídas. Nesses casos, é aconselhável que a Lista de Pendências seja um artefato mais formal, indicando, por exemplo, datas e responsáveis pela resolução dos problemas e execução das tarefas, a solução a ser adotada e a precedência entre os itens pendentes. Para equipes pequenas e centralizadas, uma mera descrição dos itens pendentes é suficiente.

#### **6.4.15 Medições do Projeto**

Esse artefato é um repositório de medições do projeto. Pode agrupar medições realizadas em várias atividades. Não é um artefato que pertence exclusivamente à disciplina Planejamento e Gerenciamento. Por funcionar como um aglutinador de informações de vários artefatos diferentes, a adaptação do artefato Medições do Projeto é uma consequência direta da adaptação dos demais artefatos do projeto, principalmente do Plano de Medições. Por esse motivo, não serão fornecidas aqui diretrizes para a adaptação deste artefato.

### **6.5 Relação entre Características e Artefatos**

Nesta seção, será feita uma correspondência entre as características do projeto descritas nas seções 5.1.1 e 5.1.2 e os artefatos descritos na seção anterior, identificando quando o artefato deve ser produzido, produzido com restrições e não produzido. Esse

relacionamento foi obtido, basicamente, através da análise das informações, fornecidas pelo próprio RUP, sobre cada artefato. Por isso, é importante frisar que as matrizes apresentadas a seguir são apenas um ponto de partida, podendo sofrer modificações conforme o Modelo de Adaptação de Processos de Software seja utilizado e melhorado.

As relações entre os artefatos e as características serão identificadas através da seguinte legenda:

S – será produzido

N – não será produzido

R – será produzido com restrições (informalmente, apenas parte do artefato ou apenas se houver necessidade durante o projeto)

I – indiferente. A característica não influi nesse artefato

A Tabela 6-3 apresenta o relacionamento entre os artefatos de Planejamento e Gerenciamento e o tamanho da equipe. Da mesma forma, a Tabela 6-4 refere-se à experiência da equipe, a Tabela 6-5 refere-se à distribuição geográfica da equipe, a Tabela 6-6 trata da criticidade do software e a Tabela 6-7 trata do tamanho do projeto.

<b>Tamanho da Equipe</b>					
<b>Artefatos</b>	<b>1-6 pessoas</b>	<b>7-20 pessoas</b>	<b>21-50 pessoas</b>	<b>51-100 pessoas</b>	<b>+100 pessoas</b>
PDS - Plano de Desenvolvimento de Software	S	S	S	S	S
EV - Estudo de Viabilidade	I	I	I	I	I
PI - Plano de Iteração	S	S	S	S	S
AI - Avaliação da Iteração	R	S	S	S	S
AS - Avaliação de Status	N	N	R	R	S
PRP - Plano de Resolução de Problemas	N	N	R	S	S
PGR - Plano de Gerenciamento de Riscos	N	R	R	S	S
PR - Lista de Riscos	S	S	S	S	S
OT - Ordem de Trabalho	N	N	N	S	S
PAP - Plano de Aceitação do Produto	I	I	I	I	I
PM - Plano de Medições	N	N	R	R	S
PGQ - Plano de Garantia da Qualidade	N	N	R	R	S
RR - Registro de Revisão	N	N	S	S	S
LP - Lista de Pendências	S	S	S	S	S

**Tabela 6-3:** Artefatos *versus* tamanho da equipe



<b>Experiência da Equipe (no processo)</b>					
<b>Artefatos</b>	<b>Nenhum projeto</b>	<b>1 projeto</b>	<b>2 a 3 projetos</b>	<b>4 a 5 projetos</b>	<b>+5 projetos</b>
PDS - Plano de Desenvolvimento de Software	S	S	S	S	S
EV - Estudo de Viabilidade	I	I	I	I	I
PI - Plano de Iteração	S	S	S	S	S
AI - Avaliação da Iteração	S	S	S	R	R
AS - Avaliação de Status	S	S	R	N	N
PRP - Plano de Resolução de Problemas	S	R	R	N	N
PGR - Plano de Gerenciamento de Riscos	S	R	R	N	N
PR - Lista de Riscos	S	S	S	S	S
OT - Ordem de Trabalho	S	S	N	N	N
PAP - Plano de Aceitação do Produto	I	I	I	I	I
PM - Plano de Medições	S	S	R	N	N
PGQ - Plano de Garantia da Qualidade	S	S	R	N	N
RR - Registro de Revisão	S	S	N	N	N
LP - Lista de Pendências	I	I	I	I	I

**Tabela 6-4:** Artefatos *versus* experiência da equipe

<b>Distribuição Geográfica da Equipe</b>					
<b>Artefatos</b>	<b>Mesma sala</b>	<b>Mesmo prédio</b>	<b>Mesma cidade, mesma empresa</b>	<b>Mesma cidade, empresas diferentes</b>	<b>Cidades diferentes</b>
PDS - Plano de Desenvolvimento de Software	S	S	S	S	S
EV - Estudo de Viabilidade	I	I	I	I	I
PI - Plano de Iteração	S	S	S	S	S
AI - Avaliação da Iteração	R	S	S	S	S
AS - Avaliação de Status	N	N	R	S	S
PRP - Plano de Resolução de Problemas	N	N	N	R	S
PGR - Plano de Gerenciamento de Riscos	N	N	N	R	S
PR - Lista de Riscos	S	S	S	S	S
OT - Ordem de Trabalho	N	N	S	S	S
PAP - Plano de Aceitação do Produto	I	I	I	I	I
PM - Plano de Medições	N	N	R	S	S
PGQ - Plano de Garantia da Qualidade	N	N	R	S	S
RR - Registro de Revisão	N	N	S	S	S
LP - Lista de Pendências	S	S	S	S	S

**Tabela 6-5:** Artefatos *versus* distribuição geográfica da equipe

<b>Criticidade do Software</b>					
<b>Artefatos</b>	<b>Perda de conforto</b>	<b>Prejuízos baixos, perdas facilmente recuperáveis</b>	<b>Prejuízos moderados, perdas recuperáveis</b>	<b>Prejuízos altos, perdas irre recuperáveis</b>	<b>Risco de vida</b>
PDS - Plano de Desenvolvimento de Software	S	S	S	S	S
EV - Estudo de Viabilidade	I	I	I	I	I
PI - Plano de Iteração	S	S	S	S	S
AI - Avaliação da Iteração	R	S	S	S	S
AS - Avaliação de Status	I	I	I	I	I
PRP - Plano de Resolução de Problemas	N	N	R	S	S
PGR - Plano de Gerenciamento de Riscos	N	R	S	S	S
PR - Lista de Riscos	S	S	S	S	S
OT - Ordem de Trabalho	I	I	I	I	I
PAP - Plano de Aceitação do Produto	N	R	R	S	S
PM - Plano de Medições	N	R	S	S	S
PGQ - Plano de Garantia da Qualidade	N	R	S	S	S
RR - Registro de Revisão	N	N	N	S	S
LP - Lista de Pendências	I	I	I	I	I

Tabela 6-6: Artefatos *versus* criticidade do software

<b>Tamanho do Projeto</b>					
<b>Artefatos</b>	<b>Até R\$ 50.000,00</b>	<b>Entre R\$ 50.000,00 e R\$ 150.000,00</b>	<b>Entre R\$ 150.000,00 e R\$ 1.000.000,00</b>	<b>Entre R\$ 1.000.000,00 e R\$ 3.000.000,00</b>	<b>Acima de R\$ 3.000.000,00</b>
PDS - Plano de Desenvolvimento de Software	S	S	S	S	S
EV - Estudo de Viabilidade	R	R	R	S	S
PI - Plano de Iteração	S	S	S	S	S
AI - Avaliação da Iteração	R	R	S	S	S
AS - Avaliação de Status	N	R	S	S	S
PRP - Plano de Resolução de Problemas	N	R	R	S	S
PGR - Plano de Gerenciamento de Riscos	N	R	S	S	S
PR - Lista de Riscos	S	S	S	S	S
OT - Ordem de Trabalho	I	I	I	I	I
PAP - Plano de Aceitação do Produto	N	R	S	S	S
PM - Plano de Medições	N	R	R	S	S
PGQ - Plano de Garantia da Qualidade	N	R	R	S	S
RR - Registro de Revisão	N	N	S	S	S
LP - Lista de Pendências	I	I	I	I	I

**Tabela 6-7:** Artefatos *versus* tamanho do projeto

## 6.6 Diretrizes para a Implementação do PConfig

A definição de PConfig para outras disciplinas deve ser uma tarefa bem mais simples que a definição para a disciplina Planejamento e Gerenciamento feita nesse trabalho, uma vez que a estrutura e funcionamento de PConfig já estão definidos. É importante, porém, a participação de especialistas de cada disciplina nessa definição, já que a definição de que características impactam uma determinada disciplina e como essas características se relacionam com os artefatos da disciplina não é, de forma alguma, uma tarefa trivial para profissionais pouco familiarizados com a disciplina em questão.

Uma dificuldade adicional para a implementação do PConfig é o relacionamento entre os artefatos do processo. Esse relacionamento pode impedir que determinado artefato seja excluído do processo enquanto outro artefato fizer parte do mesmo. Por exemplo, o artefato Modelo de Dados, da disciplina Análise e Projeto, não pode ser produzido sem as Classes de Projeto. Assim, não faria sentido um processo que incluísse o artefato Modelo de Dados e excluísse o artefato Classe de Projeto. A disciplina de Planejamento e Gerenciamento, nesse sentido, é menos complexa que disciplinas mais técnicas, como Análise e Projeto e Implementação, já que existe uma maior independência entre os artefatos. Essa independência pode ser vista mais claramente na Tabela 6-8, que mostra o nível de dependência entre os artefatos. Uma dependência entre artefatos indica que um artefato é entrada para uma atividade que tem o outro artefato como saída. A dependência será fraca se o artefato de saída puder ser produzido mesmo que o artefato de entrada não faça parte do processo. A dependência será forte se o artefato de saída só puder ser produzido se o artefato de entrada existir.

É importante perceber, também, que as disciplinas do processo podem não ser totalmente independentes umas das outras. Tomando como exemplo o RUP, alguns artefatos, apesar de pertencerem a apenas uma disciplina, estão relacionados com atividades de disciplinas diferentes. Portanto, é importante que, ao se integrar duas ou mais disciplinas diferentes em PConfig, esses artefatos recebam atenção especial, evitando qualquer incompatibilidade entre as disciplinas. Um exemplo de medida que pode ser tomada para prevenir esse tipo de problema pode ser visto na Tabela 6-9 e na Tabela 6-10, que indicam, respectivamente, os artefatos de outras disciplinas que são entradas para atividades de Planejamento e Gerenciamento e artefatos de Planejamento e Gerenciamento que são entradas para atividades de outras disciplinas.

Artefatos Depende de	PDS	EV	PI	AI	AS	PRP	PGR	LR	OT	PAP	PM	PGQ	LP
PDS	-	-	forte	forte	forte	forte	-	forte	forte	-	fraca	fraca	forte
EV	fraca	-	fraca	fraca	-	-	-	-	-	fraca	fraca	fraca	-
PI	-	-	-	forte	forte	-	-	forte	forte	-	-	-	forte
AI	fraca	-	-	-	fraca	-	-	-	-	-	-	-	fraca
AS	-	-	-	fraca	-	-	-	-	fraca	-	-	-	fraca
PRP	fraca	-	-	-	-	-	-	-	fraca	-	-	fraca	fraca
PGR	fraca	-	-	-	-	-	-	fraca	-	-	-	fraca	fraca
LR	forte	forte	forte	-	fraca	-	forte	-	-	-	fraca	fraca	fraca
OT	-	-	-	-	-	-	-	-	-	-	-	-	-
PAP	fraca	-	-	-	-	-	-	-	-	-	-	-	-
PM	fraca	-	-	fraca	-	-	-	fraca	-	-	-	fraca	fraca
PGQ	fraca	-	-	-	-	-	-	-	-	-	-	-	-
LP	fraca	-	-	fraca	forte	-	-	fraca	fraca	-	-	-	-

PDS – Plano de Desenvolvimento de Software

EV – Estudo de Viabilidade

PI – Plano de Iteração

AI – Avaliação da Iteração

AS – Avaliação de Status

PRP – Plano de Resolução de Problemas

PGR – Plano de Gerenciamento de Riscos

LR – Lista de Riscos

OT – Ordem de Trabalho

PAP – Plano de Aceitação do Produto

PM – Plano de Medições

PGQ – Plano de Garantia da Qualidade

RR – Registro de Revisão

LP – Lista de Pendências

**Tabela 6-8:** Interdependência entre artefatos de Planejamento e Gerenciamento

Artefato	Presente na Disciplina	Utilizado na Atividade
Lista de Riscos	Requisitos	Gerenciar Dependências
Plano de Iteração	Requisitos	Revisar Requisitos
Lista de Riscos	Análise e Projeto	Revisar a Arquitetura
Lista de Riscos	Análise e Projeto	Avaliar Viabilidade da Prova de Conceito Arquitetural (como saída)
Estudo de Viabilidade	Análise e Projeto	Avaliar Viabilidade da Prova de Conceito Arquitetural (como saída)
Lista de Pendências	Testes	Obter Compromisso de Testabilidade (como saída)
Lista de Pendências	Testes	Avaliar e Defender Qualidade (como saída)
Lista de Pendências	Testes	Avaliar e Melhorar Aplicação de Testes (como saída)
Plano de Iteração	Implementação	Planejar Integração do Sistema
Plano de Iteração	Implementação	Planejar Integração do Subsistema
Plano de Desenv. de Software	Testes	Acordar Objetivos
Plano de Iteração	Testes	Identificar Motivadores dos Testes
Plano de Iteração	Testes	Identificar Objetivos dos Testes
Plano de Iteração	Implantação	Desenvolver Plano de Implantação + Definir Lista de Entrega
Plano de Desenv. de Software	Implantação	Desenvolver Plano de Implantação + Definir Lista de Entrega
Plano de Aceitação do Produto	Implantação	Desenvolver Plano de Implantação + Definir Lista de Entrega
Plano de Aceitação do Produto	Implantação	Gerenciar Testes de Aceitação
Ordem de Trabalho	Gerência de Configuração	Fazer Mudanças
Ordem de Trabalho	Gerência de Configuração	Criar Linhas de Base
Ordem de Trabalho	Gerência de Configuração	Entregar Mudanças (como saída)

**Tabela 6-9:** Artefatos de Planejamento e Gerenciamento que impactam outras disciplinas

Artefato	Da Disciplina	Utilizado na Atividade
Visão	Requisitos	Identificar e Avaliar Riscos
Visão	Requisitos	Revisão de Aprovação do Projeto
Visão	Requisitos	Desenvolver Estudo de Viabilidade
Visão	Requisitos	Definir Organização e Alocação de Pessoal do Projeto
Visão	Requisitos	Desenvolver Plano de Iteração
Visão	Requisitos	Avaliar Iteração
Requisição de Mudanças	Gerência de Configuração	Alocar e Atribuir Tarefa
Requisição de Mudanças	Gerência de Configuração	Gerenciar Exceções e Problemas (como saída)
Requisição de Mudança	Gerência de Configuração	Avaliar Iteração (como saída)
Documento de Arquitetura de Software	Análise e Projeto	Desenvolver Plano de Iteração
Atributos de Requisitos	Requisitos	Desenvolver Plano de Iteração
Plano de Testes	Testes	Avaliar Iteração
Plano de Testes	Testes	Revisar Critério de Avaliação da Iteração
Resumo dos Resultados de Teste	Testes	Avaliar Iteração
Caso de Desenvolvimento	Ambiente	Avaliar Iteração
Avaliação da Organização Desenvolvedora	Ambiente	Avaliar Iteração

**Tabela 6-10:** Artefatos de outras disciplinas que impactam Planejamento e Gerenciamento

Através de uma análise da Tabela 6-8, da Tabela 6-9 e da Tabela 6-10, pode-se perceber algumas características da disciplina Planejamento e Gerenciamento:

- Os artefatos Plano de Desenvolvimento de Software, Plano de Iteração e Lista de Riscos são essenciais ao processo de Planejamento e Gerenciamento, já que grande parte dos outros artefatos dependem fortemente deles. Isso indica que esses três artefatos devem estar presentes na grande maioria dos processos adaptados.
- O artefato Lista de Pendências é um artefato importante para o acompanhamento do projeto. Até mesmo a Avaliação de Status, que é um artefato que informa de maneira mais completa a situação do projeto em um dado instante, é dependente da Lista de Pendências. Essa dependência está relacionada à estrutura das versões anteriores do RUP, onde a Lista de Pendência era uma seção da Avaliação de Status que se destacava por ser constantemente atualizada. Na versão 2002 do RUP, a Avaliação de Status e a Lista de Pendências foram transformadas em dois artefatos distintos, porém a Avaliação de Status continua precisando das informações da Lista de Pendências.
- O artefato Visão, da disciplina Requisitos, é de grande importância para a disciplina Planejamento e Gerenciamento. De fato, como se trata da primeira descrição do software a ser produzido, a Visão é utilizada como subsídio para o desenvolvimento de vários artefatos de Planejamento e Gerenciamento como, por exemplo, o Estudo de Viabilidade, a Lista de Riscos e o Plano de Desenvolvimento de Software.
- A transmissão de informações de Planejamento e Gerenciamento para outras disciplinas é feita, principalmente, pelos artefatos Plano de Iteração, Ordem de Trabalho e Lista de Pendências, o que é natural, já que a função desses artefatos é comunicar o que precisa ser feito em cada momento do projeto.
- O artefato Registro de Revisão não foi incluído na Tabela 6-8. Isso deve-se às características do artefato, que tem como objetivo formalizar os resultados de uma revisão para facilitar a utilização desses resultados. Assim, esse artefato pode existir sempre que houver uma revisão, não dependendo especificamente da existência de outro artefato. Da mesma forma, ainda que



uma determinada revisão não gere um Registro de Revisão, os resultados da revisão continuarão existindo e poderão ser utilizados por artefatos que necessitem desses resultados, ou seja, mesmo utilizando os resultados de revisões, os artefatos não têm sua existência dependente da existência do Registro de Revisão. Como o Registro de Revisão não depende de outros artefatos e os outros artefatos não dependem do Registro de Revisão, optou-se por não incluir o Registro de Revisão na Tabela 6-8 como forma de simplificar a visualização das dependências entre artefatos.

Uma alternativa para evitar inconsistências no processo, tanto entre disciplinas como dentro de uma única disciplina, é a utilização de apoio automatizado na modelagem do processo. Ferramentas como o Rational Process Workbench, descrito na Seção 5.2.1, são capazes de detectar erros na modelagem do processo e podem ser de grande valia para o trabalho do Engenheiro de Processos.

## 6.7 Considerações

Neste capítulo, foi apresentado o PConfig, que é um processo para configuração de processos de software com base nos seus artefatos. Foi descrito como o PConfig, a partir de uma relação entre os artefatos do processo padrão e as características dos projetos, determina que artefatos devem fazer parte de cada projeto adaptado.

Também foi feita uma análise dos artefatos da disciplina Planejamento e Gerenciamento do RUP, identificando em que situações cada um deles é útil e em que situações cada artefato é dispensável.

Além disso, foram identificados os artefatos de outras disciplinas que impactam a disciplina Planejamento e Gerenciamento e artefatos de Planejamento e Gerenciamento que impactam outras disciplinas.



---

## Capítulo 7

# Avaliação do MAPS

---

Este Capítulo descreve o estudo de caso realizado para avaliar o Modelo de Adaptação de Processos de Software (MAPS). O Capítulo possui a seguinte estrutura:

- A Seção 7.1 apresenta os objetivos do estudo de caso, identificando os principais pontos a serem avaliados.
- A Seção 7.2 define, passo a passo, a abordagem utilizada para realizar o estudo de caso, bem as como justificativas para a escolha dessa abordagem.
- A Seção 7.3 descreve a avaliação realizada, apresentando os dados obtidos e tecendo alguns comentários relevantes sobre os mesmos.
- A Seção 7.4 apresenta uma análise geral dos resultados obtidos.

## 7.1 Objetivos

Somente a definição do Modelo de Adaptação de Processos de Software (MAPS) não é suficiente para afirmar que ele pode ser aplicado na prática, com desempenho satisfatório. Por isso, para tentar validar a consistência e a viabilidade do MAPS, foi realizada uma avaliação do uso do Modelo em situações práticas.

Os principais aspectos a serem considerados eram:

- A aplicação do MAPS em um primeiro projeto, quando a Base de Processos estivesse vazia, para avaliar o esforço de adaptação sem reuso de processos e estimar a dificuldade de calibrar o PConfig para o contexto específico de uma organização.
- A aplicação do MAPS em um segundo projeto, semelhante ao primeiro, para avaliar os ganhos do reuso de processos e o Modelo de Caracterização de Projetos.
- A identificação das vantagens da utilização do MAPS e de oportunidades de melhoria do Modelo.

## 7.2 Abordagem Utilizada

O cenário ideal para avaliar um modelo como o MAPS seria a sua aplicação em alguns projetos de desenvolvimento de software reais. Esse cenário, entretanto, mostrou-se inviável por diversas razões:

1. Para avaliar o reuso de processos e o Modelo de Caracterização de Projetos, seria necessário aplicar o MAPS a, pelo menos, dois projetos.
2. A aplicação do MAPS teria que ser feita desde o início dos projetos e só poderia ser avaliada em uma fase adiantada dos projetos, o que demandaria um tempo considerável. Além disso, os projetos não poderiam ocorrer ao mesmo tempo, sob pena de não levar em consideração a avaliação e a melhoria do processo utilizado no primeiro projeto antes do reuso do processo para o segundo projeto.
3. A dificuldade de encontrar uma organização com um processo padrão definido e que estivesse disposta a aplicar um modelo de adaptação ainda não testado. Essa dificuldade é devida, sobretudo, à pequena faixa de manobra da grande maioria dos projetos de software, o que não permite correr nenhum risco desnecessário que possa afetar o custo e o tempo de conclusão do projeto.

Por conta dessas dificuldades, optou-se por realizar um estudo alternativo que, apesar de não ser o ideal, permite uma boa avaliação de alguns aspectos do MAPS e é muito mais viável do ponto de vista da organização desenvolvedora. A abordagem aplicada é composta dos seguintes passos:

**Passo 1: Escolha dos projetos.** Foram escolhidos dois projetos de uma mesma empresa, que utilizavam processos baseados no RUP. Foram escolhidos, propositalmente, dois projetos, aqui referenciados como Projeto A e Projeto B<sup>18</sup>, com características semelhantes, para que se pudesse testar o reuso de processos. Os dois projetos escolhidos estavam em uma fase adiantada do desenvolvimento, o que era um pré-requisito essencial para que se pudesse avaliar, com segurança, o processo utilizado, já que no início do projeto não existem subsídios suficientes para analisar se o processo foi bem sucedido.

---

<sup>18</sup> Os nomes reais dos projetos serão omitidos a pedido da organização desenvolvedora.

**Passo 2: Caracterização dos projetos.** Foram realizadas reuniões iniciais com os gerentes dos projetos escolhidos, com o objetivo de conhecer o escopo e as características dos projetos, particularmente aquelas relacionadas com a disciplina Planejamento e Gerenciamento e que foram discutidas na Seção 5.1.2. Para caracterizar os projetos, foi utilizado o questionário apresentado no Apêndice A.

**Passo 3: Adaptação do processo para o Projeto A.** Com base nas características do Projeto A, foi realizada a adaptação do processo padrão através do MAPS. A adaptação foi realizada com base apenas no PConfig, já que não existiam processos armazenados na Base de Processos. Para esse passo, considerou-se o RUP como processo padrão a ser adaptado.

**Passo 4: Avaliação e compatibilização dos processos do Projeto A.** Foi realizada, em uma segunda reunião com o gerente do primeiro projeto, uma avaliação comparativa entre o processo sugerido pelo MAPS e o processo realmente utilizado no projeto. Essa avaliação teve, como principal objetivo, verificar se o processo sugerido pelo MAPS era realmente adequado ao projeto e identificar ganhos e perdas caso esse tivesse sido o processo utilizado no projeto. Essa avaliação foi feita com base nos artefatos do processo e foi considerada como a Avaliação Final do Processo sugerida na Seção 5.2.3. Para avaliar o processo utilizado no Projeto A, foi utilizado o questionário apresentado no Apêndice C. Cada artefato foi avaliado em relação à sua utilidade para o projeto e ao custo/benefício da produção do artefato.

A utilidade dos artefatos foi avaliada classificando-os em 4 grupos:

- **Muito Útil:** o artefato é essencial para o projeto, sua ausência pode causar danos irreparáveis.
- **Útil:** o artefato é importante para o projeto e sua existência aumenta a qualidade, produtividade ou controle sobre o projeto.
- **Pouco Útil:** o artefato agrega pouco valor ao projeto, sua ausência não causa grandes problemas.
- **Inútil:** o artefato não tem nenhuma importância para o projeto, sua ausência não traz nenhum prejuízo.

Os artefatos também foram classificados em grupos para avaliar o custo/benefício da produção do artefato, levando em consideração o esforço de produção de cada artefato e os benefícios gerados pelo artefato, como, por exemplo, maior controle sobre o projeto, ganhos na qualidade do produto e do processo e melhor

comunicação entre os membros da equipe de desenvolvimento:

- Ótimo: os benefícios gerados pelo artefato compensam largamente seu custo.
- Bom: os benefícios gerados pelo artefato compensam seu custo, mas melhorias podem ser feitas para tornar essa relação mais satisfatória.
- Regular: os benefícios gerados pelo artefato não compensam seu custo, porém a produção do artefato não acarreta grandes prejuízos ao projeto. É provável que o artefato não devesse ser produzido.
- Péssimo: os benefícios gerados pelo artefato são ínfimos perto do seu custo e a produção do artefato acrescenta uma sobrecarga de trabalho considerável ao projeto. O artefato não deve ser produzido.

Para manter a consistência entre a avaliação realizada e a estrutura do PConfig, era necessário utilizar o RUP como processo padrão a ser adaptado, já que a implementação do PConfig descrita no Capítulo 6 está baseada no RUP. Os processos utilizados no Projeto A e no Projeto B, entretanto, seguiam o processo padrão utilizado pela organização desenvolvedora. Esse processo, apesar de ser fortemente baseado no RUP, possui algumas especificidades. Por isso, além de avaliar os artefatos, era preciso tornar o processo padrão da organização compatível com o RUP, já que existiam diferenças de nomenclatura e estrutura entre os artefatos dos dois processos. Foi realizada uma avaliação da função dos artefatos do processo padrão da organização, que foram, então, mapeados em artefatos do RUP.

**Passo 5: Adaptação do processo para o Projeto B.** Foi aplicado, ao Projeto B, o método de comparação de projetos sugerido na Seção 5.1.3, para verificar se a suposta semelhança entre o Projeto A e o Projeto B se confirmava quando da aplicação das regras do Modelo de Caracterização de Projetos. A seguir, foi aplicada a estratégia de reuso sugerida nas Seções 5.1.4 e 5.2.2. Por último, foi aplicado o PConfig para finalizar a adaptação.

**Passo 6: Avaliação e compatibilização dos processos do Projeto B.** Foi realizado, para o Projeto B, a mesma avaliação feita para o Projeto A (Passo 4).

**Passo 7: Análise dos resultados obtidos.** Finalmente, foram analisadas as informações obtidas com as avaliações dos processos utilizados pelo Projeto A e pelo Projeto B e dos processos sugeridos pelo MAPS para os dois projetos, para determinar se a utilização do MAPS, com e sem reuso de processos, poderia trazer ganhos para os projetos avaliados.

## 7.3 Avaliação Realizada

A avaliação do MAPS foi realizada em uma organização de grande porte que, entre outras atividades, desenvolve software para várias áreas. A organização possui um processo padrão de desenvolvimento, fortemente baseado no RUP, que é adaptado para cada projeto, mas sem seguir critérios documentados e bem definidos. A organização também possui uma equipe de garantia da qualidade, que é responsável por acompanhar a utilização do processo nos vários projetos. Foram escolhidos dois projetos da organização, Projeto A e Projeto B, para a realização da avaliação do MAPS. Esses projetos tiveram os artefatos dos seus processos mapeados para artefatos do RUP para manter a consistência com a implementação do PConfig descrita no Capítulo 6.

Serão descritas, a seguir, as aplicações do MAPS realizadas no Projeto A e no Projeto B com o objetivo de avaliar o Modelo. Para cada projeto, é feita uma breve descrição, seguida de uma caracterização do projeto. São descritos os processos utilizados nos projetos e os processos sugeridos pelo MAPS. No caso do Projeto B, também é descrito o método de reuso utilizado. É descrita, também, a avaliação dos processos, tanto os utilizados quanto os sugeridos pelo MAPS, realizada em conjunto com os gerentes dos projetos.

### 7.3.1 Projeto A

O Projeto A é um projeto de desenvolvimento de um sistema para a área tributária de um órgão público. O projeto, no período em que foi avaliado, estava em sua fase final de desenvolvimento. A duração prevista do Projeto A é de 17 meses, com a implementação de 240 casos de uso. O Projeto A comporta desenvolvedores de duas organizações diferentes, mas que trabalham em um mesmo espaço físico.

#### Caracterização do Projeto

O Projeto A foi caracterizado de acordo com as características e níveis definidos na Seção 5.1.2. Como a Base de Processos, para esse projeto, estava vazia, a caracterização visa somente o reuso do processo utilizado no Projeto A em outros projetos (no caso específico deste trabalho, o reuso será feito no Projeto B).

Com base nas informações da gerência do projeto, o Projeto A ficou caracterizado da seguinte forma:

**Tamanho da Equipe:** Nível 3 (Média; 21 a 50 pessoas)

**Experiência da Equipe no Processo:** Nível 2 (um projeto)

**Distribuição Geográfica da Equipe:** Nível 1 (Mesma sala)

**Criticidade do Projeto:** Nível 3 (Prejuízos moderados, perdas recuperáveis)

**Tamanho do Projeto:** Nível 4 (Entre R\$ 1.000.000,00 e R\$ 3.000.000,00)

**Prioridade do Projeto:** Nível 3 (Qualidade = 50, Tempo = 50)

### Processo Utilizado

O processo que foi utilizado no Projeto A está resumido na Tabela 7-1. Para cada artefato, é indicado se ele foi utilizado, ou não, no projeto. Existe também a possibilidade de utilização do artefato com restrições, que são, em geral, simplificações do artefato sugerido pelo RUP. A coluna Observações traz informações sobre os artefatos relativas à cultura e ao processo padrão da organização estudada.

Artefatos	Utilização	Observações
Plano de Desenvolvimento de Software	S	É chamado Plano de Projeto.
Estudo de Viabilidade	N	O projeto já veio pronto da área de negócios, com estudo de viabilidade e documento de requisitos.
Plano de Iteração	S	
Avaliação da Iteração	S	
Avaliação de Status	S	O artefato tem outro nome, mas tem os mesmos objetivos da Avaliação de Status. É realizada mensalmente.
Plano de Resolução de Problemas	R	Não existe um plano escrito, porém existe um procedimento padronizado para reportar os erros e problemas encontrados.
Plano de Gerenciamento de Riscos	S	O projeto utiliza uma lista de riscos estendida para capturar informações que, no RUP, estão no Plano de Gerenciamento de Riscos.
Lista de Riscos	S	
Ordem de Trabalho	S	Existe uma ferramenta que produz a Ordem de Trabalho automaticamente a partir do WBS do Plano de Desenvolvimento de Software.
Plano de Aceitação do Produto	N	
Plano de Medições	N	
Plano de Garantia da Qualidade	N	
Registro de Revisão	S	Existe uma política organizacional de armazenar os resultados de todas as revisões e torná-las disponíveis em uma página web.
Lista de Pendências	S	O artefato tem outro nome, mas tem os mesmos objetivos da Lista de Pendências. É atualizado semanalmente.

LEGENDA: S – é produzido N – não é produzido R – é produzido com restrições

**Tabela 7-1:** Processo utilizado no Projeto A



## Processo MAPS

Como, para este estudo de caso, o Projeto A está sendo considerado como o primeiro projeto da organização a utilizar o MAPS, a adaptação do processo padrão, o RUP, será feita através apenas do PConfig, já que não existem processos cadastrados na Base de Processos.

A Tabela 7-2 mostra o relacionamento entre os artefatos do RUP, utilizado como processo padrão, e as características do Projeto A segundo as regras estabelecidas no PConfig. A coluna Resultados mostra o resultado final da aplicação de PConfig ao Projeto A. Comentários são inseridos sempre que necessário.

O artefato Estudo de Viabilidade não foi incluído no mapeamento porque, na organização desenvolvedora, esse artefato é de responsabilidade da área de negócios e não faz parte da disciplina Planejamento e Gerenciamento. Essa especificidade implica em uma mudança no PConfig, excluindo o Estudo de Viabilidade que, a partir de agora, para o estudo de caso, não será considerado como um artefato de Planejamento e Gerenciamento.

A Avaliação de Status deve ser realizada de forma simplificada, incluindo apenas informações sobre custos e cronograma. As informações mais técnicas devem ser incluídas apenas na Avaliação da Iteração.

O Plano de Resolução de Problemas, o Plano de Gerenciamento de Riscos, o Plano de Aceitação do Produto, o Plano de Medições e o Plano de Garantia da Qualidade devem ser apenas seções do Plano de Desenvolvimento de Software, e não documentos separados.

O Registro de Revisão só deve ser realizado para as revisões mais importantes, como, por exemplo, revisões de marcos principais.

Artefatos	Tamanho da Equipe	Experiência da Equipe	Distribuição Geográfica da Equipe	Criticidade do Software	Tamanho do Projeto	Resultado
Plano de Desenvolvimento de Software	S	S	S	S	S	S
Plano de Iteração	S	S	S	S	S	S
Avaliação da Iteração	S	S	S	S	S	S
Avaliação de Status	N	S	N	I	S	R
Plano de Resolução de Problemas	N	R	N	R	S	R
Plano de Gerenc. de Riscos	R	R	N	S	S	R
Lista de Riscos	S	S	S	S	S	S
Ordem de Trabalho	N	S	N	I	I	N
Plano de Aceitação do Produto	I	I	I	R	S	R
Plano de Medições	N	S	N	S	S	R
Plano de Garantia da Qualidade	N	S	N	S	S	R
Registro de Revisão	N	S	N	N	S	R
Lista de Pendências	S	I	S	I	I	S

LEGENDA: S – é produzido N – não é produzido R – é produzido com restrições I - indiferente

**Tabela 7-2:** Processo sugerido pelo MAPS para o Projeto A, sem características restritivas

Para chegar ao processo final sugerido pelo MAPS, porém, deve-se levar em conta as características restritivas do projeto. Essas características provocam alterações no processo adaptado.

Existe uma ferramenta que produz a Ordem de Trabalho automaticamente a partir do cronograma do Plano de Desenvolvimento de Software. Logo, o artefato Ordem de Trabalho, mesmo tendo sido identificado como dispensável, será incluído, já que sua produção não acrescenta nenhum custo ao projeto.

Existe uma política organizacional de armazenar os resultados de todas as revisões e torná-las disponíveis através de páginas HTML na intranet da organização. Assim, apesar do MAPS recomendar que os Registros de Revisão fossem utilizados apenas nas revisões mais importantes, esse artefato será produzido em todas as revisões por uma restrição imposta pela organização.

O processo, em termos de artefatos, sugerido pelo MAPS para o Projeto A é apresentado na Tabela 7-3.

Artefatos	Utilização	Observações
Plano de Desenvolvimento de Software	S	
Plano de Iteração	S	
Avaliação da Iteração	S	
Avaliação de Status	R	Apenas informações sobre cronograma e custos. Informações técnicas serão incluídas apenas nas Avaliações de Iteração
Plano de Resolução de Problemas	R	Deve ser apenas uma seção do Plano de Desenvolvimento de Software.
Plano de Gerenciamento de Riscos	R	Deve ser apenas uma seção do Plano de Desenvolvimento de Software.
Lista de Riscos	S	
Ordem de Trabalho	S	Existe uma ferramenta que produz a Ordem de Trabalho automaticamente a partir do WBS do Plano de Desenvolvimento de Software.
Plano de Aceitação do Produto	R	Deve ser apenas uma seção do Plano de Desenvolvimento de Software.
Plano de Medições	R	Deve ser apenas uma seção do Plano de Desenvolvimento de Software.
Plano de Garantia da Qualidade	R	Deve ser apenas uma seção do Plano de Desenvolvimento de Software.
Registro de Revisão	S	Existe uma política organizacional de armazenar os resultados de todas as revisões e torná-las disponíveis em uma página web.
Lista de Pendências	S	

LEGENDA: S – é produzido N – não é produzido R – é produzido com restrições I - indiferente

**Tabela 7-3:** Processo sugerido pelo MAPS para o Projeto A

## Avaliação dos Processos

Foi realizada, para o Projeto A, a avaliação sugerida na Seção 7.2 (Passo 4). Os artefatos utilizados no Projeto A foram avaliados da seguinte forma:

Artefatos	Utilidade	Custo/Benefício
PDS – Plano de Desenvolvimento de Software	Muito Útil	Ótimo
PI – Plano de Iteração	Muito Útil	Bom
AI – Avaliação da Iteração	Muito Útil	Ótimo
AS – Avaliação de Status	Pouco Útil	Péssimo
PRP – Plano de Resolução de Problemas	Inútil	Péssimo
PGR – Plano de Gerenciamento de Riscos	Muito Útil	Ótimo
LR – Lista de Riscos	Muito Útil	Ótimo
OT – Ordem de Trabalho	Útil	Ótimo
RR – Registro de Revisão	Muito Útil	Ótimo
LP – Lista de Pendências	Muito Útil	Ótimo

**Tabela 7-4:** Avaliação dos artefatos utilizados no Projeto A

Os artefatos do processo padrão que não foram utilizados no projeto receberam a seguinte avaliação:

Artefatos	Utilidade	Custo/Benefício
PAP – Plano de Aceitação do Produto	Útil	Bom
PM – Plano de Medições	Pouco Útil	Regular
PGQ – Plano de Garantia da Qualidade	Útil	Regular

**Tabela 7-5:** Avaliação dos artefatos não utilizados no Projeto A

## Análise dos Processos

Comparando o processo sugerido pelo MAPS (Tabela 7-3) e o processo realmente utilizado no Projeto A (Tabela 7-1), e analisando a avaliação dos artefatos feita pelo gerente do projeto (Tabela 7-4 e Tabela 7-5), pode-se concluir que:

- No processo utilizado no Projeto A, em relação aos artefatos definidos no RUP, existem 9 artefatos produzidos de forma completa, 1 artefato produzido com restrições e 3 artefatos, fora o Estudo de Viabilidade, não produzidos. Já no processo sugerido pelo MAPS, 7 artefatos devem ser produzidos de forma completa, 6 artefatos devem ser produzidos com restrições e nenhum artefato, fora o Estudo de Viabilidade, deve deixar de ser produzido. Esses números mostram um certo equilíbrio entre os dois

processos, sendo difícil apontar qual dos dois é o processo mais leve.

- A Avaliação de Status foi vista como pouco útil e teve seu custo/benefício avaliado como péssimo. Segundo informações do gerente do projeto, isso se deve, principalmente, ao alto nível de detalhamento desse artefato. Assim, além de um grande esforço para produzi-lo, há uma certa resistência das partes envolvidas em acompanhar o projeto através desse artefato por julgá-lo muito complexo. A abordagem utilizada pelo MAPS, nesse caso, mostrou-se mais apropriada, já que diminui o número de informações do artefato, tornando-o mais sucinto e, portanto, menos custoso e com maior probabilidade de ser utilizado pelas partes envolvidas.
- O Plano de Resolução de Problemas é realizado informalmente e a existência de um artefato formal foi avaliada como desnecessária, além de ter um custo/benefício péssimo. O MAPS sugeriu que o Plano de Resolução de Problemas fosse uma seção do Plano de Desenvolvimento de Software. O mais provável, analisando a avaliação feita pela gerência do projeto, é que até mesmo essa seção no Plano de Desenvolvimento de Software seria desnecessária ao projeto. Essas informações, a sugestão inicial de produzir o Plano de Resolução de Problemas como uma seção do Plano de Desenvolvimento de Software e a avaliação negativa do artefato devem ser armazenadas na Base de Processos.
- O processo sugerido pelo MAPS propôs que as informações do Plano de Gerenciamento de Riscos fossem resumidas em uma seção do Plano de Desenvolvimento de Software. O processo utilizado no projeto utilizou um artefato mais completo, produzido em conjunto com a Lista de Riscos. Como o artefato foi avaliado como muito útil e com um ótimo custo/benefício, pode-se concluir que, nesse caso, o processo utilizado é mais adequado que o processo sugerido pelo MAPS. Essa necessidade de contar com um artefato mais completo deve ser incluída na avaliação do artefato que será armazenada na Base de Processos.
- O Plano de Aceitação do Produto deveria ser produzido, de acordo com o MAPS, como uma seção do Plano de Desenvolvimento de Software. O processo utilizado no projeto não previa a produção do Plano de Aceitação do Produto. Como o artefato foi avaliado como útil e com um ótimo

custo/benefício, conclui-se que, nesse caso, o processo sugerido pelo MAPS seria o mais adequado.

- De acordo com o MAPS, o Plano de Medições deveria ser, para o Projeto A, apenas uma seção do Plano de Desenvolvimento de Software. O processo utilizado no projeto não previa a produção do Plano de Medições. Como o artefato foi avaliado como pouco útil e com um custo/benefício apenas regular, conclui-se que, nesse caso, o processo utilizado seria o mais adequado. Tanto a sugestão inicial de produzir o Plano de Medições como uma seção do Plano de Desenvolvimento de Software quanto a avaliação negativa do artefato devem ser armazenadas na Base de Processos.
- O Plano de Garantia da Qualidade não estava previsto no processo utilizado no projeto. No processo sugerido pelo MAPS, o Plano de Garantia da Qualidade aparece como uma seção do Plano de Desenvolvimento de Software. O artefato foi avaliado, pelo gerente do projeto, como muito útil, porém com um custo/benefício apenas regular, o que indica uma preocupação com o custo de produção do artefato. A sugestão do MAPS, nesse caso, está perfeitamente de acordo com a avaliação da gerência, ou seja, produzir o artefato, mas de forma simplificada para diminuir o custo associado à sua produção.
- No processo utilizado no projeto, a Ordem de Trabalho foi produzida automaticamente, através de uma ferramenta. No uso real do MAPS, essa prática poderia ser capturada pelas avaliações do processo (Apêndice C) e incorporada ao processo padrão.

### 7.3.2 Projeto B

O Projeto B é um projeto de desenvolvimento de um sistema para a área financeira de uma grande empresa de varejo. O projeto estava, no período em que foi avaliado, em um estágio avançado do desenvolvimento, com dois dos seus três módulos implantados e um em fase de implantação. A duração prevista do projeto é de 21 meses e terão sido implementados, ao final do projeto, 239 casos de uso.

O Projeto B está sendo desenvolvido, em parceria, por duas empresas diferentes. Entretanto, como essas empresas trabalham em módulos independentes, o Projeto B não chega a ser um exemplo de desenvolvimento com equipe geograficamente distribuída.

## Caracterização do Projeto

A caracterização do Projeto B teve como principal objetivo compará-lo com o Projeto A para verificar a possibilidade de reuso do processo do Projeto A no Projeto B segundo as regras estabelecidas no MAPS.

Com base nas informações da gerência do projeto, o Projeto B ficou caracterizado da seguinte forma:

**Tamanho da Equipe:** Nível 3 (Pequena; 7 a 20 pessoas)

**Experiência da Equipe no Processo:** Nível 2 (um projeto)

**Distribuição Geográfica da Equipe:** Nível 1 (mesma sala)

**Criticidade do Projeto:** Nível 3 (prejuízos moderados, perdas recuperáveis)

**Tamanho do Projeto:** Nível 4 (entre R\$ 1.000.000,00 e R\$ 3.000.000,00)

**Prioridade do Projeto:** Nível 3 (Qualidade = 50, Tempo = 50)

## Processo Utilizado

O processo que foi utilizado no Projeto B está resumido na Tabela 7-6. Assim como foi feito para o Projeto A, é indicado, para cada artefato, se ele foi utilizado, utilizado com restrições ou não utilizado no projeto. Informações sobre os artefatos relativas à cultura e ao processo padrão da organização estudada são apresentadas na coluna Observações.

Artefatos	Utilização	Observações
Plano de Desenvolvimento de Software	S	É chamado Plano de Projeto.
Plano de Iteração	S	
Avaliação da Iteração	S	
Avaliação de Status	S	O artefato tem outro nome, mas tem os mesmos objetivos da Avaliação de Status. É realizada trimestralmente.
Plano de Resolução de Problemas	N	
Plano de Gerenciamento de Riscos	S	
Lista de Riscos	S	
Ordem de Trabalho	N	
Plano de Aceitação do Produto	R	É uma seção do Plano de Desenvolvimento de Software.
Plano de Medições	N	
Plano de Garantia da Qualidade	N	
Registro de Revisão	S	
Lista de Pendências	S	O artefato tem outro nome, mas tem os mesmos objetivos da Lista de Pendências. É atualizado semanalmente.

LEGENDA: S – é produzido N – não é produzido R – é produzido com restrições

**Tabela 7-6:** Processo utilizado no Projeto B

## Processo MAPS

Caso o processo para o Projeto B fosse definido apenas através do PConfig, sem reusar o processo utilizado no Projeto A, o processo sugerido pelo MAPS seria o apresentado na Tabela 7-9.

Porém, ao aplicarmos o método de comparação definido no Modelo de Caracterização de Projetos, podemos observar a grande semelhança entre o Projeto B e o Projeto A. Aplicando a fórmula definida na Seção 5.1.3, temos:

$$S_{PG} = 1 - \frac{|n_1 - n_1'| + |n_2 - n_2'| + |n_3 - n_3'| + |n_4 - n_4'| + |n_5 - n_5'|}{5}$$

onde  $n_1, n_2, n_3, n_4$  e  $n_5$  são, respectivamente, as características tamanho da equipe, experiência da equipe, distribuição geográfica da equipe, criticidade do software e tamanho do projeto do Projeto A. Da mesma forma,  $n_1', n_2', n_3', n_4'$  e  $n_5'$  são, respectivamente, as características tamanho da equipe, experiência da equipe, distribuição geográfica da equipe, criticidade do software e tamanho do projeto do Projeto B. Substituindo os valores, temos:

$$S_{PG} = 1 - \frac{|3 - 2| + |2 - 2| + |1 - 1| + |3 - 3| + |4 - 4|}{5}$$

$$S_{PG} = 1 - \frac{1}{5} = 1 - 0,2 = 0,8$$

Assim, ainda segundo o método de comparação definido na Seção 5.1.3, o reuso do processo do Projeto A no Projeto B cai no caso 2, já que o valor de  $S_{PG}$  está entre 0,5 e 1,0, o que implica que o Projeto A e o Projeto B são muito semelhantes em relação à disciplina Planejamento e Gerenciamento. A abordagem recomendada para esse caso, na Seção 5.1.4, é reusar o processo, adaptando-o de acordo com as diferenças apontadas pela comparação dos projetos.

Como a única característica que possui valores diferentes nos projetos é o tamanho da equipe, essa será a única característica que será analisada. Para realizar essa análise, devemos recuperar, de PConfig, as informações sobre o relacionamento entre os artefatos do processo padrão e o tamanho da equipe, níveis 2 (Projeto B) e 3 (Projeto A), conforme apresentado na Tabela 7-7.



<b>Tamanho da Equipe</b>		
<b>Artefatos</b>	<b>Nível 2 (7-20 pessoas)</b>	<b>Nível 3 (21-50 pessoas)</b>
PDS - Plano de Desenvolvimento de Software	S	S
PI – Plano de Iteração	S	S
AI - Avaliação da Iteração	S	S
AS - Avaliação de Status	N	R
PRP - Plano de Resolução de Problemas	N	R
PGR - Plano de Gerenciamento de Riscos	R	R
PR - Lista de Riscos	S	S
OT - Ordem de Trabalho	N	N
PAP - Plano de Aceitação do Produto	I	I
PM - Plano de Medições	N	R
PGQ - Plano de Garantia da Qualidade	N	R
RR - Registro de Revisão	N	S
LP – Lista de Pendências	S	S

LEGENDA: S – é produzido N – não é produzido R – é produzido com restrições I - indiferente

**Tabela 7-7:** Comparação do Projeto A e do Projeto B quanto ao tamanho da equipe

Do nível 3 (Projeto A) para o nível 2 (Projeto B), notamos que a Avaliação de Status, o Plano de Resolução de Problemas, o Plano de Medições, o Plano de Garantia da Qualidade e o Registro de Revisão perdem importância, podendo deixar de ser produzidos. A decisão de modificar, ou não, a utilização desses artefatos depende da avaliação feita no projeto anterior e das características restritivas do projeto.

A Tabela 7-8 apresenta o processo final sugerido pelo MAPS para o Projeto B.

Artefatos	Utilização	Observações
Plano de Desenvolvimento de Software	S	Reusado do Projeto A.
Plano de Iteração	S	Reusado do Projeto A.
Avaliação da Iteração	S	Reusado do Projeto A.
Avaliação de Status	R	Apenas informações sobre cronograma e custos. Informações técnicas serão incluídas apenas nas Avaliações de Iteração. Reusado do Projeto A.
Plano de Resolução de Problemas	N	Foi mal avaliado no Projeto A, por isso foi retirado do processo.
Plano de Gerenciamento de Riscos	S	A necessidade desse artefato foi identificada através da avaliação feita no Projeto A.
Lista de Riscos	S	Reusado do Projeto A.
Ordem de Trabalho	S	Informalmente. No Projeto A, este artefato foi incluído como artefato formal porque existia uma ferramenta (característica restritiva) que eliminava o esforço necessário para a produção do artefato. Como o Projeto B utiliza a mesma ferramenta, o artefato será incluído no processo utilizando a lição aprendida do Projeto A.
Plano de Aceitação do Produto	R	Deve ser apenas uma seção do Plano de Desenvolvimento de Software. Reusado do Projeto A.
Plano de Medições	N	Foi mal avaliado no Projeto A, por isso foi retirado do processo.
Plano de Garantia da Qualidade	R	Deve ser apenas uma seção do Plano de Desenvolvimento de Software. Reusado do Projeto A.
Registro de Revisão	S	Existe uma política organizacional de armazenar os resultados de todas as revisões e torná-las disponíveis em uma página web. Característica restritiva que se aplica ao Projeto A e ao Projeto B.
Lista de Pendências	S	Reusado do Projeto A.

LEGENDA: S – é produzido N – não é produzido R – é produzido com restrições

**Tabela 7-8:** Processo sugerido pelo MAPS para o Projeto B

Caso não houvesse reuso de processos do Projeto A para o Projeto B, o processo gerado pelo MAPS para o Projeto B seria baseado apenas no PConfig. Esse processo é apresentado na Tabela 7-9 para permitir a comparação com o processo gerado utilizando reuso.

<b>Artefatos</b>	<b>Tamanho da Equipe</b>	<b>Experiência da Equipe</b>	<b>Distribuição Geográfica da Equipe</b>	<b>Criticidade do Software</b>	<b>Tamanho do Projeto</b>	<b>Resultado</b>
Plano de Desenv. de Software	S	S	S	S	S	S
Plano de Iteração	S	S	S	S	S	S
Avaliação da Iteração	S	S	S	S	S	S
Avaliação de Status	N	S	N	I	S	R
Plano de Resolução de Problemas	N	R	N	R	S	R
Plano de Gerenc. de Riscos	R	R	N	S	S	R
Lista de Riscos	S	S	S	S	S	S
Ordem de Trabalho	N	S	N	I	I	R
Plano de Aceitação do Produto	I	I	I	R	S	R
Plano de Medições	N	S	N	S	S	R
Plano de Garantia da Qualidade	N	S	N	S	S	R
Registro de Revisão	N	S	N	N	S	R
Lista de Pendências	S	I	S	I	I	S

LEGENDA: S – é produzido N – não é produzido R – é produzido com restrições I - indiferente

**Tabela 7-9:** Processo sugerido pelo MAPS para o Projeto B, sem reuso e sem características restritivas

## Avaliação dos Processos

Assim como para o Projeto A, foi realizada, para o Projeto B, a avaliação sugerida na Seção 7.2 (passo 4). Os artefatos utilizados no Projeto B foram avaliados da seguinte forma:

Artefatos	Utilidade	Custo/Benefício
PDS – Plano de Desenvolvimento de Software	Muito Útil	Ótimo
PI – Plano de Iteração	Muito Útil	Bom
AI – Avaliação da Iteração	Muito Útil	Ótimo
AS – Avaliação de Status	Útil	Bom
PRP – Plano de Resolução de Problemas	Inútil	Péssimo
PGR – Plano de Gerenciamento de Riscos	Muito Útil	Ótimo
PAP – Plano de Aceitação do Produto	Útil	Ótimo
LR – Lista de Riscos	Muito Útil	Ótimo
OT – Ordem de Trabalho	Útil	Ótimo
RR – Registro de Revisão	Muito Útil	Ótimo
LP – Lista de Pendências	Muito Útil	Ótimo

**Tabela 7-10:** Avaliação dos artefatos utilizados no Projeto B

Os artefatos do processo padrão que não foram utilizados no Projeto B receberam a seguinte avaliação:

Artefatos	Utilidade	Custo/Benefício
PM – Plano de Medições	Pouco Útil	Regular
PGQ – Plano de Garantia da Qualidade	Muito Útil	Regular

**Tabela 7-11:** Avaliação dos artefatos não utilizados no Projeto B

## Análise dos Processos

Para o Projeto B, além de comparar o processo sugerido pelo MAPS com o processo utilizado no projeto, é necessário também comparar os processos gerados pelo MAPS com e sem o reuso do processo do Projeto A para avaliar o impacto do reuso de processos na adaptação. Os principais pontos observados a partir dessas comparações foram:

- No processo utilizado no Projeto B, existem 8 artefatos produzidos de forma completa, 1 artefato produzido com restrições e 4 artefatos não produzidos. No processo sugerido pelo PConfig, sem levar em conta o reuso de

processos, 5 artefatos devem ser produzidos de forma completa, 8 artefatos devem ser produzidos com restrições e nenhum artefato deve deixar de ser produzido. Já no processo sugerido pelo MAPS, utilizando o reuso de processos, 7 artefatos devem ser produzidos de forma completa, 4 artefatos devem ser produzidos com restrições e 2 artefatos não devem ser produzidos. Por esses números, pode-se perceber que, apesar de diferentes quanto à utilização dos artefatos, os três processos são relativamente semelhantes em relação ao peso, sendo difícil determinar quando um é mais leve que o outro.

- O processo sugerido pelo MAPS é bastante semelhante ao processo utilizado no Projeto B. Como a maioria dos artefatos foi reusada do Projeto A, o ganho, em termos de diminuição de esforço, caso o MAPS tivesse sido utilizado nos dois projetos, seria considerável.
- No processo utilizado no Projeto B, não existia um Plano de Resolução de Problemas. Esse artefato foi avaliado, pelo gerente do projeto, como desnecessário e com um custo/benefício péssimo. O MAPS propôs que não fosse utilizado um Plano de Resolução de Problemas, o que está de acordo com a avaliação do gerente. Isso só foi possível devido à avaliação do artefato feita pelo gerente do Projeto A, cujo processo foi reusado. Caso a adaptação fosse feita apenas através do PConfig, o Plano de Resolução de Problemas teria sido incluído como uma seção do Plano de Desenvolvimento de Software, o que acrescentaria um custo desnecessário ao projeto. Essa mesma situação pode ser vista quanto ao Plano de Medições, que foi retirado do processo gerado pelo MAPS para o Projeto B devido à avaliação negativa do artefato feita no Projeto A.
- O Plano de Gerenciamento de Riscos estava presente no processo utilizado no projeto e foi avaliado, pelo gerente, como muito útil e com um custo/benefício ótimo. O processo sugerido pelo MAPS também indica a necessidade de produzir esse artefato. Essa necessidade, entretanto, só pode ser identificada através da avaliação do artefato Plano de Gerenciamento de Riscos feita no Projeto A. Caso a adaptação tivesse sido feita apenas através do PConfig, esse artefato teria sido incluído apenas como uma seção do Plano de Desenvolvimento de Software.

- Caso o MAPS tivesse sido utilizado no Projeto A e no Projeto B, o artefato Ordem de Trabalho seria incluído no processo sugerido pelo MAPS para o Projeto B. No Projeto A, a Ordem de Trabalho era produzida automaticamente, através do uso de uma ferramenta. Essa técnica seria capturada na avaliação final do projeto (Apêndice C), através das perguntas “Que melhorias no processo utilizado você recomendaria para um projeto com características semelhantes?” e “Que métodos, técnicas ou ferramentas utilizados no projeto poderiam ser incluídos no processo padrão da organização?”. Essa técnica poderia, então, ser incorporada ao processo padrão e ser utilizada nos demais projetos da organização.
- O Plano de Aceitação do Produto, no processo utilizado no projeto, foi produzido como uma seção do Plano de Desenvolvimento de Software. Essa também foi a sugestão apresentada pelo MAPS, tanto utilizando o reuso quanto utilizando apenas o PConfig. Nesse caso, os benefícios do reuso de processos seriam o menor esforço necessário e a maior confiabilidade dessa decisão, já que existia uma avaliação positiva anterior feita em um projeto com características semelhantes.
- O Plano de Garantia da Qualidade não foi incluído no processo utilizado no Projeto B. No processo sugerido pelo MAPS, é indicada a necessidade de produzir esse artefato como uma seção do Plano de Desenvolvimento de Software, posição reforçada pela avaliação positiva do artefato feita no Projeto A. A abordagem sugerida pelo MAPS está em concordância com a avaliação do artefato feita pelo gerente do Projeto B, que avaliou o artefato como necessário, mas com um custo/benefício apenas regular, o que demonstra a preocupação com o custo de produção do artefato e reforça a idéia de produzi-lo apenas como uma seção do Plano de Desenvolvimento de Software.
- O Registro de Revisão foi adotado para todas as revisões devido a uma restrição organizacional que já havia sido tratada no Projeto A e que se aplica também ao Projeto B.

## 7.4 Análise dos Resultados

Apesar das limitações já citadas, a realização do estudo de caso trouxe uma valiosa contribuição para o trabalho, evidenciando alguns aspectos da aplicação prática do MAPS.

A adaptação do processo padrão (RUP) para o Projeto A mostrou que o PConfig está relativamente bem ajustado, mas que é preciso adaptá-lo às condições específicas da organização desenvolvedora. A diferença de nomenclatura dos artefatos, por exemplo, mostram que mesmo para um processo baseado no RUP, o PConfig ainda pode precisar de ajustes. Essa necessidade ficou mais evidente no caso do artefato Estudo de Viabilidade, que, ao contrário do que acontece no RUP, não faz parte da disciplina Planejamento e Gerenciamento do processo da organização estudada, já que sua produção é de responsabilidade da área de negócios da empresa, e não da gerência do projeto.

Um ponto positivo evidenciado pelo estudo de caso foi o grande potencial do reuso de processos, tanto do ponto de vista da diminuição do esforço de adaptação quanto do ponto de vista da melhoria dos processos adaptados. A adaptação para o Projeto B mostrou-se muito mais fácil e eficiente do que a adaptação para o Projeto A. Isso se deveu, principalmente, à avaliação, no contexto específico do projeto, da utilidade e do custo/benefício dos artefatos do processo padrão utilizados e não utilizados no projeto, o que permite corrigir facilmente erros cometidos em adaptações anteriores.

O estudo de caso também salientou um aspecto importante do MAPS que precisa ser melhorado, que é a necessidade de uma ferramenta que automatize a comparação de projetos e facilite a escolha do processo a ser reusado. Apesar da relativa facilidade em realizar essas atividades de forma manual no estudo de caso, ficou claro que, com um maior número de disciplinas, características, artefatos a serem considerados e com o previsível aumento das informações contidas na Base de Processos, ficará inviável realizar as adaptações de forma eficiente sem a ajuda de uma ferramenta de apoio.

A forma de avaliação da prioridade do projeto (qualidade *versus* tempo) foi outro ponto de melhoria evidenciado pelo estudo de caso. Ambos os projetos, através de suas gerências, indicaram que a qualidade do produto e o tempo de conclusão do projeto têm a mesma importância. Por isso, a prioridade do projeto não foi utilizada como fator

de decisão para a inclusão, ou não, de um artefato. Essa análise da gerência, entretanto, tanto pode ser real como pode ser tendenciosa, já que a gerência do projeto é responsável por atingir tanto as metas de qualidade quanto as de tempo de conclusão. Assim, a gerência do projeto pode não se sentir confortável para indicar que um desses aspectos tem menor importância. Esse é, portanto, um aspecto que merece ser avaliado de uma forma mais objetiva, tornando a avaliação menos dependente da opinião do gerente do projeto.



---

## Capítulo 8

# Conclusões

---

Este capítulo apresenta algumas conclusões sobre o trabalho realizado. A estrutura do Capítulo é a seguinte:

- Na Seção 8.1 são feitas algumas considerações sobre o trabalho e são apontadas suas principais contribuições.
- A Seção 8.2 descreve algumas dificuldades encontradas durante o desenvolvimento do MAPS, salientando o impacto que essas dificuldades tiveram no trabalho.
- Na Seção 8.3 é feita uma análise comparativa do presente trabalho com outros trabalhos semelhantes.
- A Seção 8.4 apresenta trabalhos futuros que podem agregar valor ao MAPS.

### 8.1 Considerações Gerais e Principais Contribuições

Neste trabalho, foi proposto um modelo (MAPS) que tem como objetivo auxiliar a adaptação de um processo padrão de desenvolvimento de software para projetos específicos. O MAPS tem uma estrutura semelhante à estrutura para processos de software sugerida pelo CMM, sendo compatível com o nível 3 do CMM, que trata da definição e adaptação de processos.

O trabalho realizado na elaboração do MAPS foi bastante abrangente. Essa abrangência, apesar de ter impedido que um modelo mais completo fosse produzido, foi importante para identificar futuras melhorias e extensões do MAPS, descritas como trabalhos futuros na Seção 8.4, e definir uma arquitetura capaz de abrigar essas modificações.

As principais contribuições do trabalho são:

- Um modelo (MAPS) para adaptação de processos de software a partir de um processo padrão. O MAPS contempla, ainda, a avaliação, melhoria e reuso

dos processos adaptados. O MAPS tem, como objetivo final, o estabelecimento de uma “família” de processos adaptados, todos derivados do processo padrão, em que cada “membro” da “família” é um processo adaptado, testado e aprovado para uma circunstância específica.

- Um Modelo de Caracterização de Projetos que permite comparar projetos de software a partir de suas características.
- Uma análise das principais características dos projetos de desenvolvimento de software que influenciam o planejamento e o gerenciamento desses projetos.
- Um relacionamento entre os artefatos da disciplina Planejamento e Gerenciamento do RUP e as características que influenciam a disciplina, identificando, caso a caso, a necessidade de produzir, ou não, cada artefato.

## 8.2 Dificuldades Encontradas

Serão apresentadas, a seguir, as principais dificuldades encontradas durante a realização deste trabalho. Serão descritas as soluções adotadas e o impacto dessas soluções no trabalho.

### 8.2.1 Complexidade do Processo de Software

A idéia inicial do trabalho era desenvolver um modelo de adaptação de processos de software que contemplasse o processo de software como um todo, abrangendo todas as suas disciplinas. Essa idéia, entretanto, mostrou-se inviável por uma série de motivos, entre os quais podemos destacar:

- O grande número de características que impactam o processo de software, o que, além de tornar mais difícil a caracterização de projetos, diminui bastante a possibilidade de reuso de processos.
- A complexidade das disciplinas do processo, especialmente considerando o processo padrão como um processo concreto e abrangente, com um grande número de elementos, funcionando como uma base de conhecimentos que deve ser adaptada para cada projeto.
- A falta de conhecimentos específicos sobre cada disciplina, dificultando a escolha das características que impactam cada disciplina e a definição de que artefatos deveriam ser produzidos em que circunstâncias.

A solução encontrada para atenuar essas dificuldades foi dividir o processo em disciplinas e realizar todos os passos da adaptação com base nessa divisão. Essa estratégia facilita o reuso de processos, já que é mais fácil encontrar projetos com características semelhantes em relação a apenas uma disciplina do que em relação a todo o processo. Além disso, essa divisão diminui a complexidade do processo, já que é possível analisar uma disciplina de cada vez, o que também atenua o problema da necessidade de conhecimentos específicos de cada disciplina, já que cada disciplina pode ser incorporada ao MAPS a partir do trabalho de um especialista na disciplina.

### **8.2.2 Estudo de Caso**

Como já foi dito no 0, não foi possível realizar o estudo de caso previsto inicialmente para validar o MAPS, ou seja, aplicar processos adaptados gerados pelo MAPS em projetos reais. Além do grande tempo que seria consumido nessa atividade, outro fator que inviabilizou essa abordagem foi a dificuldade em encontrar uma organização que possuísse um processo padrão e que estivesse disposta a avaliar o MAPS em alguns projetos. Parte dessa dificuldade deriva, obviamente, da impossibilidade de realizar testes em projetos reais com a garantia de não causar nenhuma espécie de prejuízo ao projeto. Mesmo com a real possibilidade de futuros ganhos para a organização, a realidade da maioria das empresas nacionais de desenvolvimento de software não permite bancar esse investimento. Uma boa alternativa para realizar a validação das disciplinas que serão futuramente adicionadas ao MAPS é tentar incluir a avaliação do Modelo no escopo de um projeto para alcançar o nível 3 do CMM. Como para alcançar o nível 3 do CMM é necessário definir critérios para a adaptação do processo padrão da organização, é provável que houvesse maior interesse da organização em adotar e desenvolver o MAPS para cumprir essa exigência do CMM. Infelizmente, ainda são poucas as empresas que estão buscando atingir esse nível de maturidade de processos.

O problema do estudo de caso foi contornado utilizando uma avaliação comparativa entre o processo sugerido pelo MAPS e o processo que foi realmente utilizado no projeto. Essa abordagem, apesar de fornecer informações importantes para o desenvolvimento do Modelo, tem o defeito de tornar mais subjetiva a avaliação do processo adaptado através do MAPS. A avaliação fica, em parte, baseada na opinião da

equipe de projeto sobre o processo que seria utilizado e não nos resultados reais do projeto.

A inexistência das informações históricas que o MAPS precisa sobre a utilização de processos em projetos também foi outro fator que impediu a realização de um estudo de caso mais completo, principalmente em relação às atividades de comparação e reuso de processos, em um espaço de tempo curto.

### 8.3 Trabalhos Relacionados

Existem inúmeros trabalhos que tratam sobre adaptação de processos de software, cada um com suas particularidades e contribuições. A seguir, serão brevemente descritos alguns desses trabalhos, ressaltando a diferença em relação a este trabalho.

Budlong e Szulewski [12] propõem um método de adaptação baseado na escolha de “blocos de construção”, que são conjuntos de atividades interrelacionadas. Entretanto, não é descrito claramente nenhum mecanismo para reusar os processos e não é feita uma associação dos blocos de construção com as características do projeto, embora seja dito que a escolha dos blocos que serão utilizados depende dessas características.

Cameron [45] propõe uma adaptação de processos baseada na descrição de artefatos do processo e das circunstâncias sob as quais cada artefato deve ser produzido. A adaptação é feita escolhendo os artefatos que serão utilizados e definindo em que seqüência eles serão produzidos. Essa abordagem é bastante semelhante à utilizada no PConfig, mas Cameron limita-se a descrever o método de adaptação, não descrevendo as características e circunstâncias que interferem na escolha dos artefatos. Também não existe nenhuma proposta explícita de reuso de processos.

Em [63], Berger propõe a construção de uma ferramenta para apoiar a adaptação de processos, utilizando, para isso, conceitos de gestão do conhecimento. Como se trata de um trabalho ainda em andamento, não foi possível fazer uma comparação mais aprofundada com o MAPS.

Machado [1] define um modelo para definição e adaptação de processos de software com base na norma ISO 12207 [64] e descreve uma ferramenta, DEF-PRO, para auxiliar essas atividades. A norma ISO 12207, e não um processo padrão, é utilizada como ponto de partida para a adaptação. São descritos alguns critérios de

adaptação, mas, como não existe um processo padrão pré-estabelecido, não é feita correspondência entre esses critérios e os artefatos do processo. Além dessa diferença, não é descrito nenhum mecanismo para reuso de processos.

Borges e Falbo [48] apresentam uma ferramenta, chamada ProKnowHow, para apoiar a adaptação de processos e a coleta e disseminação do conhecimento adquirido, além da melhoria do processo padrão. As informações coletadas para a base de conhecimento são artefatos, planos, código, ferramentas, técnicas, idéias, fatos, questões, discussões etc. Não é feito reuso de processos ou disciplinas de processos e, como consequência, não existe um método de comparação de projetos para reuso de processos. Os critérios de adaptação considerados são aqueles definidos por Machado [1].

Henninger [27] propõe uma abordagem baseada em uma ferramenta para identificar “em que contexto uma prática ou processo específico é aplicável”. A ferramenta, chamada BORE (*Building an Organizational Repository of Experiences*), captura as mudanças feitas no processo padrão e as circunstâncias sob as quais essas mudanças foram necessárias, utilizando essas informações para auxiliar a adaptação de processos para outros projetos. Provavelmente, o trabalho de Henninger é o que possui mais semelhanças com o MAPS. Entretanto, Henninger não detalha que características impactam o processo nem a forma de recuperar as informações do BORE para realizar a adaptação de processos para projetos com características semelhantes a projetos anteriores. Também não é detalhada a forma de avaliar as mudanças feitas no processo padrão para determinar se elas foram, ou não, realmente efetivas.

Barros et al. [65] propõem uma estratégia para armazenar e reusar o conhecimento sobre gerenciamento de projetos ao longo dos vários projetos da organização utilizando cenários. Essa tarefa, de certa forma, também é realizada pelo MAPS, se considerarmos que o conjunto de características de um projeto constitui um cenário. O trabalho de Barros, entretanto, não trata da adaptação de processos e não se limita a representar os processos utilizados, mas sim toda espécie de conhecimento do gerente sobre o gerenciamento de projetos. Apesar da grande diferença de escopo dos dois trabalhos, o trabalho de Barros é citado aqui por existirem algumas possíveis oportunidades de integração entre a abordagem por ele sugerida e o MAPS.

## 8.4 Trabalhos Futuros

A definição de um modelo para a adaptação de processos de software mostrou-se uma tarefa bastante complexa. Apesar da estrutura e do funcionamento básico do Modelo terem sido definidos e da disciplina Planejamento e Gerenciamento ter sido detalhada, o MAPS não está concluído. Algumas tarefas importantes não puderam ser realizadas e são candidatas naturais a trabalhos futuros.

### 8.4.1 Detalhamento de Outras Disciplinas do Processo

Conforme foi exposto durante o trabalho, o MAPS, atualmente, contempla apenas a disciplina Planejamento e Gerenciamento. Entretanto, como a estrutura e o funcionamento do MAPS já estão definidos, espera-se que o esforço necessário para incluir novas disciplinas no Modelo seja consideravelmente menor que o esforço para incluir a disciplina Planejamento e Gerenciamento.

O trabalho necessário para a inclusão de uma nova disciplina pode ser resumido nas seguintes atividades:

- Determinar as características de projeto que impactam a disciplina.
- Relacionar os artefatos da disciplina com as características de projeto escolhidas para determinar em que circunstâncias cada artefato deve ser utilizado.
- Determinar a dependência entre a disciplina e as demais disciplinas do processo, ou seja, que artefatos da disciplina incluída impactam outras disciplinas e que artefatos de outras disciplinas impactam a disciplina incluída.

### 8.4.2 Desenvolvimento do PConfig para Outros Processos

O MAPS, atualmente, precisa que o processo padrão seja baseado no RUP. Para que o MAPS possa ser utilizado com outros tipos de processo padrão, é necessário adaptar o PConfig, mapeando as características do projeto nos artefatos existentes no novo processo padrão. Caso o processo não seja dividido em disciplinas, essa divisão passa a constituir um esforço adicional antes da utilização do MAPS.

### 8.4.3 Automação da Comparação de Projetos

Existem várias ferramentas para modelagem, armazenamento e publicação de processos de software, conforme já foi discutido na Seção 4.4. Porém, o MAPS não se resume a essas atividades. Seria importante o desenvolvimento de uma ferramenta capaz de implementar a Base de Processos e automatizar a atividade de comparação de projetos conforme as regras estabelecidas no Modelo de Caracterização de Projetos. Além disso, essa ferramenta deveria dar suporte ao processo de reuso de processos, conforme definido na Seção 5.2.2, e à avaliação dos processos adaptados descrita na Seção 5.2.3.

### 8.4.4 Avaliação do MAPS

O estudo de caso realizado foi bastante útil, já que permitiu uma melhor avaliação dos potenciais benefícios do MAPS. Entretanto, é necessário um maior número de aplicações práticas do MAPS, utilizando o processo sugerido pelo MAPS como processo real do projeto, para que seja possível identificar problemas e possibilidades de melhoria não identificados no estudo de caso e avaliar melhor os benefícios do MAPS em médio prazo, principalmente em relação ao reuso e à melhoria de processos, que são duas das principais contribuições do trabalho.

### 8.4.5 Integração com Outros Trabalhos

O MAPS possui um mecanismo para melhorar os processos adaptados. Esse mecanismo está baseado na avaliação dos processos utilizados para que, ao ser reusado, esse processo incorpore as melhorias sugeridas na avaliação. Assim, ao longo dos projetos, serão construídos processos cada vez melhores para cada situação.

Uma forma de acelerar a obtenção de processos ótimos para cada situação seria incorporar ao MAPS, como processo adaptado, processos desenvolvidos para situações específicas em outros trabalhos. Um exemplo disso seria a utilização de *plug-ins* e *roadmaps* do RUP [16], que contêm informações sobre como utilizar o RUP em situações específicas como, por exemplo, em pequenos projetos e em projetos de sistemas de tempo real, além de informações sobre como incorporar técnicas das metodologias ágeis ao RUP.

Uma outra vantagem de integrar o MAPS com trabalhos mais específicos seria uma melhor definição sobre como simplificar os artefatos e atividades dos processos, já que o MAPS, devido à sua abrangência, não se preocupa em avaliar detalhes da realização de atividades e produção de artefatos.

Um complicador para a integração de processos definidos em outros trabalhos com o MAPS é a necessidade de garantir a consistência entre os processos incorporados e o processo padrão da organização. Os processos sugeridos pelo MAPS são naturalmente compatíveis com o processo padrão, já que foram gerados a partir dele. Isso não é verdade para processos externos que sejam incorporados ao MAPS. Como esses processos foram desenvolvidos fora do contexto da organização, é necessário estabelecer uma forma de garantir a compatibilidade com o processo padrão, seja através de modificações feitas no processo incorporado ou através de modificações feitas no próprio processo padrão.

#### **8.4.6 Comparação de Projetos e Reuso de Processos Baseados em Artefatos**

É possível adaptar o MAPS para que ele faça a comparação de projetos tendo, como unidade de comparação, os artefatos do processo ao invés das disciplinas. Assim, seriam estabelecidas as características que influenciam a produção de cada artefato. A comparação de projetos definiria que projetos possuem características que indiquem necessidades semelhantes de produção dos artefatos.

Essa abordagem permitiria um reuso ainda maior dos processos mas, por outro lado, dificultaria a tarefa de adaptação e manutenção do MAPS. Porém, seria interessante, no futuro, implementar o MAPS dessa forma para fazer um estudo comparativo entre as duas abordagens.

### **8.5 Considerações Finais**

O Modelo de Adaptação de Processos de Software (MAPS) mostrou, através do estudo de caso realizado, que tem potencial para ser utilizado na prática. A utilização do MAPS pode trazer benefícios imediatos para a organização em termos de utilização de um processo mais adequado para as especificidades de cada projeto. Esses benefícios tendem a se acentuar com a contínua utilização do MAPS, já que processos cada vez



melhores devem ser obtidos com um esforço cada vez menor, por conta das atividades de avaliação, melhoria e reuso de processos definidas pelo MAPS.

Além disso, o desenvolvimento do MAPS permitiu um melhor entendimento da área de processos de software e provocou uma reflexão sobre os benefícios e as dificuldades da adaptação de processos. Desse melhor entendimento e dessa reflexão surgiu a necessidade de melhorar o MAPS, apontando para a realização de trabalhos futuros que tornarão o Modelo mais completo e aplicável, aproximando-o cada vez mais de um produto de uso prático para a indústria de software.



---

## Referências

---

- [1] MACHADO, L.F.C., *Modelo para Definição de Processos de Software na Estação TABA*, COPPE/UFRJ, Dissertação de Mestrado, 2000.
- [2] MINISTÉRIO DA CIÊNCIA E TECNOLOGIA / SECRETARIA DE POLÍTICA DE INFORMÁTICA (MCT / SEPIN), *Qualidade e Produtividade no Setor de Software Brasileiro*, 1999.
- [3] HUMPHREY, W. S., *Managing the Software Process*, Addison-Wesley, 1990.
- [4] KRUCHTEN, P., *Agility with the RUP*, The Rational Edge, January, 2002.
- [5] KRUCHTEN, P., *The Rational Unified Process*, Addison-Wesley, 1998.
- [6] RATIONAL UNIFIED PROCESS WEB SITE, [www.rational.com/rup](http://www.rational.com/rup), último acesso em fevereiro de 2003.
- [7] OPEN WEB SITE, [www.open.org.au](http://www.open.org.au), último acesso em fevereiro de 2003.
- [8] PERSONAL SOFTWARE PROCESS WEB SITE, [www.sei.cmu.edu/psp](http://www.sei.cmu.edu/psp), último acesso em fevereiro de 2003.
- [9] TEAM SOFTWARE PROCESS WEB SITE, [www.sei.cmu.edu/tsp](http://www.sei.cmu.edu/tsp), último acesso em fevereiro de 2003.
- [10] EXTREME PROGRAMMING WEB SITE, [www.extremeprogramming.org](http://www.extremeprogramming.org), último acesso em fevereiro de 2003.
- [11] COCKBURN, A., *Selecting a Project's Methodology*, IEEE Software, July/August 2000.
- [12] BUDLONG, F. C., SZULEWSKI, P. A., *Tailoring Your Software Process for Software Project Plans – Part 1: Setting the Context and Making Tailoring Decisions*, Crosstalk, STSC, Hill Air Force Base, 1996.
- [13] EBERT, C., DE MAN, J., *A Method and tool for Managing Process Diversity in an Industrial Setting*, net.objectdays, 2000, disponível em [www.netobjectdays.org/pdf/00/papers/ooss/ebert.pdf](http://www.netobjectdays.org/pdf/00/papers/ooss/ebert.pdf), último acesso em dezembro de 2002.

- 
- [14] HALEY, T., et al., *Raytheon Electronic Systems Experience in Software Process Improvement*, Software Engineering Institute, relatório técnico, 1995.
- [15] PAULK, M.C., WEBER, C.V., CURTIS, B., CHRISISS, M.B., *The Capability Maturity Model: Guidelines for Improving the Software Process*, Addison-Wesley, 1997.
- [16] RATIONAL SOFTWARE CORPORATION, *Rational Unified Process 2002*, 2002.
- [17] SPAWAR SYSTEMS CENTER, *A Description Of The Space And Naval Warfare Systems Center*, relatório técnico, 2001.
- [18] BECK, K., et al. *Manifesto for Agile Software Development*, disponível em <http://agilealliance.org>, último acesso em dezembro de 2002.
- [19] FOWLER, M., HIGHSMITH, J., *The Agile Manifesto*, Software Development Magazine, August, 2001.
- [20] FOWLER, M., *The New Methodology*, disponível em <http://www.martinfowler.com/articles/newMethodology.html>, último acesso em dezembro de 2002.
- [21] HIGHSMITH, J., *Agile Software Development Ecosystems*, Addison-Wesley, 2002.
- [22] CHARETTE, R., *The Decision Is In: Agile Versus Heavy Methodologies*, Cutter Consortium Executive Update, 2 (19), 2002.
- [23] AMBLER, S., *Duking It Out*, Software Development Magazine, July, 2002.
- [24] BOEHM, B., *Get Ready for Agile Methods, with Care*, IEEE Computer, 35(1), 2002.
- [25] TURK, D., FRANCE, R., RUMPE, B., *Limitations of Agile Software Processes*, XP2002: Extreme Programming Conference, 2002.
- [26] BOOCH, G., MARTIN, R., NEWKIRK, J., *dx: The Process*, capítulo de livro ainda não publicado, disponível em <http://www.objectmentor.com/publications/RUPvsXP.pdf>, último acesso em janeiro de 2003.
- [27] HENNINGER, S., et al., *Supporting Adaptable Methodologies to Meet Evolving Project Needs*, XP and Agile Universe Conference, Chicago, USA, 2002.

- 
- [28] LINDVALL, M., RUS, I., *Process Diversity in Software Development*, IEEE Software, 17(4), 2000.
- [29] ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS, SUBCOMITÊ DE SOFTWARE, COMISSÃO DE ESTUDOS EM GERÊNCIA DO CICLO DE VIDA DO SOFTWARE, [www.pr.gov.br/abntsoftware/ce10013](http://www.pr.gov.br/abntsoftware/ce10013), último acesso em fevereiro de 2003.
- [30] FIORINI, S. T., VON STAA, A., BAPTISTA, R. M., *Engenharia de Software com CMM*, Brasport, 1998.
- [31] GINSBERG, M.P., QUINN, L.H., *Process Tailoring and the Software Capability Maturity Model*, CMU/SEI – Relatório Técnico, 1995.
- [32] FOWLER, M., *Variations on a Theme of XP*, MartinFowler.com, 2001, disponível em [www.martinfowler.com/articles/xpVariation.html](http://www.martinfowler.com/articles/xpVariation.html), último acesso em dezembro de 2002.
- [33] ASD WEB SITE, [www.adaptivesd.com](http://www.adaptivesd.com), último acesso em Abril de 2002.
- [34] HENDERSON-SELLERS, B., *Process Flexibility and Process Construction*, Cobol Report, 2002, disponível em [www.cobolreport/columnists/brian/01212002.asp](http://www.cobolreport/columnists/brian/01212002.asp), último acesso em dezembro de 2002.
- [35] EXTREME PROGRAMMING DISCUSSION GROUP, <http://groups.yahoo.com/group/extremeprograming/message/48777>, último acesso em dezembro de 2002.
- [36] EXTREME PROGRAMMING DISCUSSION GROUP, <http://groups.yahoo.com/group/extremeprograming/message/45860>, último acesso em dezembro de 2002.
- [37] CONTROL CHAOS WEB SITE, [www.controlchaos.com/xpScrum.htm](http://www.controlchaos.com/xpScrum.htm), último acesso em fevereiro de 2003.
- [38] XBREED WEB SITE, [www.xbreed.net](http://www.xbreed.net), último acesso em fevereiro de 2003.
- [39] CRYSTAL METHODOLOGIES WEB SITE, [www.crystallmethodologies.org](http://www.crystallmethodologies.org), último acesso em fevereiro de 2003.

- 
- [40] SCHULTZ, D. et al., *A Matrix Approach to Software Process Definitions*, 25th Annual Software Engineering Workshop, Goddard Space Flight Center, 2000.
- [41] MENESES, J. B., MOURA, H. P., *Inspector: Um Processo de Avaliação de Progresso para Projetos de Software*, Centro de Informática, Universidade Federal de Pernambuco, Dissertação de Mestrado, 2001.
- [42] MENESES, J. B., MOURA, H. P., *Inspector: Um Processo de Avaliação de Progresso para Projetos de Software*, XV Simpósio Brasileiro de Engenharia de Software (SBES 2001), 2001.
- [43] BOOCH, G., RUMBAUGH, J., JACOBSON, I., *The unified Modeling Language User Guide*, Addison Wesley, 1998.
- [44] PMI STANDARDS COMMITTEE, *A Guide to the Project Management Body of Knowledge*, Project Management Institute, 1996.
- [45] CAMERON, J., *Configurable Development Processes*, Communications of the ACM, 45 (3), 2002.
- [46] CMMI WEB SITE, [www.sei.cmu.edu/cmmi](http://www.sei.cmu.edu/cmmi), último acesso em fevereiro de 2003.
- [47] CMMI PRODUCT TEAM, *CMMI for Systems Engineering / Software Engineering / Integrated Product and Process Development / Supplier Sourcing, Version 1.1, Continuous Representation*, Software Engineering Institute, 2002.
- [48] BORGES, L. M. S., FALBO, R. A., *Uma Ferramenta para Instanciação de Processos de Software e Apoio ao Compartilhamento de Experiências*, I Simpósio Brasileiro de Qualidade de Software (SBQS 2002), 2002.
- [49] HENNINGER, S., *Using Software Process to Support Learning Software Organizations*, 1st Workshop on Learning Organizations, Kaiserlautern, Alemanha, 1999.
- [50] JONES, C., *Positive and Negative Factors that Influence Software Productivity*, Software Productivity Research, 1998.
- [51] JONES, C., *Project Management Tools and Software Failures and Successes*, Software Productivity Research, CrossTalk, Julho 1998.
- [52] THE STANDISH GROUP INTERNATIONAL, *CHAOS: A Recipe for Success*, The Standish Group International, Inc., 1999.

- 
- [53] ROYCE, W., *Software Project Management: A Unified Framework*, Addison-Wesley, 1998.
- [54] SILVA JÚNIOR, C. R., *Reestruturação e Expansão do Methodology Explorer*, Centro de Informática, Universidade Federal de Pernambuco, Trabalho de Graduação, 2003.
- [55] RATIONAL SOFTWARE CORPORATION, *Rational Process Workbench*, 2002.
- [56] HERBSLEB, J.D., et al., *An Empirical Study of Global Software Development: Distance and Speed*, International Conference on Software Engineering (ICSE 2201), 2001.
- [57] BOEHM, B., et al., *Software Cost Estimation with COCOMO II*, Prentice Hall, 2000.
- [58] AMBLER, S., *Introduction to Agile Modeling*, Ronin International, 2002.
- [59] JEFFRIES, R., ANDERSON, A., HENDRICKSON, C., *Extreme Programming Installed*, Addison-Wesley, 2001.
- [60] ISO ONLINE, [www.iso.ch](http://www.iso.ch), último acesso em fevereiro de 2003.
- [61] YOURDON, E., *Death March: managing "mission impossible" projects*, Prentice Hall, 1997.
- [62] LEFFINGWELL, D., *Agile Requirements Methods*, The Rational Edge, July, 2002.
- [63] BERGER, P. M., *Instanciação de Processos para Projetos Específicos*, VII Workshop de Teses em Engenharia de Software (WTES 2002), 2002.
- [64] ISO/IEC 12207, *Information Technology – Software Life-Cycle Processes*, 1995.
- [65] BARROS, M. B., WERNER, C. M. L., TRAVASSOS, G. H., *Gerenciamento de Projetos Baseado em Cenários: uma Abordagem de Modelagem Dinâmica e Simulação*, I Simpósio Brasileiro de Qualidade de Software (SBQS 2002), 2002.





---

## Apêndice A

### Caracterização do Projeto

---

Será apresentado, a seguir, um formulário que serve de modelo para a realização da caracterização de projetos prevista no MAPS.

#### Tamanho da Equipe

- Muito pequena (1 a 6 pessoas)
- Pequena (7 a 20 pessoas)
- Média (21 a 50 pessoas)
- Grande (51 a 100 pessoas)
- Muito grande (mais de 100 pessoas)

#### Distribuição Geográfica da Equipe

- Mesma sala
- Mesmo prédio, salas diferentes
- Mesma cidade, mesma empresa, prédios diferentes
- Mesma cidade, empresas diferentes
- Cidades diferentes

#### Experiência da Equipe no Processo (em média)

- Nenhum projeto
- 1 projeto
- 2 a 3 projetos
- 4 a 5 projetos
- Mais de 5 projetos

#### Criticidade do Projeto (possível consequência de uma falha do sistema)

- Perda de conforto
- Prejuízos baixos, perdas facilmente recuperáveis
- Prejuízos moderados, perdas recuperáveis
- Prejuízos altos, perdas irre recuperáveis
- Risco de vida

## Tamanho do Projeto (investimento no projeto)

- Menos de R\$ 50.000,00
- Entre R\$ 50.000,00 e R\$ 150.000,00
- Entre R\$ 150.000,00 e R\$ 1.000.000,00
- Entre R\$ 1.000.000,00 e R\$ 3.000.000,00
- Mais de R\$ 3.000.000,00

---

## Apêndice B

### Avaliação Parcial do Projeto

---

Será apresentado, a seguir, um formulário que serve de modelo para a realização da avaliação parcial do projeto prevista no MAPS.

PARA ARTEFATOS QUE ESTÃO NO PROCESSO PADRÃO E NÃO ESTÃO NO PROCESSO ADAPTADO (PODE SER DIFERENÇA DE FORMALISMO OU COMPLEXIDADE)

**Artefato:**

Descrição:

Objetivos:

Como você avalia a utilidade do artefato para o projeto?

Muito útil     Útil     Pouco útil     Inútil

Como você avalia o custo/benefício do artefato?

Ótimo     Bom     Regular     Péssimo

O artefato deveria ter sido incluído no processo para esse projeto? Porquê? Em caso afirmativo, seria interessante incluir esse artefato agora, no meio do projeto?

O projeto tem alguma necessidade que não pode ser atendida por nenhum artefato do processo padrão? Qual? Que tipo de artefato solucionaria o problema?

## PARA ARTEFATOS QUE ESTÃO NO PROCESSO ADAPTADO

**Artefato:**

Descrição:

Objetivos:

Como você avalia a utilidade do artefato para o projeto?

 Muito útil     Útil     Pouco útil     Inútil

Qual é, aproximadamente, o custo de produção do artefato? (em pessoas/hora)

Como você avalia o custo/benefício do artefato?

 Ótimo     Bom     Regular     Péssimo

O artefato deveria ter sido incluído no processo para esse projeto? Porquê?

A artefato, nesse projeto, poderia ser produzido de forma mais simplificada ou informal? Como? Quais seriam as conseqüências?

---

## Apêndice C

### Avaliação Final do Projeto

---

Será apresentado, a seguir, um formulário que serve de modelo para a realização da avaliação final do projeto prevista no MAPS.

PARA ARTEFATOS QUE ESTÃO NO PROCESSO PADRÃO E NÃO FORAM UTILIZADOS NO PROCESSO ADAPTADO (PODE SER DIFERENÇA DE FORMALISMO OU COMPLEXIDADE)

**Artefato:**

Descrição:

Objetivos:

Como você avalia a utilidade do artefato para o projeto?

Muito útil     Útil     Pouco útil     Inútil

Como você avalia o custo/benefício do artefato?

Ótimo     Bom     Regular     Péssimo

O artefato deveria ter sido incluído no processo para esse projeto? Porquê?

O projeto teve alguma necessidade que não pode ser atendida por nenhum artefato do processo padrão? Qual? Que tipo de artefato solucionaria o problema?

## PARA ARTEFATOS QUE FORAM UTILIZADOS NO PROCESSO ADAPTADO

**Artefato:**

Descrição:

Objetivos:

Como você avalia a utilidade do artefato para o projeto?

 Muito útil     Útil     Pouco útil     Inútil

Qual foi, aproximadamente, o custo de produção do artefato? (em pessoas/hora)

Como você avalia o custo/benefício do artefato?

 Ótimo     Bom     Regular     Péssimo

O artefato deveria ter sido incluído no processo para esse projeto? Porquê?

A artefato, nesse projeto, poderia ter sido produzido de forma mais simplificada ou informal? Como? Quais seriam as conseqüências?

## SOBRE O PROJETO COMO UM TODO

Como você avaliaria o nível de formalismo do processo em relação às características do projeto?

- Muito formal
- Um pouco formal
- Adequado
- Um pouco informal
- Muito informal

Que melhorias no processo utilizado você recomendaria para um projeto com características semelhantes?

Que métodos, técnicas ou ferramentas utilizados no projeto poderiam ser incluídos no processo padrão da organização?





# Índice Remissivo

- A**
- adaptação, 5, 9, 47
  - Ambiente, 33, 35, 38
  - Artefatos, 34, 85
  - Atividades, 34
  - avaliação
    - final, 52, 53, 151
    - parcial, 52, 53, 149
  - Avaliação da Iteração, 90
  - Avaliação de Status, 90
- B**
- Banco de Dados de Processos de Software da Organização, 11, 50
  - Base de Processos, 48, 50, 73
  - Biblioteca de Documentação
    - Relacionada a Processos de Software, 11, 50
  - BORE, 135
- C**
- Capability Maturity Model, 9, 49
  - Capability Maturity Model Integrated, 49
  - características
    - de desenvolvimento, 59, 69, 70
    - restritivas, 59, 60, 68, 72
  - configuração, 5
  - Criticidade do software, 59
- D**
- Crystal, 22
  - DEF-PRO, 134
  - Disciplinas, 34
  - Distribuição geográfica da equipe, 59
- E**
- Estudo de Viabilidade, 89
  - experiência da equipe, 59
- F**
- Fases, 35
- K**
- Key Process Areas, 13
- L**
- Lista de Pendências, 96
  - Lista de Riscos, 92
- M**
- Medições do Projeto, 96
  - Methodology Explorer, 56
  - metodologia, 5
  - metodologias
    - ágeis, 4, 22
    - tradicionais, 4, 22
  - Modelo de Adaptação de Processos de Software, 6, 7, 47, 48, 49, 50, 51, 54,

55, 56, 57, 58, 75, 82, 86, 97, 109,  
138  
Modelo de Caracterização de Projetos,  
48, 50, 52, 53, 72, 74, 75, 78, 79

## O

OPEN, 17, 18  
Ordem de Trabalho, 92

## P

Papéis, 34  
PConfig, 48, 49, 50, 81  
Planejamento e Gerenciamento, 6, 33,  
35, 42  
Plano de Aceitação do Produto, 93  
Plano de Desenvolvimento de Software,  
87  
Plano de Garantia da Qualidade, 93  
Plano de Gerenciamento de Riscos, 91  
Plano de Iteração, 89  
Plano de Medições, 95  
Plano de Resolução de Problemas, 91  
PMBOK, 42  
prioridades do projeto, 22, 59, 60, 70,  
71, 72, 74  
processo, 1, 5  
adaptado, 6, 24, 48, 52, 53, 76, 79,  
80, 82, 83, 86, 117, 132, 133  
especializado, 6, 83

padrão, 2, 3, 4, 6, 7, 10, 15, 28, 29,  
31, 32, 33, 39, 47, 48, 49, 50, 52,  
54, 55, 56, 74, 75, 76, 79, 80, 81,  
82, 83, 84, 85, 110, 111, 112, 113,  
114, 115, 118, 120, 121, 122, 126,  
128, 129, 131, 132, 133, 134, 135,  
136, 138, 149, 151, 153

Processo de Software Padrão da  
Organização, 10

Projeto A, 113

Projeto B, 120

Projeto de Software Definido do  
Projeto, 11

ProKnowHow, 56, 135

## R

Rational Process Workbench, 56, 77,  
108

Rational Unified Process, 7, 33, 49, 54,  
77

Registro de Revisão, 95

## T

Tamanho da equipe, 25, 59

Tamanho do projeto, 24, 59

## X

XP, 17, 22