

BRUNO TORRES SATLER

**SELEÇÃO DE MELHORES PRÁTICAS DE ENGENHARIA DE
SOFTWARE COM BASE EM PARÂMETROS EXTRAÍDOS DO
AMBIENTE DO PROBLEMA**

Dissertação apresentada à
Universidade Federal de Viçosa, como
parte das exigências do Programa de Pós-
Graduação em Ciência da Computação,
para obtenção do título de *Magister
Scientiae*.

VIÇOSA
MINAS GERAIS - BRASIL
2010

BRUNO TORRES SATLER

**SELEÇÃO DE MELHORES PRÁTICAS DE ENGENHARIA DE
SOFTWARE COM BASE EM PARÂMETROS EXTRAÍDOS DO
AMBIENTE DO PROBLEMA**

Dissertação apresentada à
Universidade Federal de Viçosa, como
parte das exigências do Programa de Pós-
Graduação em Ciência da Computação,
para obtenção do título de *Magister
Scientiae*.

APROVADA: 28 de setembro de 2010.

Prof. Alcione de Paiva Oliveira
(Co-orientador)

Prof. André Gustavo dos Santos
(Co-orientador)

Prof. Clarindo Isaías Pereira da Silva e Pádua

Prof. Mauro Nacif Rocha

Prof. José Luis Braga
(Orientador)

*Dedico esta conquista a meus pais, que tiveram a sabedoria
de ensinar seus filhos a importância do precioso
bem chamado conhecimento. Às minhas
amadas esposa e filha pelo carinho.
A vocês, minha eterna gratidão.*

AGRADECIMENTOS

A Deus, pelo dom da vida, saúde e inspiração para realização desse trabalho. Agradeço também por escolher para mim a família maravilhosa em que nasci e por colocar em meu caminho tantas pessoas iluminadas, sem as quais nada disso teria sentido.

Aos meus pais José e Maria Aparecida que sempre dedicaram suas vidas a mim e a meu irmão, nos ensinando não só com palavras, mas também com exemplos a honestidade, humildade e retidão. Devo tudo que sou a vocês.

A meu irmão Rodrigo por sua amizade incondicional e por estar sempre presente.

A minha esposa e eterna namorada Marina pelo amor, compreensão e dedicação, sem você não tenho certeza se teria conseguido.

A minha filha Maria de Cássia, por ser o sol das minhas manhãs e a força que me move. Cada sorriso e gesto seu me tornou mais forte. Obrigado simplesmente por existir.

Ao meu orientador, conselheiro e amigo, professor José Luis Braga, por acreditar que eu seria capaz.

Aos meus co-orientadores professor Alcione e professor André pela contribuição no trabalho.

A professora Liziane Soares pelas boas idéias e grande ajuda na revisão do texto.

Aos grandes amigos Igor Guadalupe, Antônio Júnior, Páblisson Araújo e Clayton Carneiro pela amizade e alegria, e meus amigos e companheiros da graduação, “Maneh”, “Cefet”, “Biel”, “PH” e “Stein”, por compartilharem comigo bons momentos.

Aos muitos amigos que fiz na Cientec, em especial ao Alessandro e Paulo Márcio pela oportunidade ímpar de crescimento profissional e pessoal.

A todos que contribuíram e participaram comigo dessa jornada minha sincera gratidão.

BIOGRAFIA

BRUNO TORRES SATLER, filho de José Fátima Satler e Maria Aparecida Benevides Torres Satler, brasileiro, nascido em 22 de novembro de 1983, na cidade de Timóteo no estado de Minas Gerais.

No ano de 2003 ingressou no curso de Bacharelado em Ciência da Computação da Universidade Federal de Viçosa – UFV, concluído em 2007. Em janeiro de 2008 foi aprovado no programa de pós-graduação do DPI - Departamento de Informática da UFV, onde em março de 2008 iniciou o mestrado em Ciência da Computação, defendendo sua dissertação em setembro de 2010.

Realizou entre janeiro de 2005 e dezembro de 2007 um estágio em desenvolvimento de sistemas na Cientec – Consultoria e Desenvolvimento de Sistemas, onde desenvolveu entre outras atividades dois projetos de Iniciação Científica de parcerias entre universidade e empresa. Em janeiro de 2008 assumiu o cargo de gerente de projetos e posteriormente de analista de processos na mesma empresa onde atua até hoje.

SUMÁRIO

| | |
|--|-------------|
| LISTA DE FIGURAS..... | vii |
| LISTA DE TABELAS..... | viii |
| LISTA DE ABREVIATURAS..... | ix |
| RESUMO..... | x |
| ABSTRACT | xi |
| 1 INTRODUÇÃO | 1 |
| 1.1 O problema e sua importância | 2 |
| 1.1.1 Cenário das Micro e Pequenas Empresas | 4 |
| 1.2 Objetivos..... | 7 |
| 1.3 Metodologia..... | 8 |
| 1.4 Organização da dissertação | 10 |
| 2 REFERENCIAL TEÓRICO | 11 |
| 2.1 Visão Geral | 11 |
| 2.2 Processo de Desenvolvimento de Software | 12 |
| 2.2.1 Métodos Dirigidos por Planejamento (MDP)..... | 12 |
| 2.2.2 Métodos Ágeis (MA)..... | 13 |
| 2.2.3 Processo Híbrido | 14 |
| 2.3 Customização de Processo de Software | 14 |
| 2.3.1 Framework para Balanceamento de MA e MDP..... | 14 |
| 2.3.2 Discriminantes para Customização de Processo | 19 |
| 2.4 Goal Question Metric (GQM) | 21 |
| 2.5 Modelos de Qualidade de Processo | 23 |
| 2.5.1 CMM, SW-CMM e CMMI..... | 24 |
| 2.5.2 MPS.BR – Melhoria do Processo de Software Brasileiro | 26 |
| 3 DIAGNÓSTICO DO PERFIL DE COMPLEXIDADE DO AMBIENTE | 30 |
| 3.1 Características do Perfil..... | 30 |
| 3.2 Aplicação da Abordagem GQM | 32 |
| 3.2.1 Projeto da Estrutura do Questionário..... | 32 |
| 3.2.2 Elaboração do Questionário..... | 39 |
| 3.3 Análise e Interpretação do Questionário | 40 |
| 4 CONJUNTO DE BOAS PRÁTICAS – BUILDING BLOCKS | 50 |
| 4.1 Investigação de Práticas para nível G MPS.BR | 50 |
| 4.1.1 Gerência de Projeto (Complexidade Gerencial)..... | 50 |
| 4.1.2 Gerência de Requisitos (Complexidade Técnica) | 53 |
| 4.2 Investigação de Práticas CMMI | 55 |
| 4.2.1 Gestão de Projetos | 55 |
| 4.2.2 Engenharia | 57 |
| 4.3 Levantamento e Estrutura de Boas Práticas | 60 |
| 4.4 Lista de Boas Práticas – <i>Building Blocks</i> | 65 |
| 4.4.1 Relação das práticas com nível G do MPS.BR | 67 |

| | | |
|----------|---|------------|
| 5 | SELEÇÃO DE PRÁTICAS BASEADO EM PARÂMETROS DO PROBLEMA. | 70 |
| 5.1 | Visão Geral | 70 |
| 5.2 | Relacionamento com as Primitivas do Processo de Software | 70 |
| 5.3 | Mapeamento de Complexidade Técnica e Gerencial | 76 |
| 5.3.1 | Divisão dos Quadrantes de Complexidade | 76 |
| 5.3.2 | Posicionamento do Perfil de Complexidade nos Quadrantes | 77 |
| 5.4 | Proposição do Conjunto de Práticas | 82 |
| 5.4.1 | Práticas para Complexidade Baixa do Ambiente | 83 |
| 5.4.2 | Práticas para Complexidade Média Gerencial do Ambiente | 85 |
| 5.4.3 | Práticas para Complexidade Média Técnica do Ambiente | 87 |
| 5.4.4 | Práticas para Complexidade Alta do Ambiente | 88 |
| 6 | ESTUDO DE CASO | 90 |
| 6.1 | Formato de Apresentação dos Estudos de Caso | 90 |
| 6.2 | Empresa 1 | 91 |
| 6.3 | Empresa 2 | 94 |
| 7 | CONCLUSÕES E PERSPECTIVAS | 98 |
| 7.1 | Principais Contribuições | 99 |
| 7.2 | Trabalhos Futuros | 100 |
| | APÊNDICE A | 101 |
| | APÊNDICE B | 109 |
| | APÊNDICE C | 112 |
| | REFERÊNCIAS BIBLIOGRÁFICAS | 114 |

LISTA DE FIGURAS

| | |
|---|----|
| Figura 1.1- Porte das empresas, segundo força de trabalho efetiva. Resultado da pesquisa de 2009 do MCT | 5 |
| Figura 1.2 – Representação do objetivo geral do trabalho | 7 |
| Figura 1.3 – Diagrama de atividades dos passos de realização do trabalho | 9 |
| Figura 2.1 – Mapa mental de relação entre o referencial teórico e os capítulos do trabalho | 11 |
| Figura 2.2 - Resumo do método proposto em Boehm and Turner (adaptado: Boehm and Turner 2004a)..... | 16 |
| Figura 2.3 – Gráfico polar com cinco fatores críticos (adaptado: Boehm and Turner 2004a) | 18 |
| Figura 2.4 – Duas dimensões primárias de variação de processo (adaptado: Royce 1998) | 19 |
| Figura 2.5 – Prioridades do framework de customização de processos (adaptado: Royce 1998)..... | 21 |
| Figura 2.6 - Estrutura Hierárquica da Abordagem GQM (adaptado: Basili, 1994)..... | 23 |
| Figura 2.7 - Níveis propostos no CMM (Paulk et al., 1993)..... | 25 |
| Figura 2.8 – Componentes do Modelo MPS.BR..... | 28 |
| Figura 3.1 – Relacionamento entre os parâmetros de avaliação de processos. | 30 |
| Figura 3.2 - Gráfico de sete eixos do perfil de complexidade do ambiente | 32 |
| Figura 3.3 – Mapeamento dos intervalos da escala de perdas..... | 44 |
| Figura 4.1 – Áreas de Gestão de Projetos Básica (adaptado: SEI, 2006)..... | 56 |
| Figura 4.2 – Áreas do Processo de Engenharia (adaptado: SEI, 2006) | 58 |
| Figura 4.3 – Princípios de um processo moderno de software (Royce 1998)..... | 63 |
| Figura 4.4 – Elementos para apoio às boas práticas de desenvolvimento (adaptado: Leal, 2009) | 64 |
| Figura 5.1 – Quadrantes de complexidade do ambiente | 77 |
| Figura 5.2 – Quadrantes de complexidade do ambiente em termos das pontuações..... | 79 |
| Figura 5.3 – Organização do conjunto de práticas propostas para cada perfil identificado | 83 |
| Figura 6.1 – Gráfico radial traçado com o perfil de complexidade do ambiente da empresa 1 .. | 92 |
| Figura 6.2 – Gráfico radial traçado com o perfil de complexidade do ambiente da empresa 1 .. | 96 |

LISTA DE TABELAS

| | |
|--|-----|
| Tabela 2.1 – Riscos associados ao projeto (adaptado de Boehm and Turner 2004a)..... | 15 |
| Tabela 2.2 – Níveis de Maturidade e Processo MPS.BR | 29 |
| Tabela 3.1 – GQM na quantificação do fator Escala..... | 33 |
| Tabela 3.2 – GQM na quantificação do fator Dinamismo..... | 34 |
| Tabela 3.3 – GQM na quantificação do fator Criticalidade / Flexibilidade | 35 |
| Tabela 3.4 – GQM na quantificação do fator Cultura / Maturidade em Processo..... | 35 |
| Tabela 3.5 – GQM na quantificação do fator Previsibilidade Arquitetural | 36 |
| Tabela 3.6 – GQM na quantificação do fator Experiência no Domínio | 37 |
| Tabela 3.7 – GQM na quantificação do fator Competência Pessoal | 38 |
| Tabela 3.8 – Classificação do fator Escala | 41 |
| Tabela 3.9 - Classificação do fator Dinamismo | 43 |
| Tabela 3.10 – Classificação do fator Criticalidade / Flexibilidade..... | 44 |
| Tabela 3.11 – Classificação do fator Cultura / Maturidade m Processo | 45 |
| Tabela 3.12 – Classificação do fator Previsibilidade Arquitetural | 46 |
| Tabela 3.13 – Classificação do fator Experiência no Domínio | 47 |
| Tabela 3.14 – Níveis de Competência Pessoal (Cockburn 2002)..... | 48 |
| Tabela 3.15 – Pontuação necessária para cada nível de habilidade..... | 49 |
| Tabela 3.16 – Classificação do fator Competência Pessoal | 49 |
| Tabela 4.1 – Relação entre GPRs do nível G e práticas selecionadas | 67 |
| Tabela 4.2– Relação entre GREs do nível G e práticas selecionadas..... | 68 |
| Tabela 5.1 – Relação das primitivas de processo com o fator Escala (adaptado de Royce, 1998) | 71 |
| Tabela 5.2 – Relação das primitivas de processo com o fator Dinamismo (adaptado de Royce, 1998) | 72 |
| Tabela 5.3 - Relação das primitivas de processo com o fator Criticalidade/Flexibilidade (adaptado de Royce, 1998)..... | 73 |
| Tabela 5.4 - Relação das primitivas de processo com o fator Cultura/Maturidade em Processo (adaptado de Royce, 1998)..... | 74 |
| Tabela 5.5 - Relação das primitivas de processo com o fator Previsibilidade Arquitetural (adaptado de Royce, 1998)..... | 75 |
| Tabela 5.6 - Relação das primitivas de processo com o fator Experiência no Domínio | 75 |
| Tabela 5.7 - Relação das primitivas de processo com o fator Competência Pessoal | 76 |
| Tabela 5.8 – Práticas gerenciais para o quadrante de complexidade baixa do ambiente | 84 |
| Tabela 5.9 – Práticas técnicas para o quadrante de complexidade baixa do ambiente..... | 84 |
| Tabela 5.10 – Práticas gerenciais para o quadrante de complexidade média gerencial do ambiente | 85 |
| Tabela 5.11– Práticas técnicas para o quadrante de complexidade média técnica do ambiente.. | 87 |
| Tabela 5.12 – Práticas gerenciais para o quadrante de complexidade alta do ambiente | 88 |
| Tabela 5.13 – Práticas técnicas para o quadrante de complexidade alta do ambiente..... | 88 |
| Tabela 6.1 – Classificação da empresa 1 nos sete fatores de complexidade do ambiente..... | 92 |
| Tabela 6.2 – Pontuação do perfil de complexidade do ambiente da empresa 1 | 93 |
| Tabela 6.3 – Classificação da empresa 2 nos sete fatores de complexidade do ambiente..... | 95 |
| Tabela 6.4 – Pontuação do perfil de complexidade do ambiente da empresa 2 | 96 |
| Tabela B.1 – Pesos das respostas do questionário gerencial | 109 |
| Tabela B.2 – Pesos das respostas do questionário individual..... | 110 |
| Tabela C.1 – Respostas do questionário gerencial da empresa 1 | 112 |
| Tabela C.2 – Respostas do questionário individual da empresa 1 | 112 |
| Tabela C.3 – Respostas do questionário gerencial da empresa 2 | 113 |
| Tabela C.4 – Respostas do questionário individual da empresa 2..... | 113 |

LISTA DE ABREVIATURAS

- APF – Análise de Pontos de Função
- CMM – *Capability Maturity Model* (ou Modelo de Capacidade e Maturidade)
- CMMI - *Capability Maturity Model Integration*
- CMMI-DEV – *Capability Maturity Model Integration for Development* (Integração de Modelos de Maturidade da Capacidade para Desenvolvimento)
- EAP – Estrutura Analítica do Projeto
- ES – Engenharia de Software
- GPR – Gerência de Projeto
- GRE – Gerência de Requisitos
- IEEE - *Institute of Electrical and Electronics Engineers*
- ISO/IEC - *International Organization for Standardization / International Electrotechnical Commission*
- MA – Métodos Ágeis
- MA-MPS – Método de Avaliação para Melhoria de Processo de Software
- MDP – Métodos Dirigidos por Planejamento
- MN-MPS – Modelo de Negócio para Melhoria de Processo de Software
- MPE – Micro e Pequena Empresa
- MPS.BR – Melhoria de Processo do Software Brasileiro
- MR-MPS – Modelo de Referência para Melhoria de Processo de Software
- PF – Pontos de Função
- PI – *Product Integration* (Integração de Produto)
- PMC – *Project Monitoring and Control* (Monitoramento e Controle de Projeto)
- PP – *Project Planning* (Planejamento de Projeto)
- RD – *Requirements Development* (Desenvolvimento de Requisitos)
- REQM – *Requirements Management* (Gestão de Requisitos)
- RUP – *Rational Unified Process*
- SAM – *Supplier Agreement Management* (Gestão de Contrato com Fornecedores)
- SCAMPI – *Standard CMMI Appraisal Method for Process Improvement* (Método Padrão de Avaliação do CMMI para Melhoria de Processo)
- SOFTEX – Associação para Promoção da Excelência do Software Brasileiro
- SW-CMM – *Capability Maturity Model for Software*
- TS – *Technical Solution* (Solução Técnica)
- VAL – *Validation* (Validação)
- VER – *Verification* (Verificação)

RESUMO

SATLER, Bruno Torres. M. Sc., Universidade Federal de Viçosa, setembro de 2010. **Seleção de melhores práticas de engenharia de software com base em parâmetros extraídos do ambiente do problema.** Orientador: José Luis Braga. Co-orientadores: Alcione de Paiva Oliveira e André Gustavo dos Santos.

Atualmente, a utilização de software é tratada como elemento estratégico nas organizações, sobretudo na diferenciação de produtos e serviços. Processos de desenvolvimento e mais especificamente as práticas para desenvolvimento, são estudados dentro da engenharia de software como um dos principais mecanismos para se obter software de qualidade e cumprir corretamente os contratos de desenvolvimento. A engenharia de software é, portanto, a área da ciência da computação mais visivelmente relacionada com o desenvolvimento propriamente dito de softwares, através dos aspectos de desenvolvimento com qualidade, processos e práticas de desenvolvimento e com as técnicas de gerenciamento de processo. Sabemos que problemas presentes em diversas áreas são resolvidos através do desenvolvimento de software. Cada uma dessas áreas apresenta diferentes tipos de perfis de problemas e estão inseridos em ambientes distintos, sendo que cada perfil apresenta características específicas inerentes a seu contexto. O presente trabalho apresenta uma metodologia para adoção de práticas de engenharia de software (técnicas e gerenciais), para micro e pequenas empresas, baseada em parâmetros extraídos do ambiente em que os problemas estão inseridos. Para isso é traçado um perfil de complexidade do ambiente, baseado na classificação de sete fatores que são obtidos através da abordagem GQM. O trabalho apresenta também um levantamento de boas práticas de engenharia de software, baseado em modelos de qualidade de processos consagrados como CMMI e MPS.BR. As práticas propostas têm inicialmente um caráter de independência entre si, sendo chamadas no trabalho de *building blocks*. O mapeamento das práticas para determinado perfil se dá através do posicionamento do mesmo em um dos quatro quadrantes de complexidade (baixa, média técnica, média gerencial e alta). São realizados também estudos de caso nos quais o método de diagnóstico e mapeamento propostos foram aplicados a duas empresas diferentes com o objetivo de sugerir boas práticas de engenharia de software adequadas a seus respectivos perfis.

ABSTRACT

SATLER, Bruno Torres. M. Sc., Universidade Federal de Viçosa, September, 2010. **Selection of best practices of software engineering based on parameters extracted from the environment of the problem.** Advisor: José Luis Braga. Co-advisors: Alcione de Paiva Oliveira and André Gustavo dos Santos.

Currently, the use of software is treated as a strategic element in organizations, especially in the differentiation of products and services. Development processes and more specifically the practices for development, are studied in software engineering as a major mechanism for achieving quality software and properly fulfill contracts for the development. Software engineering is therefore the area of computer science more visibly linked to the actual development of software applications through development aspects of quality, processes and practices of development and technical management processes. We know that problems in different areas are solved through the development of software. Each of these areas has different types of problems and is embedded in different environments, each profile having specific characteristics inherent in its context. This paper presents a methodology for the proposition of software engineering practices (technical and managerial) for micro and small companies, based on parameters extracted from the environment in which the problems belong. For this, was trace a profile of complexity of the environment, based on the classification of seven factors that are obtained through the GQM approach. The paper also presents a collection of best practices in software engineering, based on recognized quality model processes as CMMI and MPS.BR. The proposed practices have initially a character of independence among them and are called, in this work, of building blocks. The mapping of practices for a certain profile is made through the placement in one of four quadrants of complexity (low, average technical, middle management and high). In addition was carried out case studies in which the method of diagnosis and mapping proposed were applied to two different companies in order to suggest software engineering best practices appropriate to their respective profiles.

1 INTRODUÇÃO

Apesar das turbulências que marcaram o ano de 2008, com grandes variações nos indicadores econômicos internacionais, o mercado brasileiro de software e serviços manteve a 12ª posição no cenário mundial, tendo movimentado 15 bilhões de dólares, equivalente a 0,96% do PIB brasileiro daquele ano. Deste total, foram movimentados 5 bilhões em software, o que representou perto de 1,68% do mercado mundial. Os restantes 10 bilhões foram movimentados em serviços relacionados (ABES 2009).

Em 2008, a participação de programas de computador desenvolvidos no país atingiu 32,5 % do total do mercado brasileiro de software e, embora tenha representado uma participação ligeiramente menor do que no ano anterior, ainda indica a tendência de crescimento que vem sendo apontada desde 2004, quando a participação era de 27%.

No mercado mundial, a participação das empresas brasileiras é ainda mais irrisória: dos US\$ 617 bilhões movimentados em 2004, o Brasil contribuiu com apenas US\$ 126 milhões, o que representa míseros 0,002%. Para se ter uma idéia, a Índia exportou US\$ 16,3 bilhões no setor de software em 2005, com um crescimento de 32% com relação ao ano anterior.

O desenvolvimento de software está, em geral, atrelado a um processo de desenvolvimento que é um conjunto de atividades, parcialmente ordenadas, com a finalidade de obter um produto final. O processo de desenvolvimento é estudado dentro da engenharia de software, sendo considerado um dos principais mecanismos para se obter software de qualidade e cumprir corretamente os contratos de desenvolvimento.

A engenharia de software é, portanto, a área da ciência da computação mais visivelmente relacionada com o desenvolvimento propriamente dito de softwares, através dos aspectos de desenvolvimento com qualidade, dos processos e práticas de desenvolvimento e com as técnicas de gerenciamento de processo. Problemas presentes em diversas áreas são resolvidos através do desenvolvimento de software. Cada uma dessas áreas apresenta diferentes tipos de perfis de problemas, sendo que cada perfil apresenta características específicas inerentes a seu contexto.

Devemos então entender a natureza do software, que é um produto diferenciado dos demais, possuindo características que lhe dão uma natureza própria. Conhecer essas características é o primeiro passo para compreender os problemas enfrentados pelos desenvolvedores, e tentar lançar uma luz sobre as razões da existência da “crise do

software”. A partir da análise do software em estudos anteriores, foram enumeradas as seguintes características (Brooks, 1987; Kruchten, 2001):

- **Grande complexidade:** softwares representam modelos do mundo real, assim cada detalhe que o software se propõe a resolver é um elemento a mais dentro da aplicação, tornando a complexidade cada vez maior;
- **Ausência de conformidade:** diversas áreas do conhecimento científico se baseiam na convicção de que existem princípios unificadores que, uma vez descobertos, facilitam sua compreensão. A física é um bom exemplo, com suas leis fundamentais, ela é utilizada como sustentáculo para a engenharia. Sistemas de software, ao contrário disso, não costumam existir em conformidade com princípios básicos e imutáveis. Sendo assim, atualmente, é impossível formar uma base sólida de conhecimento como a encontrada na engenharia tradicional. Essa inexistência de princípios básicos faz com que os padrões de software costumem se basear apenas em boas práticas;
- **Invisibilidade:** softwares não podem ser representados de maneira física, e não podem ser representados geometricamente. Mesmo os diagramas não descrevem o software em si, apenas fluxos de dados, fluxos de controle, padrões de dependência, entre outras.
- **Ilusão de maleabilidade:** por não possuir natureza física, o software transmite ilusão de alta maleabilidade, passando a impressão de baixo custo para qualquer eventual modificação. Se compararmos à construção de uma ponte, essa falta de flexibilidade é bem visível. Você não pode dizer: “Hum, agora que eu já vejo os pilares, eu gostaria que essa ponte fosse colocada duas milhas rio acima”.

1.1 O problema e sua importância

Não basta, contudo, conhecermos a natureza do desenvolvimento de software sem investigar os melhores métodos e técnicas, os procedimentos e ferramentas mais adequados e, porque não, o paradigma de desenvolvimento mais robusto para uma determinada situação. Um método ou técnica é um procedimento formal para produzir algum resultado. Uma ferramenta é um instrumento ou sistema automatizado utilizado para realizar uma tarefa da melhor maneira. Um procedimento é como uma receita: combinação de ferramentas e técnicas que, em harmonia, produzem um resultado específico. E por fim, um paradigma representa uma abordagem ou filosofia em

particular para a construção de software (Pfleeger, 2004).

Assim sendo, precisamos compreender a necessidade de se investir na melhoria do processo de desenvolvimento, através inicialmente, da escolha das melhores práticas para cada caso. A decisão de investir ou não depende, entre outras coisas, da análise do mercado em potencial, dos riscos envolvidos e das perspectivas de retorno do investimento.

Atualmente, a utilização de software é aplicada nas mais diversas áreas, e o mesmo assume um duplo papel, onde é produto e ao mesmo tempo, veículo para entrega de produto. A estrutura da maioria das organizações depende do suporte fornecido por sistemas de software incorporados às atividades, uma vez que ele trata o mais importante dos produtos atualmente, a informação. Dessa forma, o software torna-se um elemento estratégico na diferenciação de produtos e serviços (Pressman, 2001), e por isso faz-se necessária a aplicação de práticas, métodos, ferramentas, procedimentos ou mesmo paradigmas adequados para auxiliar o desenvolvimento desses sistemas.

Com o crescimento significativo da importância do software na economia atual e com tantas técnicas e métodos disponíveis para o desenvolvimento, a escolha do mais adequado para cada perfil de problema ainda é feita de maneira empírica, sem bases científicas que justifiquem a escolha e procurem a adequabilidade do método ou técnica ao perfil do problema em questão.

O mercado de software no Brasil é explorado por quase 8.500 empresas, dedicadas ao desenvolvimento, produção e distribuição de software e de prestação de serviços. Das que atuam no desenvolvimento e produção de software, 94% são classificadas como micro e pequenas empresas (ABES 2009).

Há algumas limitações básicas para que uma empresa seja considerada micro ou pequena empresa (MPEs) no Brasil, e possa aproveitar algumas vantagens desse status como, por exemplo, tributação diferenciada. Atualmente, há pelo menos três definições utilizadas para limitar o que seria uma pequena ou micro empresa.

A definição mais comum e mais utilizada é a que está na Lei Geral para Micro e Pequenas Empresas. De acordo com essa lei, que foi promulgada em dezembro de 2006, as Micro Empresas são as que possuem um faturamento anual de, no máximo, R\$ 240 mil. As Pequenas Empresas devem faturar entre R\$ 240.000,01 e R\$ 2,4 milhões anualmente para serem enquadradas como tal.

Outra definição vem do Serviço Brasileiro de Apoio às Micro e Pequenas Empresas (SEBRAE). A entidade limita as micro às que empregam até nove pessoas no

caso do comércio e serviços, ou até 19, no caso dos setores industrial ou de construção. Já as pequenas são definidas como as que empregam de 10 a 49 pessoas, no caso de comércio e serviços, e 20 a 99 pessoas, no caso de indústria e empresas de construção.

Já órgãos federais como Banco Nacional de Desenvolvimento Econômico e Social (BNDES) têm outro parâmetro para a concessão de créditos. Nessa instituição de fomento, uma microempresa deve ter receita bruta anual de até R\$ 1,2 milhão; as pequenas empresas, superior a R\$ 1,2 milhão e inferior a R\$ 10,5 milhões.

As principais limitações brasileiras estão justamente no porte da grande maioria das empresas (em geral micro e pequenas) e na qualificação insuficiente dos seus processos (MIT Project, 2002). O valor dos projetos a que uma empresa consegue se dedicar depende diretamente do tamanho da sua equipe e da sua capacitação.

1.1.1 Cenário das Micro e Pequenas Empresas

Em um contexto de produção, empresas de diferentes portes passam por dificuldades para obter qualidade em seus produtos finais. Em particular as MPEs tendem a ter maior dificuldade por trabalharem com processos informais e a escassez de recursos humanos e financeiros. No setor de software, a dificuldade ainda é maior devido à ubiquidade do produto e à complexidade dos processos produtivos. As principais dificuldades encontradas por empresas no desenvolvimento de software são, segundo (Jones, 1996 e Yourdon, 1997):

- ✓ incompreensão das necessidades do usuário final;
- ✓ produção de software difícil de manter e entender;
- ✓ baixo desempenho e baixa qualidade;
- ✓ recursos humanos atuando nas diversas fases do ciclo de vida do software sem um planejamento de atuação;
- ✓ inexistência de processos de rastreabilidade para melhor manutenção de códigos fontes;
- ✓ falta de conhecimento em métodos e técnicas de ES.

Além das dificuldades inerentes ao desenvolvimento do produto software, as MPEs têm presentes dificuldades do tipo:

- ✓ investimentos focados em linguagens de programação e desenvolvimento e não em ES;
- ✓ inexistência de processos organizados e aplicados;
- ✓ recursos humanos e financeiros limitados;

- ✓ baixo nível de conhecimento no domínio gerencial;
- ✓ inexistência de políticas de curto ou longo prazo que auxiliem na melhoria dos processos de desenvolvimento.

Com isso as MPEs encontram grandes problemas na sua produtividade, qualidade e competitividade no mercado, ocasionando na grande maioria das vezes o fechamento dessas empresas.

A falta de um processo sistemático de desenvolvimento de software é citada no relatório de 2005 do Ministério da Ciência e Tecnologia (MCT, 2005) como um dos pontos que prejudica a pequena empresa na qualidade e produtividade do processo e do produto desenvolvido. Um outro fator importante no insucesso das empresas frente a sua sobrevivência no mercado é o acesso limitado aos incentivos fiscais e de políticas de crédito para ciência e tecnologia. Além disso, frequentemente ignoram a existência destes incentivos (Rovere, 2000).

Em (Vos et al., 1998), cita-se que os administradores das MPEs tendem a ter um horizonte de planejamento de curto prazo, ficando presos aos problemas do dia a dia e impedidos de atuar na definição de estratégias de longo prazo e de inovação.

Segundo pesquisas do (MCT, 2009), a maior concentração de empresas de informática é de MPEs com até 49 pessoas. A Figura 1.1 apresenta a estatística por porte de empresa, segundo pesquisa do (MCT, 2009).

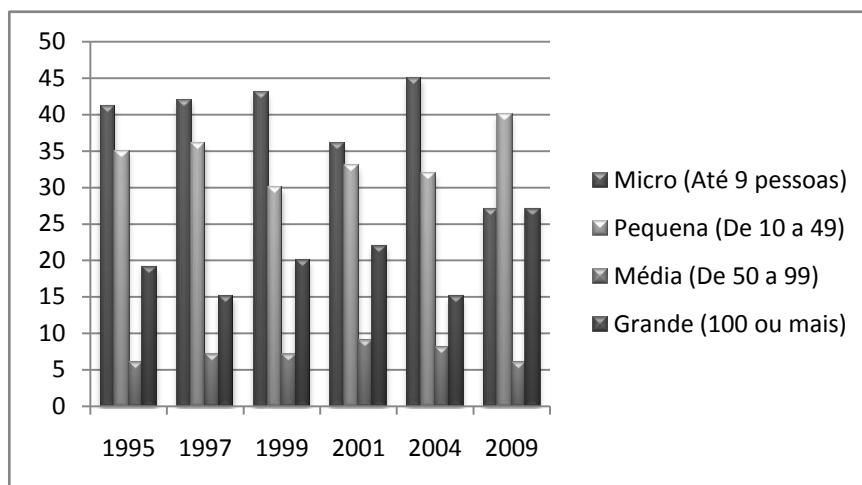


Figura 1.1- Porte das empresas, segundo força de trabalho efetiva. Resultado da pesquisa de 2009 do MCT

Com o intuito de auxiliar as empresas de software na busca pela melhoria na qualidade de seus processos produtivos existem no mercado nacional e internacional

vários modelos de qualidade: (MPS.BR, 2009), a série da Norma ISO 9000 (ABNT, 2001), CMM/CMMI (SEI, 2002), (ISO/IEC 15504, 2003) , entre outros.

Apesar das propostas de avaliação para melhoria dos processos organizacionais de desenvolvimento de software, as empresas envolvidas têm que dedicar um grande esforço até a qualificação oficial. Os modelos sugeridos não vêm acompanhados de técnicas ou métodos que possam ser seguidos para que sejam atingidos os objetivos. Por exemplo, para se atingir o nível G do modelo MPS.BR é necessário que a empresa cumpra uma série de passos, mas não há na abordagem a sugestão de práticas ou mesmo ferramentas computacionais sugeridas para se atingir uma melhor gerência de projetos e de requisitos. Portanto, o caminho até a avaliação requer da empresa muita dedicação, disciplina e a implantação de métodos e técnicas formais que sejam satisfatórias às exigências do nível requerido.

As pesquisas do MCT apresentam um baixo número de MPEs que atingem o nível de avaliação desejado. Isso é causado pela falta de aderência dos modelos à realidade dessas MPEs. Segundo pesquisas do (MCT, 2005), a parcela de MPEs com certificação ISO 9000 ou CMM até o ano de 2001 era apenas 7% do total. Apesar da possibilidade da adaptação dos modelos existentes para serem aderentes as MPEs, há uma necessidade de um maior esforço e de grande experiência para sua implantação, pois os modelos atuais atendem a um perfil de empresas que estão mais estruturadas em termos financeiros e organizacionais, caracterizando-se em um nível acima da grande maioria das empresas consideradas como MPEs.

Apesar da recompensa pelo ganho da qualidade, maturidade e controle dos processos de desenvolvimento de software, a opção pela implantação do modelo deve ser bem avaliada, pois a curva de aprendizado de um método é longa. Para (Aggarwal, 1998), a taxa de sobrevivência das pequenas empresas no mercado depende do ambiente tecnológico onde a empresa atua e do seu tempo de atuação. Curvas de aprendizado longas podem trazer dificuldades na implantação de políticas que requerem maior esforço de adoção, uma vez que a taxa da mortalidade das empresas no Brasil é elevada, sobretudo nos seus anos iniciais de existência.

Conforme estatísticas do (SEBRAE, 2004), os índices de encerramento de MPEs no Brasil são altos: 49,4% acabam antes de completar dois anos, 56,4% fecham durante os três primeiros anos e 59,9% nos quatro primeiros anos. Diante dessa realidade, são altas as chances das empresas encerrarem antes de terminarem o processo de certificação de melhoria de qualidade.

Portanto, há uma necessidade premente de essas empresas iniciarem suas atividades de melhoria da qualidade com passos mais curtos a partir de adoção de boas práticas da ES. Após a escolha e utilização das boas práticas, devem iniciar seu esforço de implantação dos níveis mais básicos de avaliação disponíveis nos atuais modelos no mercado. Com base no contexto, entende-se que serão minimizados os impactos iniciais das exigências desses níveis de avaliação nas empresas onde a cultura do formalismo é inexistente. Objetivando a implantação inicial de métodos mais aderentes à sua realidade, as MPEs terão condições de aumentar seu controle dos processos produtivos e melhorar as chances de sobrevivência no mercado, além de estarem mais preparadas para aumentar a competitividade.

1.2 Objetivos

O objetivo geral deste projeto é obter um mapeamento entre os perfis técnicos de micro e pequenas empresas, obtidos através de parâmetros investigados e extraídos do ambiente do problema, e um conjunto de melhores práticas de engenharia de software, como pode ser visto na Figura 1.2.

Especificamente pretende-se:

- ✓ Selecionar um conjunto de boas práticas de desenvolvimento de software.
- ✓ Determinar parâmetros que permitam estabelecer um modelo ou perfil de micro empresas produtoras de software.
- ✓ Definir uma relação justificada entre os modelos de empresa e as melhores práticas mais indicadas para eles.
- ✓ Testar a aplicabilidade da técnica.

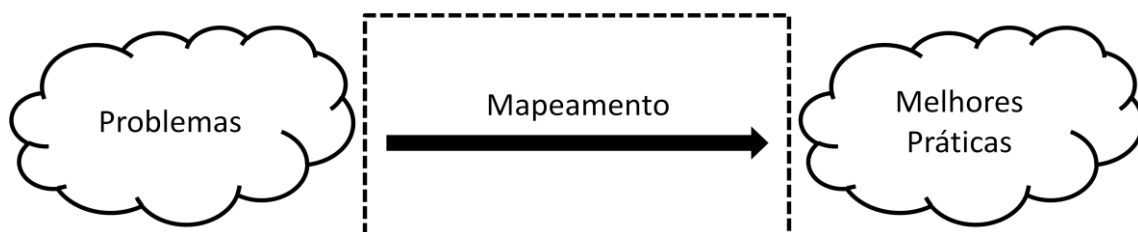


Figura 1.2 – Representação do objetivo geral do trabalho

1.3 Metodologia

A realização do trabalho foi composta de atividades que podem ser discriminadas como:

- ✓ **Atividade 1:** Investigação na literatura sobre critérios utilizados para customização de processo de desenvolvimento de software. Mais especificamente investigação dos cinco fatores críticos utilizados para determinar o risco de adoção de MA (Métodos Ágeis) ou MDP (Métodos Dirigidos por Planejamento) em projetos de software (Boehm, 2004 apud Soares, 2007) e os seis discriminantes de processo descritos por (Royce, 1998) como características para customização de processos de desenvolvimento de software.
- ✓ **Atividade 2:** Avaliação e combinação das características investigadas na atividade anterior para determinar os fatores para compor o **perfil de complexidade do ambiente**, que é o perfil proposto pelo autor para diagnosticar a situação da empresa (ambiente do problema).
- ✓ **Atividade 3:** Modificação da abordagem GQM (*Goal Question Metrics*) proposta por (Soares, 2007) para contemplar todos os fatores do **perfil de complexidade do ambiente**.
- ✓ **Atividade 4:** Investigação na literatura especializada a fim de compor uma estrutura bem definida das “melhores” práticas de Engenharia de Software para desenvolvimento de sistemas. Estudo específico da taxonomia de melhores práticas proposta por (Leal, 2009).
- ✓ **Atividade 5:** Separação das práticas de Engenharia em “blocos” independentes, chamados de “*building blocks*” classificados entre práticas gerenciais e práticas técnicas.
- ✓ **Atividade 6:** Investigação sobre a natureza do mapeamento em questão entre as empresas (perfil de complexidade do ambiente) e o conjunto de práticas definidas.
- ✓ **Atividade 7:** Relação propriamente dita entre o ambiente do problema e as práticas de Engenharia de Software propostas. O Mapeamento é realizado através do posicionamento do ambiente nos quatro quadrantes de (Royce, 1998) e a relação com as primitivas de processo descritas pelo mesmo autor.

A seqüência proposta está descrita na Figura 1.3 a seguir.

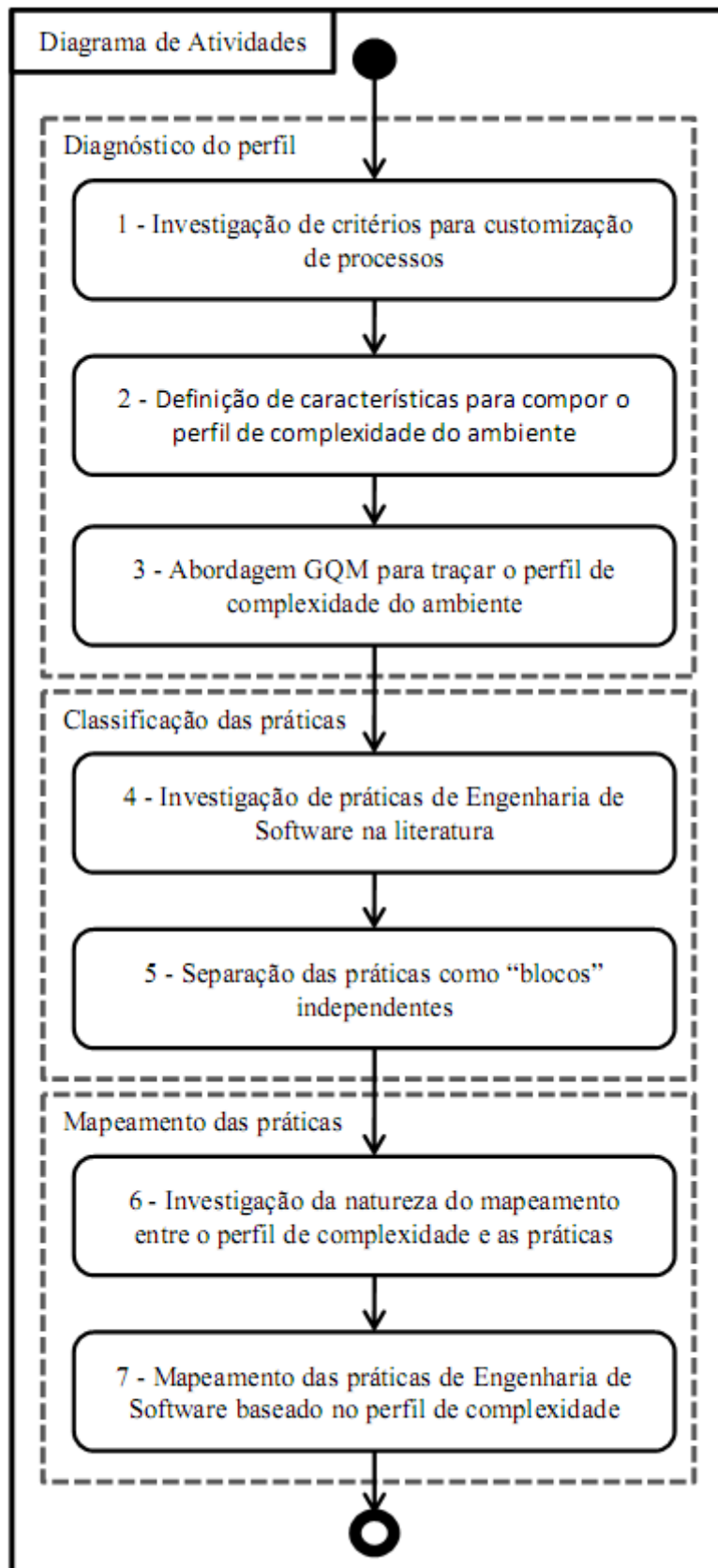


Figura 1.3 – Diagrama de atividades dos passos de realização do trabalho

1.4 Organização da dissertação

A dissertação está organizada da seguinte forma:

- ✓ Apresentação do problema e contexto
 - Capítulo 1. Apresenta o problema, sua importância e os objetivos do trabalho.
 - Capítulo 2. Descreve o referencial teórico, resultado da pesquisa bibliográfica sobre as áreas técnicas e tecnologias relacionadas ao trabalho.
- ✓ Desenvolvimento do tema e contribuições
 - Capítulo 3. Apresenta o método realizado na definição do perfil de complexidade de ambiente proposto, bem como o questionário e interpretação dos dados.
 - Capítulo 4. Descreve o estudo de melhores práticas de Engenharia de Software bem como a classificação das mesmas em “blocos” independentes.
 - Capítulo 5. Apresenta o relacionamento entre os fatores que determinam o perfil de complexidade do ambiente proposto e as primitivas de customização de processo, além do mapeamento propriamente dito das práticas.
- ✓ Apresentação dos resultados e fechamento
 - Capítulo 6. Apresenta um estudo de caso de aplicação da técnica em duas micro empresas de desenvolvimento de software.
 - Capítulo 7. Apresenta as conclusões, possíveis desdobramentos e trabalhos futuros desta dissertação.

Ao final do trabalho, são apresentados os apêndices contendo o questionário, as atribuições de pesos e as respostas do estudo de caso.

2 REFERENCIAL TEÓRICO

2.1 Visão Geral

Para facilitar o entendimento da relação entre os tópicos apresentados no referencial teórico e os demais capítulos do trabalho foi construído um mapa mental, que é um tipo de diagrama voltado para gestão de informações e de conhecimento. Um mapa mental é formado por nós, ramos e relações, organizados em níveis a partir de uma idéia central.

O mapa mental na Figura 2.1 apresenta essa relação. O Capítulo 3, que é o diagnóstico do perfil de complexidade do ambiente foi construído baseado principalmente nos métodos e processos de desenvolvimento de software (ágeis, dirigidos por planejamento e híbridos) e também através da abordagem GQM. O Capítulo 4, que trata a seleção de melhores práticas e as classifica em blocos independentes foi construído principalmente através dos modelos de qualidade de processos (CMMI e MPS.BR). E o Capítulo 5, que mostra o método proposto para mapeamento de práticas baseado em parâmetros do problema, está fundamentado no framework de balanceamento entre MA (métodos ágeis) e MDP (métodos dirigidos por planejamento) e nos quadrantes discriminantes para customização de processo.

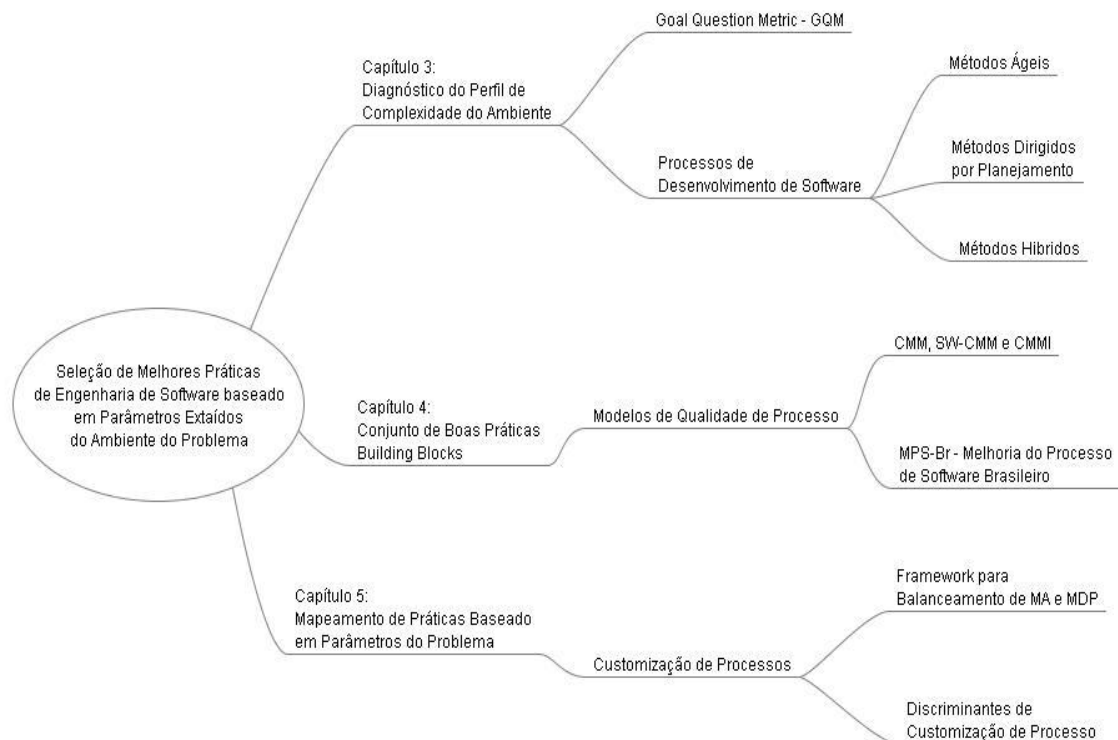


Figura 2.1 – Mapa mental de relação entre o referencial teórico e os capítulos do trabalho

2.2 Processo de Desenvolvimento de Software

Segundo (Filho, 2003), um processo é um conjunto de passos através do qual uma meta deve ser atingida. Pode ser composto de vários elementos, tais como atividades, métodos, ferramentas e artefatos. Um processo de software pode ser definido como uma seqüência de passos objetivando o desenvolvimento, manutenção ou gerenciamento de sistemas (Sommerville, 2004). Sua aplicação favorece aspectos ligados ao desenvolvimento do software, ao gerenciamento da qualidade e ao gerenciamento do projeto (Gruhn, 2002). Atividades como determinação de requisitos, análise, projeto, implementação e testes podem ser vistas como sub-processos (Filho, 2003).

A Tecnologia de Processo de Software surgiu no final da década de 1980 e representou um importante passo em direção à melhoria da qualidade de software através de mecanismos que proporcionam o gerenciamento automatizado do desenvolvimento de software (Feiler, 1993). Diversas teorias, conceitos, formalismos, metodologias e ferramentas surgiram nesse contexto enfatizando a descrição de um modelo de processo de software que é automatizado por um ambiente integrado de desenvolvimento.

Um modelo de processo descreve o processo de software e especifica as atividades a serem executadas, as ferramentas a serem utilizadas, os tipos de artefatos a serem manipulados e o papel de cada pessoa envolvida no desenvolvimento (Gruhn, 2002). As diversas atividades são realizadas individualmente por desenvolvedores, gerentes e usuários, de acordo com seu papel dentro do processo.

2.2.1 Métodos Dirigidos por Planejamento (MDP)

Os Métodos Dirigidos por Planejamento (MDP) estão mais próximos da maneira tradicional de se desenvolver software, utilizando um paradigma que envolve requisitos, projeto e desenvolvimento. Utilizam uma abordagem sistemática e disciplinada onde o software é construído a partir dos requisitos até o código final passando por uma série de fases, todas muito bem documentadas (Boehm, 2004).

Os MDP podem ser caracterizados pelos seguintes conceitos: melhoria e aquisição de competência em processos, maturidade organizacional, gerência de risco, gerência da qualidade, verificação e validação de sistemas, e utilização de arquiteturas de software que favoreçam a reutilização pelo uso de componentes (Boehm, 2004). Esses métodos se baseiam na premissa de que os desenvolvedores se sentem mais confortáveis

trabalhando com regras, procedimentos e políticas bem definidas e estabelecidas (Silva et al., 2006).

Os MDP são mais adequados para problemas com requisitos mais estáveis, os quais exigem processo definido para desenvolvimento, documentação detalhada e que podem implicar em riscos de vida para seus usuários e em riscos econômicos caso o software funcione mal ou não funcione (Boehm and Turner, 2004a).

Como exemplos de métodos dirigidos por planejamento, pode-se citar: Personal Software Process (PSP) (Humphrey, 2000a), Team Software Process (TSP) (Humphrey, 2000b) e Rational Unified Process (RUP) (Kruchten, 2004). Os MDP propiciam previsibilidade, estabilidade e maior garantia sobre o produto de software (Boehm and Turner, 2004b).

2.2.2 Métodos Ágeis (MA)

A agilidade possibilita que experiências já adquiridas sejam utilizadas na interação com novos ambientes, proporcionando reações rápidas a novos contextos e melhor adaptação a situações sem precedentes (Boehm and Turner, 2004a,b).

Os Métodos Ágeis (MA) são baseados nos princípios propostos em um manifesto definido por um grupo denominado *Agile Software Development Alliance* (Alliance, 2001). Os valores adotados favorecem o suporte ao surgimento de novos requisitos e enfatizam a importância da comunicação (Segal, 2005). Nesses métodos, a comunicação utilizada para troca de informações entre os membros da equipe é realizada, principalmente, de maneira informal ao invés da comunicação formal e documentada (Boehm, 2002). Além disso, o envolvimento dos clientes e usuários no processo de desenvolvimento é valorizado possibilitando que as dúvidas sobre o projeto sejam solucionadas e que as prioridades sejam ajustadas (Highsmith and Cockburn, 2001).

Os MA podem ser caracterizados pelos seguintes conceitos: permitir alterações e mudanças com facilidade (*embrace change*), ciclos curtos e frequentes entregas de versões, projeto simplificado, adoção de refatoração (*refactoring*) como técnica de melhoria e padronização no texto dos programas, programação em pares (*pair programming*), compartilhamento do conhecimento adquirido nos projetos e desenvolvimento dirigido por testes (*Test Driven Development - TDD*) (Boehm, 2004).

Esses métodos se baseiam na premissa de que os desenvolvedores se sentem mais confortáveis tendo um grau de liberdade maior para produzir o software (Silva et al., 2006) e são mais adequados para problemas sujeitos a alterações e que, portanto, exijam

a capacidade de adaptação mais frequente por parte do software (Boehm and Turner, 2004a).

Como exemplos de métodos ágeis, pode-se citar: *eXtreme Programming* (XP) (Beck, 2000), Crystal (Cockburn, 2002) e Scrum (Ambler, 2002; Methods, 2007).

2.2.3 Processo Híbrido

Um processo híbrido pode ser definido como aquele que combina características de Métodos Dirigidos por Planejamento (MDP) com características de Métodos Ágeis objetivando maior adequabilidade do processo ao tipo de desenvolvimento de software praticado (Boehm, 2004).

Os Métodos Dirigidos por Planejamento (MDP) são processos longos, com muita documentação, passos e resultados intermediários muito bem definidos, com padrões rigorosos de qualidade associados. Os Métodos Ágeis (MA) representam uma nova tendência em desenvolvimento de software, ainda não formalizada, mas com uso em ascensão (Boehm, 2004). A escolha de instâncias de um ou outro grupo para resolver problemas específicos ainda é um desafio a ser vencido e formalizado.

Podem ocorrer situações onde a disciplina e a documentação inerentes aos MDP tragam sobrecarga ao desenvolvimento. Um sistema simples, por exemplo, desenvolvido por uma equipe pequena, não precisa ter seu desenvolvimento sobrecarregado com excesso de documentação inerente a um processo muito extenso. Da mesma forma, um sistema complexo não pode conter falhas de desenvolvimento como consequência de um processo curto que apresenta uma baixa exigência de documentação formalizada e um planejamento de curtíssimo prazo, sem avaliação de riscos de longo prazo. O nível de detalhamento do processo deve se adaptar às características de cada problema (Boehm, 2004). Diante disso, os métodos e técnicas disponíveis podem gerar um gasto adicional de tempo e custo ao desenvolvimento, ou podem ser insuficientes como meios de estruturação e documentação.

2.3 Customização de Processo de Software

2.3.1 Framework para Balanceamento de MA e MDP

(Cockburn, 2000) propõe um framework para escolha da metodologia de desenvolvimento apropriada para um dado projeto. Segundo Cockburn considerar múltiplas metodologias é apropriado e necessário. A metodologia adequada é escolhida

considerando-se duas dimensões: o tamanho da equipe e a criticalidade do sistema.

(Boehm and Turner, 2004a) estendem o framework de Cockburn, propondo uma abordagem baseada em risco para balancear Métodos Ágeis e Métodos Dirigidos por Planejamento visando à obtenção de métodos híbridos que se adequem melhor às necessidades, objetivos, restrições e prioridade de um projeto ou organização.

O método proposto por (Boehm and Turner, 2004a) para balancear MA e MDP, é composto de cinco passos:

- ✓ Passo 1: avaliar os riscos associados ao projeto com relação ao ambiente, à agilidade e à disciplina (os riscos são mostrados na Tabela 2.1). Se houver dúvidas sobre algum aspecto da avaliação é indicado despende recursos através de prototipação, coleção de dados e análise para esclarecer esse aspecto.
- ✓ Passo 2: Se os riscos associados à agilidade prevalecem sobre os riscos associados à disciplina, um MDP é mais apropriado para o projeto. Se os riscos associados à disciplina prevalecem sobre os riscos associados à agilidade, um MA é mais apropriado para o projeto.
- ✓ Passo 3: esse passo trata dos casos onde o projeto não apresenta um contexto ágil bem definido nem um contexto disciplinado bem definido. Nesse caso, desenvolver uma arquitetura que suporte o uso de MA onde eles podem fortalecer o desenvolvimento e minimizar os riscos. Os métodos dirigidos por planejamento devem ser utilizados no restante do trabalho e devem ser escolhidos nos casos em que nenhuma outra arquitetura conveniente pode ser criada.
- ✓ Passo 4: desenvolver uma estratégia para todo o projeto para identificar os riscos. Identificar e integrar a estratégia para solução de cada risco.
- ✓ Passo 5: monitorar o progresso e os riscos apresentados pelo processo selecionado e realizar os ajustes necessários.

O resumo do método proposto por (Boehm and Turner, 2004a) pode ser visualizado na Figura 2.2.

Tabela 2.1 – Riscos associados ao projeto (adaptado de Boehm and Turner 2004a)

| |
|--|
| Risco do ambiente: riscos que resultam do ambiente geral do projeto |
| <ul style="list-style-type: none">• Incertezas relacionadas à tecnologia• Muitos tipos de pessoas envolvidas para serem coordenados• Complexidade do sistema |

| |
|--|
| <p>Riscos da agilidade: riscos que são específicos do uso de Métodos Ágeis</p> <ul style="list-style-type: none"> • Escalabilidade e criticalidade • Uso de projeto simples • Substituição de profissionais ou de clientes • Profissionais sem habilidades para trabalhar com Métodos Ágeis |
| <p>Riscos da disciplina: riscos que são específicos do uso de Métodos Dirigidos por Planejamento</p> <ul style="list-style-type: none"> • Mudanças rápidas • Necessidade de resultados rápidos • Surgimento de requisitos |

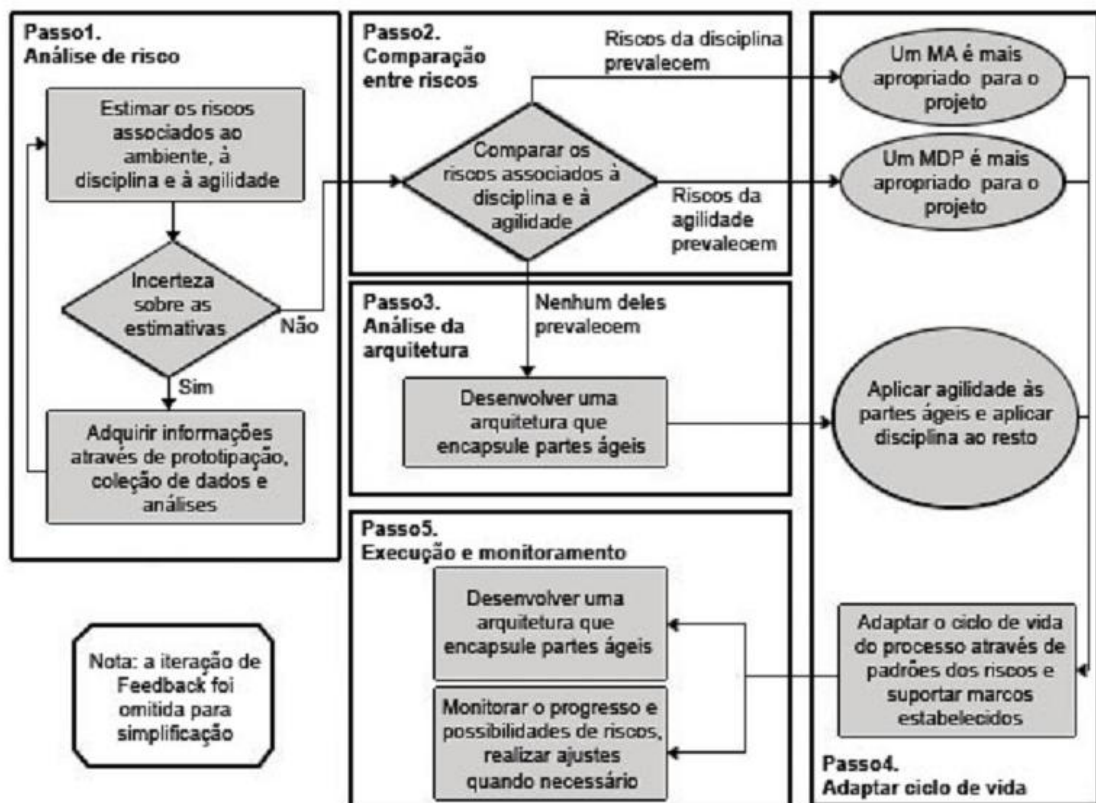


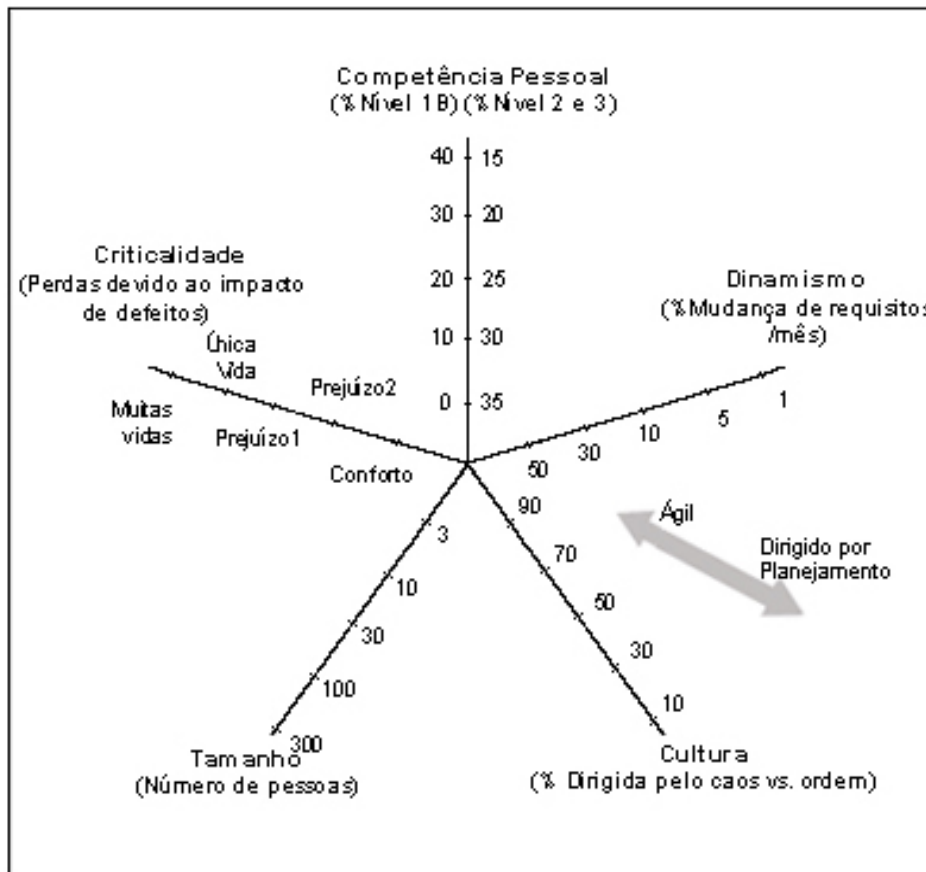
Figura 2.2 - Resumo do método proposto em Boehm and Turner (adaptado: Boehm and Turner 2004a)

São definidos por (Boehm and Turner, 2004a) cinco fatores críticos através dos quais a equipe e o contexto de desenvolvimento podem ser caracterizados. Estes cinco fatores servem como parâmetros para se determinar a adequação de MA ou MDP a projetos de software:

1. Tamanho: refere-se ao tamanho da equipe envolvida, e ao esforço necessário no desenvolvimento.

2. **Criticalidade:** refere-se às perdas ocasionadas em função de erros presentes no sistema. Relaciona-se ao risco a vidas humanas pelo uso do software e a outros riscos menos graves, com impacto direto no nível de qualidade e possibilidade de falhas.
3. **Dinamismo:** refere-se à possibilidade de mudanças nos requisitos em função de mudanças no ambiente do problema implicando em alterações no software.
4. **Competência Pessoal:** refere-se às habilidades dos profissionais envolvidos no desenvolvimento. Um profissional pode ser associado a três níveis de habilidade diferentes os quais podem ser visualizados na Tabela 3.14 apresentada no Capítulo 3.
5. **Cultura:** refere-se à cultura dos desenvolvedores em relação ao desenvolvimento e gerência de projetos de software. Os desenvolvedores podem possuir a cultura na qual se sentem mais confortáveis tendo muitos graus de liberdade para produzir o software, ou podem ter a cultura na qual se sentem mais confortáveis trabalhando com regras, procedimentos e políticas bem definidas e estabelecidas.

Os cinco fatores críticos formam um perfil do desenvolvimento e podem ser apresentados graficamente sob a forma de um gráfico polar como é mostrado na Figura 2.3, onde cada fator é representado por uma dimensão do gráfico. A partir da análise das características do contexto de desenvolvimento e do problema a ser resolvido, pode-se delimitar uma área do gráfico que represente a região de análise para a determinação da proporção entre MA e MDP na customização de um processo híbrido.



- **Prejuízo 1:** prejuízo maior, capaz de provocar impacto negativo no projeto;
- **Prejuízo 2 e Conforto:** prejuízo menor, com pouco impacto financeiro e nenhum risco à vida humana;
- **Única Vida e Muitas Vidas:** se referem aos riscos à vida humana, caso o software resultante apresente erros.

Figura 2.3 – Gráfico polar com cinco fatores críticos (adaptado: Boehm and Turner 2004a)

Essa área resultante no gráfico pode delimitar uma porção mais interna do gráfico ou uma porção que se aproxime da área total do gráfico. No primeiro caso, tem-se uma área que revela um perfil onde os riscos são amenizados com adoção de MA. No segundo caso, tem-se uma área que revela um perfil onde os riscos são amenizados com adoção de MDP. As áreas resultantes da análise podem apresentar inúmeros formatos. Vale ressaltar que quanto mais interna ao gráfico for essa área, maior é a indicação de MA para o desenvolvimento analisado. Quanto mais periférica ao gráfico (isto é, quanto mais próxima da área total do gráfico) for a área resultante da análise, maior é a indicação de MDP para o desenvolvimento.

2.3.2 Discriminantes para Customização de Processo

Segundo (Royce, 1998), na customização de processos gerenciais de domínio específico ou projetos existem duas dimensões de fatores discriminantes: Complexidade Técnica e Complexidade Gerencial. A Figura 2.4 a seguir ilustra essas duas dimensões de variação de processos e apresenta alguns exemplos de categorias de projetos que seriam classificados nos quadrantes.

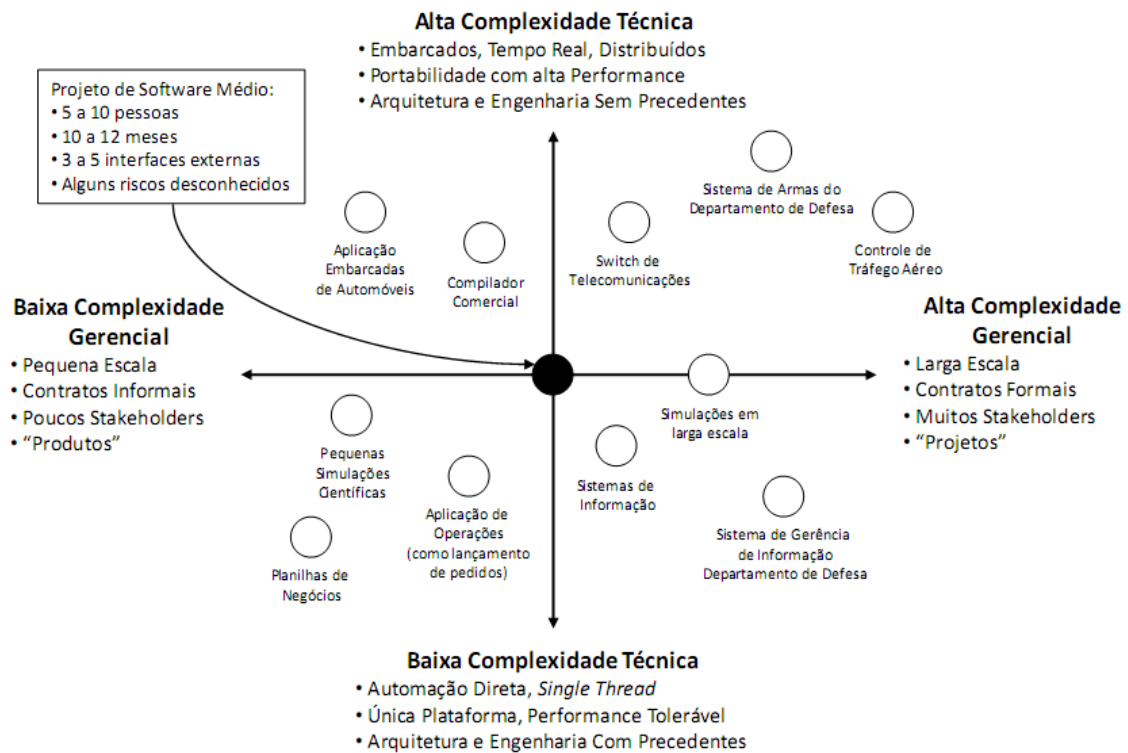


Figura 2.4 – Duas dimensões primárias de variação de processo (adaptado: Royce 1998)

Por exemplo, um sistema de controle de tráfego aéreo é considerado crítico, com grande complexidade gerencial e grande complexidade técnica, e por isso está classificado no primeiro quadrante. Já os sistemas comerciais simples, como planilhas eletrônicas simples, são classificadas no terceiro quadrante com complexidade gerencial baixa e complexidade técnica baixa.

Para (Royce, 1998) as maiores variações em processos são determinadas por seis parâmetros, que são algumas das dimensões críticas que um gerente de projeto deve considerar para customizar um framework de processo a fim de criar uma instância de implementação do processo. São eles:

1. Escala: tamanho dos projetos, medido através de quantidade de linhas de código, quantidade de pontos de função, quantidade de casos de uso, tamanho da equipe e mesmo valor financeiro.
2. Coesão dos Stakeholders: nível de coesão entre os envolvidos refere-se ao alinhamento dos objetivos. Stakeholders coesos têm objetivos em comum, habilidades complementares e facilidade de comunicação.
3. Nível de Flexibilidade ou Rigor: grau de rigor, formalidade e liberdade de mudanças nos contratos, processo de desenvolvimento, orçamento, qualidade ou qualquer conjunto de característica do projeto.
4. Maturidade do Processo: maturidade em processos das organizações, com alto nível de experiência precedente em desenvolvimento de software que permite previsibilidade no planejamento e execução dos projetos.
5. Risco Arquitetural: grau de visibilidade técnica apresentada pelos projetos desenvolvidos antes da produção em larga escala.
6. Experiência no Domínio: nível de experiência da empresa de desenvolvimento sobre o domínio dos projetos, convergindo em arquiteturas aceitáveis com um número mínimo de iterações.

A formalidade com que são realizadas revisões, o controle de qualidade dos artefatos gerados, a definição de prioridades, e vários outros parâmetros avaliados para instanciar processos são definidos pelo posicionamento do projeto em um dos quadrantes de complexidade técnica ou gerencial apresentados na Figura 2.5.

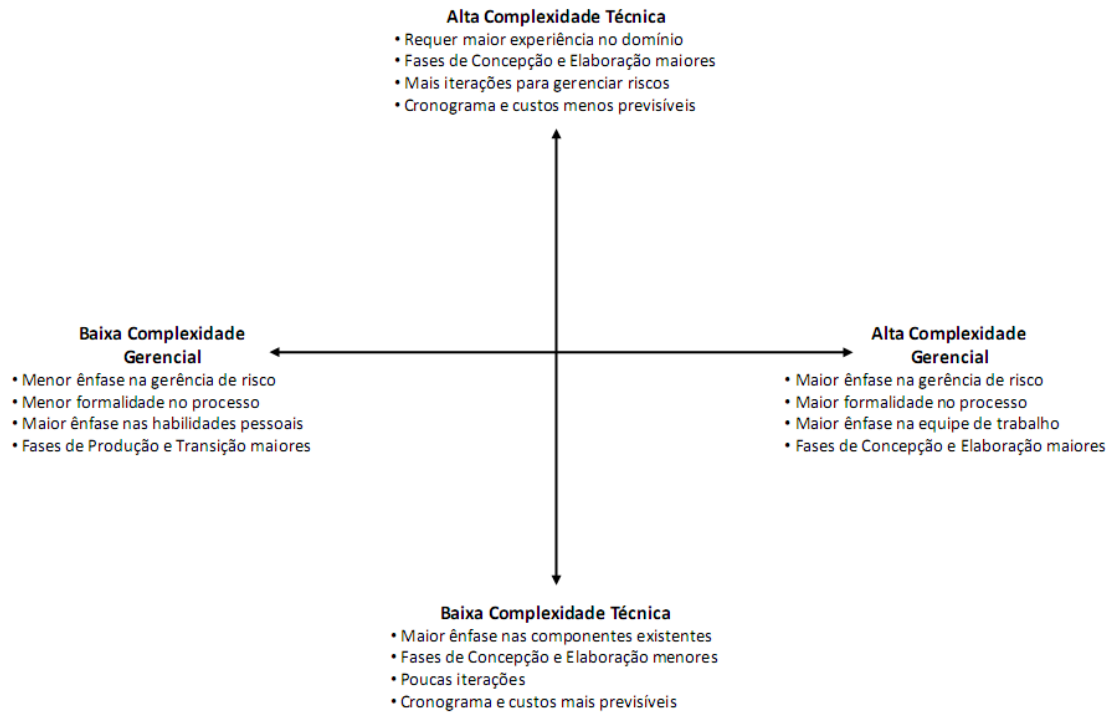


Figura 2.5 – Prioridades do framework de customização de processos (adaptado: Royce 1998)

Segundo a Figura 2.5, projetos com alta complexidade gerencial devem ser abordados com ênfase na gerência de riscos, maior formalidade no processo, maior ênfase na equipe de trabalho e fases iniciais (concepção e elaboração) maiores, e projetos com baixa complexidade gerencial devem ser abordados com menos ênfase em gerência de riscos, menor formalidade no processo, maior ênfase nas habilidades pessoais e fases de produção e transição maiores. Além disso, projetos com complexidade técnica alta requerem maior experiência no domínio, fases de concepção e elaboração maiores, mais iterações para gerência de riscos e cronograma e custos menos previsíveis, já os projetos com baixa complexidade técnica podem ser abordados com maior ênfase nos componentes existentes, fases de concepção e elaboração menores, poucas iterações e cronogramas e custos mais previsíveis.

2.4 Goal Question Metric (GQM)

A qualidade de um produto de software é o resultado das atividades realizadas no processo de desenvolvimento do mesmo. É importante que o processo de desenvolvimento seja melhorado continuamente e para isso, são necessárias métricas que permitam avaliá-lo e controlá-lo.

A abordagem GQM consiste em um mecanismo para a definição e avaliação de

métricas. Isto é feito através da avaliação de um conjunto de objetivos operacionais (Basili, 1992). Trata-se de uma abordagem sistemática para criar e integrar objetivos com modelos de processos de software, produtos e perspectivas de qualidade, baseadas nas necessidades de um projeto ou organização (Basili, 1992).

Essa abordagem parte do princípio de que para se realizar medições de forma eficiente é necessário, primeiramente, especificar cada um dos objetivos a serem alcançados. A partir da definição desses objetivos, são formuladas questões que definam cada um desses objetivos. Para finalizar, são especificadas quais métricas devem ser coletadas para responder às questões formuladas (Basili et al., 1994).

O resultado da aplicação da abordagem GQM é um sistema de medição que se concentra em um conjunto determinado de características e um conjunto de regras para a interpretação dos dados medidos. Este sistema apresenta três níveis (Basili et al., 1994; Berander and Jönsson, 2006):

- Nível conceitual (Objetivo): um objetivo é definido para um objeto, onde um objeto pode ser um produto, um processo ou um recurso. Este objetivo pode ser definido com base em vários modelos de qualidade, vários pontos de vista e com base em um ambiente em particular.
- Nível operacional (Questão): um conjunto de questões é utilizado para caracterizar a realização de um objetivo. As questões devem caracterizar o objeto a ser avaliado de acordo com algum critério de qualidade.
- Nível quantitativo (Métrica): Um conjunto de dados é associado com cada questão para respondê-la de forma quantitativa. Esses dados podem ser:
 - Objetivos: dependem apenas do objeto a ser medido.
 - Subjetivos: dependem do objeto a ser medido e do ponto de vista sob o qual foram extraídos.

Como pode ser observado na Figura 2.6, uma mesma métrica pode ser utilizada para responder mais de uma questão. As métricas são definidas seguindo-se uma perspectiva top-down (Solingen et al., 2002; Berander and Jönsson, 2006) e a interpretação dessas métricas é realizada sob uma perspectiva bottom-up (Solingen et al., 2002). A abordagem GQM pode ser considerada um padrão para a definição de frameworks para medição (Solingen and Berghout, 1999) e apresenta grande adaptabilidade a diversos tipos de ambientes e organizações o que pode ser comprovado pelo grande número de companhias que já utilizaram essa abordagem. Dentre elas podemos citar: Philips, Siemens, NASA (Solingen et al., 2002).

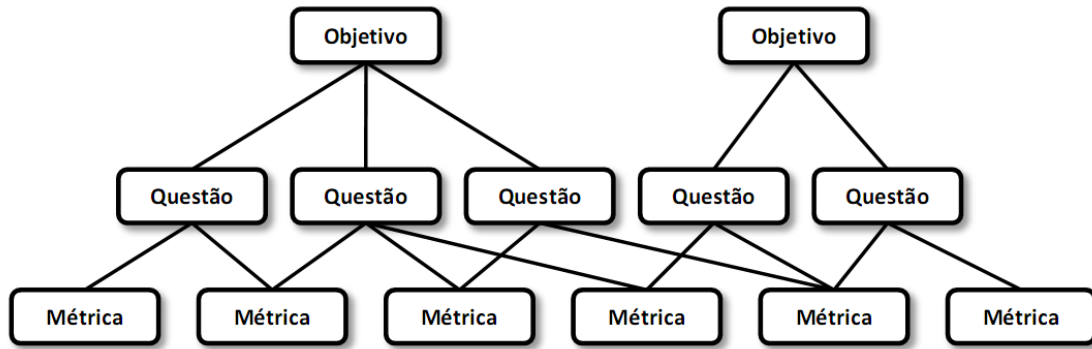


Figura 2.6 - Estrutura Hierárquica da Abordagem GQM (adaptado: Basili, 1994)

2.5 Modelos de Qualidade de Processo

A crescente demanda por sistemas computacionais no cotidiano de empresas e dos seres humanos tem direcionado organizações na busca por desenvolver seus trabalhos amparados em processos definidos e cíclicos de desenvolvimento de softwares. A busca por profissionalização desse desenvolvimento se dá basicamente por demandas cada vez mais complexas devido principalmente ao dinamismo do mercado onde atuam essas organizações.

Segundo (Sommerville, 2004), a ES é uma disciplina que envolve boas práticas de software, cujo foco é o desenvolvimento de sistemas de software de alta qualidade com eficiência relativa ao custo, para solucionar problemas cuja análise demonstrou a necessidade do uso de um sistema de software.

Com a exigência do mercado e a demanda crescente por softwares mais complexos e projetos simultâneos nas empresas, surgiram os modelos com a premissa de aumentar a qualidade dos processos organizacionais de desenvolvimento de software aplicáveis em vários níveis de competência:

CMM – *Capability Model Maturity* e SW-CMM - *Capability Maturity Model for Software* aplicável em nível organizacional, além da evolução do modelo para o CMMI - *Capability Maturity Model Integration* (SEI, 2002);

MPS.BR – Melhoria de Processo de Software Brasileiro – aplicável em nível organizacional (Softex, 2009);

SPICE – *Software Process Improvement and Capability Determination* (norma ISO/IEC 15504: *Information Technology, Software Process Assessment*) – aplicável em nível organizacional (ISO/IEC 15504, 2003).

2.5.1 CMM, SW-CMM e CMMI

O modelo *Capability Maturity Model* (ou Modelo de Capacidade e Maturidade) (CMM) foi produzido pelo *Software Engineering Institute* (SEI) da Universidade Carnegie Mellon (CMU), em Pittsburgh, EUA, por um grupo de profissionais de software, cuja 1ª versão foi lançada em ago/1991. A primeira versão do *Capability Maturity Model for Software* (SW-CMM), focava apenas na disciplina da engenharia de software. Depois do SW-CMM, outros CMM's foram desenvolvidos, focados em outras disciplinas, como engenharia de sistemas, subcontratação, pessoas e desenvolvimento integrado de produtos (SEI, 2002).

Segundo discriminado no documento guia do SW-CMM (SEI, 2002) ele é definido como um conjunto de regras (ou ações) que visa maximizar a garantia da qualidade de software através da adoção de três princípios básicos (Paulk, 1993):

1. Elaboração de um questionário com o intuito de classificar o nível de maturidade da empresa produtora ante o processo de produção;
2. Avaliação do processo de software, visando determinar o nível atual de desenvolvimento do mesmo;
3. Avaliação da capacidade do software, identificando fornecedores e práticas adequadas para sua melhor confecção.

Além disso, o SW-CMM descreve os elementos-chave da evolução de um processo de software imaturo para um processo maduro e disciplinado. Abrange práticas para planejamento, engenharia e gestão do desenvolvimento de software que, quando seguidas, melhoram a habilidade da organização em atender metas para custos, cronograma, funcionalidade e qualidade do produto.

No modelo, as definições básicas que permitem qualificar uma organização, conforme especificação do Institute of Electrical and Electronics Engineers (IEEE) (Paulk, 1993), podem ser vistas como:

1. Processo: É uma seqüência de passos cuja finalidade é atingir um objetivo próprio;
2. Processo de software: É o conjunto de atividades, métodos, ações e operações para produzir e manter os produtos de programação;
3. Capacidade do processo de software: Determina a gama de resultados esperados para o processo de software implantado;

4. Maturidade do processo de software: Caracteriza a aplicação definitiva dos processos necessários à capacitação plena.

Assim, pelas premissas mencionadas, o objetivo primordial do modelo é avaliar os processos de desenvolvimento de software utilizados pelas empresas, visando estabelecer projeções para os aspectos de qualidade, custos e prazos, abrangendo tanto o desenvolvimento de programas, quanto à manutenção dos mesmos, passando, também, pela sua aquisição (compra de pacotes prontos e respectivas manutenções) (SEI, 2002).

A estrutura do SW-CMM consiste dos cinco níveis de maturidade propostos inicialmente no modelo CMM e determina em qual dos níveis se encontram os processos de software das empresas, identificando pontos fortes e fracos e apontando caminhos para as melhorias necessárias. Como podem ser visualizados na Figura 2.7, os níveis propostos no CMM são: Inicial, Repetível, Definido, Gerenciado e Em otimização.

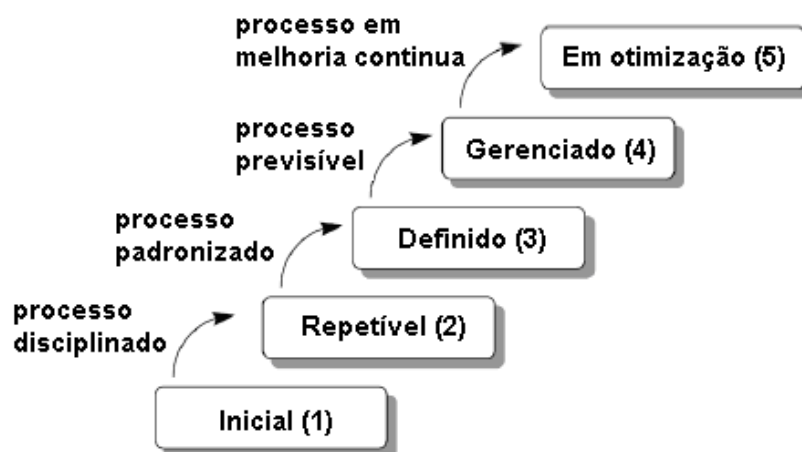


Figura 2.7 - Níveis propostos no CMM (Paulk et al., 1993).

Cada nível de maturidade do modelo é composto por um conjunto de práticas que devem ser efetivas e executadas para que cada nível atinja a maturidade desejada. Esse conjunto de práticas é conhecido por *Key Process Area* (KPAs) (Paulk, 1993).

Além do SW-CMM outras iniciativas de melhoria de processos organizacionais surgiram a partir do sucesso do modelo anterior (SEI, 2002).

- ✓ *Software Acquisition Capability Maturity Model* (SA-CMM): usado para avaliar a maturidade de uma organização em seus processos de seleção, compra e instalação de software desenvolvido por terceiros.
- ✓ *Systems Engineering Capability Maturity Model* (SE-CMM): avalia a maturidade da organização em seus processos de engenharia de sistemas,

concebidos como algo maior que o software. Um sistema inclui o hardware, o software e quaisquer outros elementos que participam do produto completo.

- ✓ *Integrated Product Development Capability Maturity Model (IPD-CMM)*: ainda mais abrangente que o SE-CMM, inclui também outros processos necessários à produção e suporte ao produto, tais como suporte ao usuário, processos de fabricação, entre outros.
- ✓ *People Capability Maturity Model (P-CMM)*: avalia a maturidade da organização em seus processos de administração de recursos humanos no que se refere ao software; recrutamento e seleção de desenvolvedores, treinamento e desenvolvimento, remuneração, entre outros.

Com base nos problemas gerados pela diversidade de modelos que surgiram, como por exemplo, a falta de padronização de termos, diferentes números de níveis e formas diferentes para avaliar o progresso de maturidade da empresa, surgiu o modelo conhecido como Capability Maturity Model Integration (CMMI) (SEI, 2002).

A proposta do CMMI é a de um modelo integrado que pode ser utilizado em várias disciplinas, como disciplinas de engenharia de sistemas, engenharia de software, desenvolvimento de integração de processos e produtos e subcontratação (fornecedor/aquisição). Com isto, observa-se que não se fala mais em software, como o SW-CMM, mas em produtos e serviços, abrangendo disciplinas distintas (SEI, 2002).

2.5.2 MPS.BR – Melhoria do Processo de Software Brasileiro

O MPS.BR atende à necessidade de implantar os princípios de Engenharia de Software de forma adequada ao contexto das empresas brasileiras, estando em consonância com as principais abordagens internacionais para definição, avaliação e melhoria de processos de software.

O MPS.BR tem como objetivo definir um modelo de melhoria e avaliação de processo de software, preferencialmente para as micro, pequenas e médias empresas, de forma a atender às suas necessidades de negócio e a ser reconhecido nacional e internacionalmente como um modelo aplicável à empresa de software. Este é o motivo pelo qual ele está aderente a modelos e normas internacionais. O MPS.BR também define regras para sua implementação e avaliação, dando sustentação e garantia de que o MPS.BR está sendo empregado de forma coerente com as suas definições (MPS.BR, 2009).

O foco principal do MPS.BR está nas pequenas e médias empresas brasileiras de software. Geralmente empresas com poucos recursos e que necessitam melhorar radicalmente seus processos de software em 1 ou 2 anos. Essas empresas precisam saber como adaptar à sua realidade para obter níveis correspondentes aos níveis de maturidade 2 e 3 de modelos para melhoria de processos de software como preconizado em CMMI-SE/SWSM (SEI, 2002) e a ISO/IEC 15504-5 (ISO/IEC 15504, 2003). Com isso, pretende-se que o modelo seja adequado ao perfil de empresas com diferentes tamanhos e características, públicas ou privadas, com especial atenção às micro, pequenas e médias empresas; e também seja compatível com os padrões de qualidade aceitos internacionalmente e que tenha como pressuposto o aproveitamento de toda a competência existente nos padrões e modelos de melhoria de processo já disponíveis. Nesse contexto, ele tem como base os requisitos de processos definidos nos modelos de melhoria de processo.

A base técnica utilizada para a construção do MPS.BR é composta pelas normas NBR ISO/IEC 12207 (Lin, 2003) – Processo de Ciclo de Vida de Software e suas emendas 1 e 2; e a ISO/IEC 15504 – Avaliação de Processo (também conhecida por SPICE: *Software Process Improvement and Capability Determination*) e seu Modelo de Avaliação de Processo de Software ISO/IEC 15504-5. O MPS.BR também cobre o conteúdo do CMMI-SE/SW, através da inclusão de processos e resultados de processos em relação aos processos da Norma NBR ISO/IEC 12207 (MPS.BR, 2009).

O MPS.BR possui três componentes:

- ✓ Modelo de Referência (MR-MPS);
- ✓ Método de Avaliação (MA-MPS);
- ✓ Modelo de Negócio (MN-MPS).

Cada componente do modelo MPS-BR está descrito através de guias e Documentos. A Figura 2.8 apresenta graficamente os componentes e seus documentos.

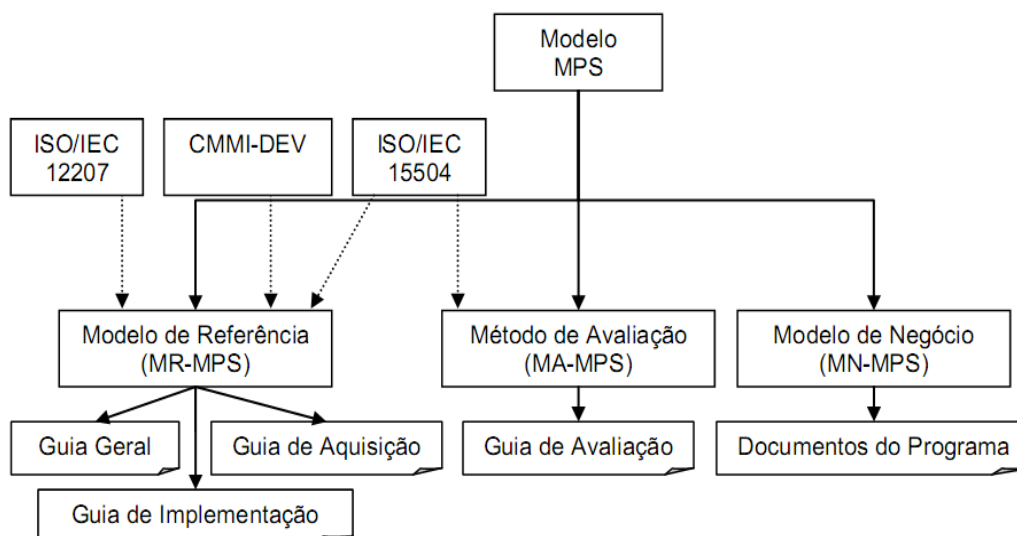


Figura 2.8 – Componentes do Modelo MPS.BR

O MPS.BR baseia-se nos conceitos de maturidade e capacidade de processo para a avaliação e melhoria da qualidade e produtividade de produtos de software e serviços.

O modelo é estruturado através de:

- **Níveis de Maturidade**

Os níveis de maturidade estabelecem patamares de evolução de processos, caracterizando estágios de melhoria da implementação de processos na organização. O nível de maturidade em que se encontra uma organização permite prever o seu desempenho futuro ao executar um ou mais processos. O MR-MPS define sete níveis de maturidade: A (Em Otimização), B (Gerenciado Quantitativamente), C (Definido), D (Largamente Definido), E (Parcialmente Definido), F (Gerenciado) e G (Parcialmente Gerenciado). A escala de maturidade se inicia no nível G e progride até o nível A (MPS.BR, 2009).

Para cada um destes sete níveis de maturidade é atribuído um perfil de processos que indicam onde a organização deve colocar o esforço de melhoria. O progresso e o alcance de um determinado nível de maturidade do MR-MPS (MPS.BR, 2009) é obtido quando são atendidos os propósitos e todos os resultados esperados dos respectivos processos e os resultados esperados dos atributos de processo estabelecidos para aquele nível.

- **Capacidade do processo**

A capacidade do processo é representada por um conjunto de atributos de processo descrito em termos de resultados esperados. A capacidade do processo expressa o grau de refinamento e institucionalização com que o processo é executado na

organização/unidade organizacional. No MR-MPS (MPS.BR, 2009), à medida que a organização/unidade organizacional evolui nos níveis de maturidade, um maior nível de capacidade para desempenhar o processo deve ser atingido.

O atendimento aos atributos do processo (AP), pelo atendimento aos resultados esperados dos atributos do processo (RAP), é requerido para todos os processos no nível correspondente ao nível de maturidade, embora eles não sejam detalhados dentro de cada processo. Os níveis são cumulativos, ou seja, se a organização está no nível F, esta possui o nível de capacidade do nível F que inclui os atributos de processo dos níveis G e F para todos os processos relacionados no nível de maturidade F (que também inclui os processos de nível G). Isto significa que, ao passar do nível G para o nível F, os processos do nível de maturidade G passam a ser executados no nível de capacidade correspondente ao nível F. Em outras palavras, na passagem para um nível de maturidade superior, os processos anteriormente implementados devem passar a ser executados no nível de capacidade exigido neste nível superior (MPS.BR, 2009).

- **Distribuição de processos por níveis**

A Tabela a seguir apresenta os níveis de maturidade do MR-MPS, os processos e os atributos de processo correspondentes a cada nível (MPS.BR, 2009).

Tabela 2.2 – Níveis de Maturidade e Processo MPS.BR

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

3 DIAGNÓSTICO DO PERFIL DE COMPLEXIDADE DO AMBIENTE

3.1 Características do Perfil

Para realizar o diagnóstico do **perfil de complexidade de ambiente** foram investigadas e selecionadas **sete características** do ambiente dos problemas de desenvolvimento de software. Tais características foram selecionadas combinando os cinco fatores críticos definidos por (Boehm and Turner, 2004a): Tamanho, Criticalidade, Dinamismo, Competência Pessoal e Cultura, discutidos na Seção 2.3.1., e os seis discriminantes de customização de processo definidos por (Royce, 1998): Escala, Coesão dos Stakeholders, Flexibilidade (Nível de Rigor Contratual), Maturidade em Processos, Previsibilidade Arquitetural e Experiência no Domínio, discutidos na Seção 2.3.2.

Foi realizado um relacionamento para estabelecer correspondências entre os parâmetros propostos pelos respectivos autores para avaliação e customização de processos, representados pela Figura 3.1.

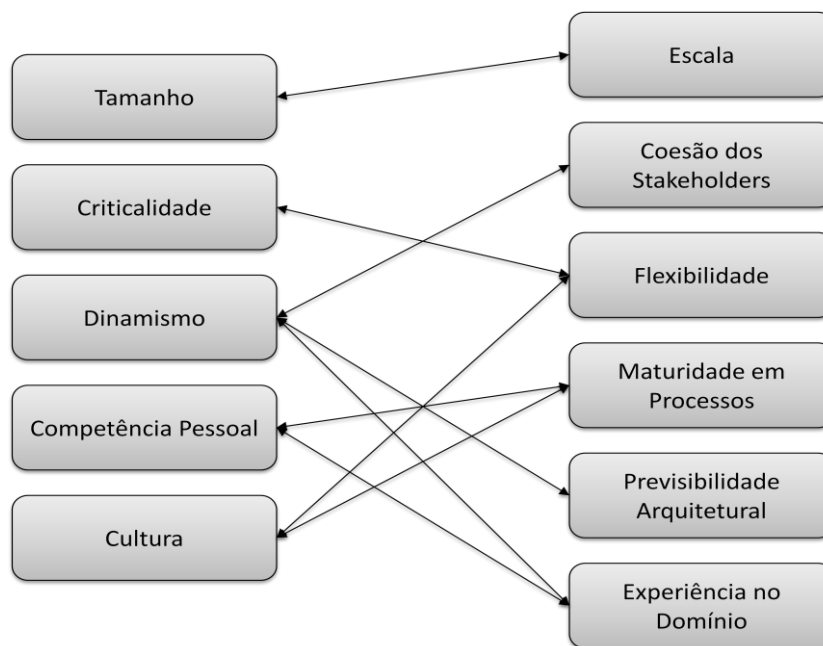


Figura 3.1 – Relacionamento entre os parâmetros de avaliação de processos.

Os fatores “Tamanho” e “Escala” são equivalentes, apesar das diversas maneiras de mensurar projetos de software. O fator “Criticalidade” está relacionado à “Flexibilidade” uma vez que para projetos críticos, por exemplo, envolvendo risco para vidas humanas, devem possuir maior rigor contratual e portanto menor flexibilidade. O

“Dinamismo” em projetos de software se deve a diversos fatores, entre eles: “Coesão dos stakeholders”, permitindo um levantamento de requisitos mais uniforme, “Previsibilidade Arquitetural”, sendo que quão mais previsível for a arquitetura do projeto ele tende a ter menos alterações e “Experiência no domínio”, permitindo o entendimento precoce do problema. A “Competência Pessoal” se relaciona, sobretudo, com “Maturidade em processos” do ponto de vista de competência organizacional e “Experiência no domínio” no sentido de competência técnica agregada ao profissional. E finalmente o fator “Cultura” que envolve tanto “Flexibilidade” quanto “Maturidade em processos”, principalmente no que diz respeito à predisposição da organização a trabalhar com rigor contratual e processos de desenvolvimento definidos.

Desse relacionamento foram determinados os fatores que serão considerados para diagnosticar o **perfil de complexidade do ambiente**. Tais fatores foram ainda classificados em **predominantemente gerencial** e **predominantemente técnico**, para que o perfil traçado possua distinção também entre complexidade técnica e gerencial.

O fator “Escala” foi considerado independente nessa classificação uma vez que o tamanho dos projetos dentro do contexto de uma MPE é muito variável e apesar de ser determinante na definição da complexidade do ambiente em geral, não permite relação direta entre complexidade técnica ou gerencial.

Os fatores selecionados através do relacionamento supracitado e suas respectivas classificações são apresentados a seguir:

1. **Escala** – referente ao tamanho dos projetos desenvolvidos.
2. **Dinamismo** – referente à quantidade de mudança de requisitos relacionados ao problema. (Gerencial)
3. **Criticalidade / Flexibilidade** – referente ao rigor contratual imposto pelos projetos e criticalidade envolvida nos projetos desenvolvidos. (Gerencial)
4. **Cultura / Maturidade em Processo** – referente à predisposição e habilidade da equipe para trabalhar com processos formais organizados. (Gerencial)
5. **Previsibilidade Arquitetural** – referente à previsibilidade técnica média do projeto em termos de desempenho, robustez e confiabilidade. (Técnico)
6. **Experiência no Domínio** – referente ao nível de experiência adquirida pela corporação no domínio do problema dos projetos. (Técnico)
7. **Competência Pessoal** – referente à competência pessoal dos integrantes da equipe. (Técnico)

O perfil de complexidade do ambiente será traçado no gráfico radial com sete eixos, análogo ao gráfico polar de cinco eixos para balanceamento de MA e MDP proposto por (Boehm and Turner, 2004a).

O gráfico radial com os sete eixos está representado na Figura 3.2 a seguir:

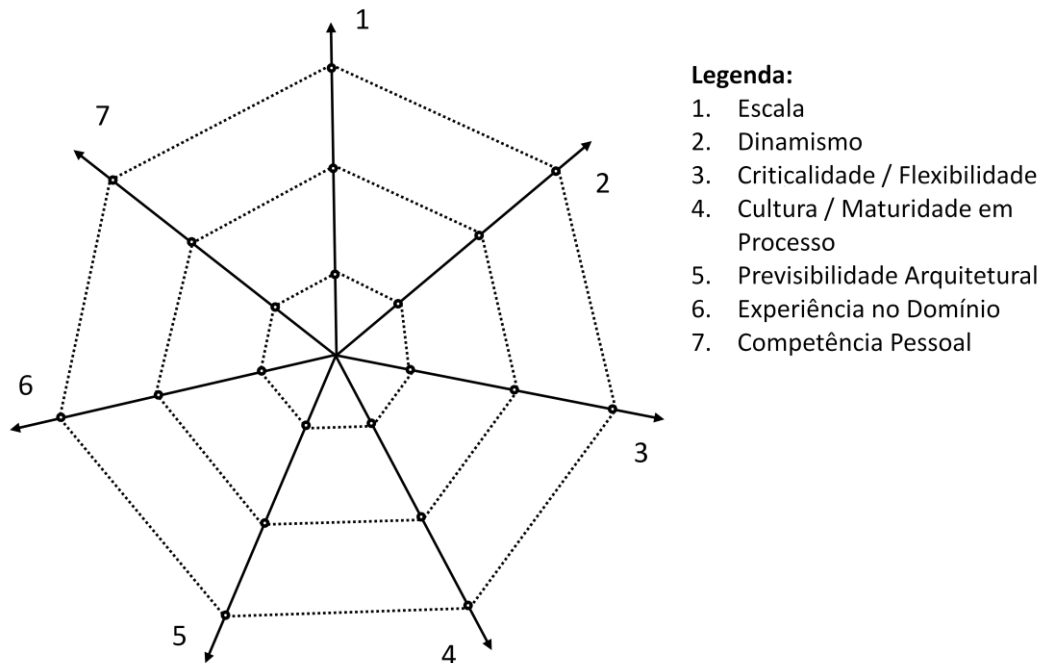


Figura 3.2 - Gráfico de sete eixos do perfil de complexidade do ambiente

Cada eixo do gráfico está representado com uma escala de três posições, uma mais interna (a), uma intermediária (b) e outra mais externa (c). Assim para traçar o perfil de complexidade do ambiente proposto, é necessário classificar cada característica em uma das três posições. A divisão de três posições foi utilizada para tornar o gráfico aderente ao método proposto apresentado no Capítulo 5 desse trabalho. Para isso foi utilizada a abordagem GQM detalhada na seção seguinte.

3.2 Aplicação da Abordagem GQM

3.2.1 Projeto da Estrutura do Questionário

Como foi descrito na Seção 2.4, o resultado da aplicação da abordagem GQM é um sistema de medição que se concentra em um conjunto determinado de características. Primeiramente, são especificados os objetivos a serem alcançados, depois são formuladas questões que definam cada um dos objetivos e, para finalizar, são especificadas quais métricas devem ser coletadas para responder às questões

formuladas.

A quantificação de cada um dos sete fatores de complexidade do ambiente, mostrados no gráfico da Figura 3.2, constitui um objetivo para a abordagem GQM neste trabalho. Para atingir esses objetivos, uma série de questões e métricas é formulada e essas métricas são usadas como guia para quantificar cada um dos fatores frente às escalas explicitadas no gráfico.

A definição das métricas esclarece quais são as informações específicas necessárias. O próximo passo é definir como tais informações serão coletadas. Uma opção é a aplicação de um questionário aos membros da equipe de desenvolvimento contendo perguntas que cubram todas as informações desejadas. O estabelecimento de métricas orienta não só a definição das informações relevantes, como também a elaboração do questionário. Na Seção 3.2.2 a atividade de elaboração do questionário é detalhada.

A seguir, é mostrada a aplicação da abordagem GQM na quantificação de cada um dos fatores. Tal abordagem está baseada no trabalho de (Soares, 2007), seguindo a mesma estrutura de tabelas com os campos: objetivo, característica, objeto, ponto de vista, além das questões e métricas para cada questão.

- **Escala**

Na Tabela 3.1 é mostrada a aplicação da abordagem GQM na quantificação do fator “Escala”. O objetivo é descobrir quais são as métricas necessárias para estimar o tamanho do projeto de desenvolvimento de software.

Tabela 3.1 – GQM na quantificação do fator Escala

| |
|---|
| Escala |
| <hr/> |
| Objetivo: Estimar |
| Característica: Escala/Tamanho dos projetos |
| Objeto: Histórico dos projetos |
| Ponto de vista: Gerente de processo ou projeto |
| <hr/> |
| Questão: Qual é o tamanho médio dos projetos? |
| <hr/> |
| Métrica (s): |
| ✓ Tamanho médio da equipe de desenvolvimento por projeto. |
| ✓ Número médio de projetos aos quais cada indivíduo se dedica ao mesmo tempo. |
| ✓ Tamanho médio dos projetos em pontos de função. |
| ✓ Quantidade média de casos de uso por projeto. |

- **Dinamismo**

Na Tabela 3.2 é mostrada a aplicação da abordagem GQM na quantificação do fator “Dinamismo” como foi proposto em (Basili et al., 1994). O objetivo é descobrir quais são as métricas necessárias para estimar a periodicidade das alterações sofridas pelos requisitos de software em função de alterações sofridas pelo contexto do problema, sob a visão de um gerente de processo.

Tabela 3.2 – GQM na quantificação do fator Dinamismo

| |
|---|
| <p>Dinamismo</p> <hr/> <p>Objetivo: Estimar Característica: Periodicidade das alterações em requisitos de software em função de características do ambiente e contexto do problema. Objeto: Requisitos do software Ponto de vista: Gerente de processo ou projeto</p> <hr/> <p>Questão: Qual é a principal causa para mudanças nos requisitos do sistema?</p> <hr/> <p>Métrica (s):</p> <ul style="list-style-type: none"> ✓ % de mudanças de requisitos ocorridas em função de alterações sofridas pelo contexto do problema a ser resolvido. ✓ % de mudanças de requisitos ocorridas em função da visualização de partes prontas do sistema (prototipação). ✓ % de mudanças de requisitos ocorridas em função do levantamento incorreto de requisitos. <hr/> <p>Questão: Com que periodicidade ocorre mudança nos requisitos para os casos em que esta mudança é proveniente de alterações sofridas pelo contexto do problema a ser resolvido?</p> <hr/> <p>Métrica (s):</p> <ul style="list-style-type: none"> ✓ % de mudança de requisitos por mês (do total estabelecido até o momento). <hr/> <p>Questão: Qual é a coesão dos Stakeholders do projeto?</p> <hr/> <p>Métrica (s):</p> <ul style="list-style-type: none"> ✓ Quantidade média de Stakeholders nos projetos? ✓ Quantidade de objetivos diferentes apontados pelos Stakeholders em cada projeto? |
|---|

- **Criticalidade / Flexibilidade**

Na Tabela 3.3 é mostrada a aplicação da abordagem GQM na quantificação do fator “Criticalidade / Flexibilidade” como foi proposto em (Basili et al., 1994). O objetivo é descobrir quais são as métricas necessárias para estimar a gravidade de

perdas ocasionadas em função de falhas presentes no software desenvolvido. A gravidade dessas perdas pode variar desde risco à vida humana até ausência de prejuízo de qualquer tipo.

Tabela 3.3 – GQM na quantificação do fator Criticalidade / Flexibilidade

| Criticalidade / Flexibilidade |
|---|
| <p>Objetivo: Estimar</p> <p>Característica: Rigor contratual devido à criticalidade estabelecida para os projetos.</p> <p>Objeto: Problema a ser resolvido pelo desenvolvimento de software.</p> <p>Ponto de vista: Gerente de processo ou projeto</p> |
| <p>Questão: Qual é a criticalidade dos problemas resolvidos pelo desenvolvimento de software?</p> |
| <p>Métrica (s):</p> <ul style="list-style-type: none"> ✓ % de problemas onde falhas no sistema implicam em perda de várias vidas. ✓ % de problemas onde falhas no sistema implicam em perda de uma vida. ✓ % de problemas onde falhas no sistema implicam em prejuízos financeiros diretos e indiretos. ✓ % de problemas onde falhas no sistema implicam em prejuízos financeiros diretos. |

- **Cultura / Maturidade em Processo**

Na Tabela 3.4 é mostrada a aplicação da abordagem GQM na quantificação do fator “Cultura / Maturidade em Processo”. O objetivo é descobrir quais são as métricas necessárias para estimar o quanto a equipe está acostumada a desenvolver software de forma organizada e estruturada, sob a visão de um gerente de processo.

Tabela 3.4 – GQM na quantificação do fator Cultura / Maturidade em Processo

| Cultura / Maturidade em Processo |
|---|
| <p>Objetivo: Estimar</p> <p>Característica: Hábito/Predisposição de trabalhar de forma organizada.</p> <p>Objeto: Equipe de Desenvolvimento</p> <p>Ponto de vista: Gerente de processo ou projeto</p> |
| <p>Questão: O ciclo de desenvolvimento de software inclui quais das seguintes fases: análise, especificação, projeto, codificação e teste?</p> |

Métrica (s):

- ✓ % dos projetos que incluem a fase de análise.
- ✓ % dos projetos que incluem a fase de especificação.
- ✓ % dos projetos que incluem a fase de projeto.
- ✓ % dos projetos que incluem a fase de codificação.
- ✓ % dos projetos que incluem a fase teste.
- ✓ % dos projetos que incluem todas estas fases.

Questão: Que tipo de técnicas e procedimentos são empregados durante o desenvolvimento de software visando sua estruturação / organização?

Métrica (s):

- ✓ % dos projetos que possuem documentação na forma de diagramas e/ou modelos.
- ✓ % dos projetos onde os desenvolvedores seguem algum padrão para a codificação.
- ✓ % dos projetos que documentam a comunicação, as decisões da equipe e as tarefas de cada membro através de documentos e/ou ferramentas.
- ✓ % dos projetos onde os desenvolvedores seguem algum padrão para a documentação sobre o desenvolvimento do projeto.
- ✓ % dos projetos que utilizam algum tipo de processo definido.
- ✓ % dos projetos onde os desenvolvedores utilizam todos os recursos citados pelas métricas anteriores.

Questão: O nível de documentação adotado atualmente é satisfatório?

Métrica (s):

- ✓ Avaliação subjetiva do gerente de projetos.

• Previsibilidade Arquitetural

Na Tabela 3.5 é mostrada a aplicação da abordagem GQM na quantificação do fator “Previsibilidade Arquitetural” descrito em (Royce, 1998). O objetivo é descobrir quais são as métricas necessárias para estimar o nível de previsibilidade técnica apresentado no projeto de desenvolvimento de software.

Tabela 3.5 – GQM na quantificação do fator Previsibilidade Arquitetural

Previsibilidade Arquitetural

Objetivo: Estimar

Característica: Nível de previsibilidade técnica apresentada pelo projeto.

Objeto: Problema a ser resolvido pelo desenvolvimento de software

Ponto de vista: Gerente de processo ou projeto

Questão: Qual o nível de previsibilidade técnica média apresentada pelos projetos em termos de desempenho?

Métrica (s):

- ✓ % dos projetos com conhecimento prévio em termos de utilização de recursos computacionais.
- ✓ % dos projetos com conhecimento prévio em termos de tempo de resposta.
- ✓ % dos projetos com conhecimento prévio em termos de taxa de transmissão de dados.

Questão: Qual o nível de previsibilidade técnica média apresentada pelos projetos em termos de adaptabilidade a mudança?

Métrica (s):

- ✓ % dos projetos com conhecimento prévio em termos de impacto de adição de novas funcionalidades.
- ✓ % dos projetos com conhecimento prévio em termos de impacto de incorporação de novas tecnologias.

Questão: Qual o nível de previsibilidade técnica média apresentada pelos projetos em termos de confiabilidade?

Métrica (s):

- ✓ % dos projetos com conhecimento prévio em termos de grau de tolerância à falhas.

- **Experiência no Domínio**

Na Tabela 3.6 é mostrada a aplicação da abordagem GQM na quantificação do fator “Experiência no Domínio” descrito em (Royce, 1998). O objetivo é descobrir quais são as métricas necessárias para estimar a experiência no domínio apresentado no projeto de desenvolvimento de software.

Tabela 3.6 – GQM na quantificação do fator Experiência no Domínio

Experiência no Domínio

Objetivo: Estimar

Característica: Nível de experiência no domínio dos projetos desenvolvidos.

Objeto: Problema a ser resolvido pelo desenvolvimento de software

Ponto de vista: Gerente de processo ou projeto

Questão: Qual é o grau de conhecimento corporativo no domínio dos projetos?

Métrica (s):

- ✓ Quantidade de áreas de atuação da empresa.
- ✓ Quantidade média de projetos desenvolvidos na mesma área de conhecimento.
- ✓ Qual a frequência de projetos desenvolvidos na mesma área de conhecimento.

Questão: Qual é o grau de conhecimento técnico da equipe no domínio dos projetos?

Métrica (s):

- ✓ Quantidade média de projetos em uma mesma área de conhecimento de que os membros da equipe participam.
- ✓ Quantidade média de profissionais com conhecimento específico da área de conhecimento dos projetos.

- **Competência Pessoal**

Na Tabela 3.7 é mostrada a aplicação da abordagem GQM na quantificação do fator “Competência Pessoal” descrito em (Boehm and Turner, 2004a). O objetivo é descobrir quais são as métricas necessárias para estimar a competência pessoal dos integrantes da equipe, isto é, o nível de habilidade de cada indivíduo de acordo com os níveis 1, 2 e 3, estabelecidos por (Cockburn, 2002).

Tabela 3.7 – GQM na quantificação do fator Competência Pessoal

| Competência Pessoal |
|---|
| <p>Objetivo: Estimar Característica: Competência pessoal dos integrantes da equipe Objeto: Equipe de desenvolvimento Ponto de vista: Gerente de processo ou projeto</p> |
| <p>Questão: Qual é a porcentagem de profissionais pertencentes à equipe de desenvolvimento que possuem outra área de atuação?</p> |
| <p>Métrica (s):</p> <ul style="list-style-type: none"> ✓ % de membros da equipe de desenvolvimento cuja área de atuação principal não é a área da computação. |
| <p>Questão: Qual é a divisão do número de membros da equipe entre as diversas atividades de desenvolvimento do software?</p> |
| <p>Métrica (s):</p> <ul style="list-style-type: none"> ✓ % de membros da equipe de desenvolvimento que desempenham atividades na fase de análise e especificação de requisitos. ✓ % de membros da equipe de desenvolvimento que desempenham atividades na fase de projeto. ✓ % de membros da equipe de desenvolvimento que desempenham atividades na fase de codificação. ✓ % de membros da equipe de desenvolvimento que desempenham atividades na fase de teste. ✓ % de membros que desempenham atividades em mais de uma fase do ciclo de vida do software. ✓ % de membros que desempenham atividades em todas as fases do ciclo de vida do software. ✓ % de membros da equipe que desempenham atividades gerenciais exclusivamente. |

Questão: Qual é o nível de experiência dos membros da equipe?

Métrica (s):

- ✓ Tempo médio de experiência dos indivíduos cuja área de atuação principal é a área de computação e que desempenham atividades como análise, especificação de requisitos, projeto, codificação ou teste.
- ✓ Tempo médio de experiência dos indivíduos cuja área de atuação principal é a área de computação e que desempenham atividades gerenciais.
- ✓ Tempo médio de experiência dos indivíduos cuja área de atuação principal é a área de computação e que desempenham atividades como análise, especificação de requisitos, projeto, codificação ou teste; e gerência.
- ✓ Tempo médio de experiência dos indivíduos cuja área de atuação principal não é a área de computação e que desempenham atividades como análise, especificação de requisitos, projeto, codificação ou teste.
- ✓ Tempo médio de experiência dos indivíduos cuja área de atuação principal não é a área de computação e que desempenham atividades gerenciais.
- ✓ Tempo médio de experiência dos indivíduos cuja a área de atuação principal não é a área de computação e que desempenham atividades como análise, especificação de requisitos, projeto, codificação ou teste; e gerência.

3.2.2 Elaboração do Questionário

O questionário destina-se a todos os membros da equipe e visa à obtenção de dados sobre o contexto de desenvolvimento, a equipe e seus membros. As informações coletadas referem-se às características gerais da equipe e às características individuais de cada membro da equipe. Assim são definidos dois tipos de questionários (Soares, 2007):

- ✓ Questionário Gerencial: visa a extrair informações relativas à equipe. Deve ser respondido por um profissional que compreenda a realidade da equipe e do desenvolvimento realizado, podendo ser um gerente ou profissional de nível de responsabilidade equivalente.
- ✓ Questionário Individual: visa extrair informações relativas aos indivíduos pertencentes à equipe. Deve ser respondido por cada um dos membros da equipe independente da sua função.

Os questionários elaborados estão anexos no Apêndice A.

Na elaboração do questionário, as recomendações a seguir foram consideradas (Firlej and Hellens, 1991 Apud Soares, 2007):

- ✓ O número de perguntas presentes em cada questionário deve ser o menor possível. Um número alto de perguntas pode promover a fadiga do respondente comprometendo assim a veracidade e exatidão das respostas.

- ✓ A diagramação do questionário deve ser realizada visando a redução do espaço total ocupado. O questionário não deve parecer maior do que realmente é.
- ✓ Cada questionário deve apresentar um cabeçalho contendo informações sobre o objetivo da pesquisa e agradecimentos ao respondente pelo tempo dedicado a responder as perguntas. Isso contribui para esclarecer e motivar o respondente.
- ✓ O enunciado de cada pergunta deve deixar claro o número permitido de respostas a serem marcadas.
- ✓ No caso de perguntas que não se adéquem a respostas enumeradas pode ser empregada uma escala onde a marcação de um ponto nessa escala é a resposta (trata-se de um artifício utilizado para obter uma melhor aproximação da resposta).

3.3 Análise e Interpretação do Questionário

As informações referentes a cada um dos fatores de complexidade do ambiente podem ser obtidas através de uma ou mais perguntas presentes nos questionários. Isto acontece porque nem todas as informações necessárias podem ser trivialmente fornecidas pelos respondentes; e assim elas podem ser obtidas através de outras perguntas indiretas. Desta forma, após a aplicação do questionário, é necessário reunir todas as informações sobre cada fator, agrupá-las e organizá-las para que estejam de acordo com as escalas do gráfico da Figura 3.2.

A atividade de agrupar e organizar as informações obtidas deve ser realizada de maneira explícita e repetível para que qualquer equipe seja analisada de forma padronizada. Assim, faz-se necessária a formalização dos passos a serem seguidos e dos aspectos a serem considerados durante essa atividade (Soares, 2007).

Nas seções seguintes, o agrupamento e a organização das informações para cada fator de complexidade são detalhados.

- **Escala**

O fator “Escala” é quantificado através das perguntas 1, 3 e 4 do questionário gerencial mostrado no Apêndice A. As opções de respostas para as perguntas são intervalos de valores que já estão de acordo com a escala do eixo escala do gráfico.

A partir das respostas obtidas, é possível categorizar o projeto de acordo com tamanho da equipe, quantidade de Casos de Uso e tamanho em PF.

Este trabalho estabelece a seguinte classificação:

Tabela 3.8 – Classificação do fator Escala

| Posição | Classificação | Descrição |
|---------|------------------------|--|
| a | Projeto Pequeno | Até 3 profissionais, até 20 casos de uso, até 200 pontos de função |
| b | Projeto Médio | Até 10 profissionais, até 50 casos de uso, até 700 pontos de função |
| c | Projeto Grande | Acima de 10 profissionais, acima 50 casos de uso, acima 700 pontos de função |

Essa classificação baseia-se principalmente no alcance dos processos ágeis que se destinam a equipes pequenas sendo que são consideradas como pequenas, equipes com tamanho médio de nove pessoas (Cockburn and Highsmith, 2001). Os adeptos de métodos ágeis defendem que a dificuldade de se obter sucesso na aplicação de um processo ágil se torna maior em equipes com mais de 15 a 20 pessoas (Boehm, 2002; Constantine, 2001).

A principal medida de tamanho de projetos de software é através da quantidade de membros na equipe, contudo, em MPEs essa relação não é direta, visto que projetos maiores e/ou de maior relevância estratégica para empresa são realizados por equipes mais qualificadas e não necessariamente por mais membros. Daí a extensão do fator escala para levar em consideração além da quantidade de profissionais, a quantidade casos de uso e a quantidade de pontos de função. Os valores contidos na Tabela 3.8 para casos de uso e pontos de função foram obtidos por (Tavares, 2004), (Belgamo, 2004) e experiência prática do autor do trabalho.

A classificação final para Escala do projeto se dará através das respostas das perguntas 1, 3 e 4, verificando os intervalos correspondentes às classificações, sendo critério de desempate a classificação cumulativa pela ordem: Tamanho da Equipe, Quantidade de Pontos de Função e Quantidade de Casos de Uso.

- **Dinamismo**

A extração de informações para quantificar o fator “Dinamismo” é feita através das perguntas 5, 6 e 7 do questionário gerencial mostrado no apêndice A. As respostas a essas perguntas não estão de acordo com a escala do gráfico da Figura 3.2. É necessário associar e mapear as respostas a estas três perguntas para gerar um valor que represente o percentual de mudanças de requisitos / mês, e que possa ser representado como um intervalo de valores no eixo dinamismo dentro do gráfico. Esse mapeamento é realizado através do seguinte cálculo (Soares, 2007):

$$\% \text{ de Mudança De Requisitos} = \frac{\text{Resposta Pergunta 5A}}{100} \times \text{Resposta Pergunta 7} \times \text{Frequência}$$

Onde Frequência é:

- ✓ 0: se a resposta à pergunta 6 é nunca
- ✓ 30: se a resposta à pergunta 6 é diariamente
- ✓ 4,29: se a resposta à pergunta 6 é semanalmente
- ✓ 1,00: se a resposta à pergunta 6 é mensalmente
- ✓ 0,50: se a resposta à pergunta 6 é bimestralmente
- ✓ 0,33: se a resposta à pergunta 6 é trimestralmente
- ✓ 0,17: se a resposta à pergunta 6 é semestralmente
- ✓ 0,08: se a resposta à pergunta 6 é anualmente

A equação acima produz um valor para a % de mudanças de requisitos em função de mudanças no contexto do problema. Esse valor é obtido a partir das respostas às perguntas 5A e 7 do questionário gerencial mostrado na Seção 3.2.3, e do valor Frequência. O valor Frequência é obtido em função da resposta à pergunta 6 que determina a periodicidade com que ocorrem mudanças de requisitos durante o desenvolvimento de software. Como o resultado final deve estar em função da % de mudanças por mês, se a resposta à pergunta 6 for diariamente, então o valor para Frequência deve ser: 30. Se a resposta for, por exemplo, bimestralmente, então o valor para Frequência deve ser: 0,50. O mesmo raciocínio aplica-se às outras respostas.

Como as opções de resposta da pergunta 7 são intervalos, então a equação acima é utilizada para os valores extremos do intervalo referente à opção de resposta selecionada. Assim é obtido um intervalo de valores para a % de mudanças de requisitos/ mês, em função de mudanças no contexto do problema (Soares, 2007).

Com a definição desse intervalo, pode-se fazer uma aproximação da situação do ambiente para o eixo dinamismo do gráfico da Figura 3.2. Além disso, pode-se categorizar o contexto dos problemas resolvidos pela equipe de acordo com o dinamismo apresentado.

Este trabalho estabelece a seguinte classificação:

Tabela 3.9 - Classificação do fator Dinamismo

| Posição | Classificação | Descrição |
|---------|--|------------------------------------|
| a | Projetos pouco dinâmicos | Até 10% de alterações por mês |
| b | Projetos com dinamismo moderado | Até 15% de alterações por mês |
| c | Projetos muito dinâmicos | Acima de 15% de alterações por mês |

Essa classificação se baseia no que é citado por Barry Boehm sobre taxas de mudanças de requisitos por mês entre 10% a 30% para aplicações altamente competitivas e que mudam rapidamente (Agerfalk and Fitzgerald, 2006).

- **Criticalidade / Flexibilidade**

O fator “Criticalidade / Flexibilidade” é quantificado através da pergunta 14 do questionário gerencial mostrado no Apêndice A. Considerando-se a dificuldade por parte do respondente em fornecer uma definição exata de qual é a gravidade das perdas ocasionadas em função de erros presentes no sistema desenvolvido, é oferecida uma opção alternativa de resposta. Ao invés de opções enumeradas, é oferecida ao respondente uma escala nominal (Soares, 2007).

Uma escala caracteriza as variáveis cujo levantamento pretende-se fazer, e em articular a forma como se ordenam as nossas observações. Especificamente, uma escala ordinal é aquela utilizada para avaliar um fenômeno em termos de onde ele se situa dentro de um conjunto de patamares ordenados, variando desde um patamar mínimo até um máximo. Geralmente, designam-se os valores de uma escala ordinal em termos de numerais, ranking ou rótulos (de Andrade Martins, 2005 Apud Soares, 2007).

O enunciado da pergunta 14 contextualiza o que é criticalidade e oferece uma escala orientada a partir do extremo que representa menor gravidade (ausência de prejuízo) para o extremo que representa maior gravidade (risco a vidas humanas).

Assim o respondente pode marcar um ponto nessa escala que reflita a realidade do seu contexto de desenvolvimento; e o ponto marcado é mapeado para uma resposta que esteja de acordo com a escala do eixo Criticalidade / Flexibilidade do gráfico.

A escala oferecida tem 9 centímetros. Para realizar o mapeamento, ela é dividida em intervalos de 3 centímetros, onde cada um dos intervalos corresponde a cada uma das situações enumeradas no eixo Criticalidade / Flexibilidade do gráfico da Figura 3.2. De acordo com o ponto marcado, é identificado o intervalo e a situação correspondente.

Tabela 3.10 – Classificação do fator Criticalidade / Flexibilidade

| Posição | Classificação | Descrição |
|---------|--|---|
| a | Projetos com baixa criticalidade e alta flexibilidade | Apesar de gerar transtorno não gera prejuízo de nenhum tipo |
| b | Projetos criticalidade e flexibilidade moderada | Risco de prejuízo financeiro e rigor contratual moderado |
| c | Projetos com alta criticalidade e baixa flexibilidade | Risco para vida humana e muito rigor contratual |

O mapeamento entre os intervalos e as possíveis situações de perdas é mostrado na Figura 3.3:

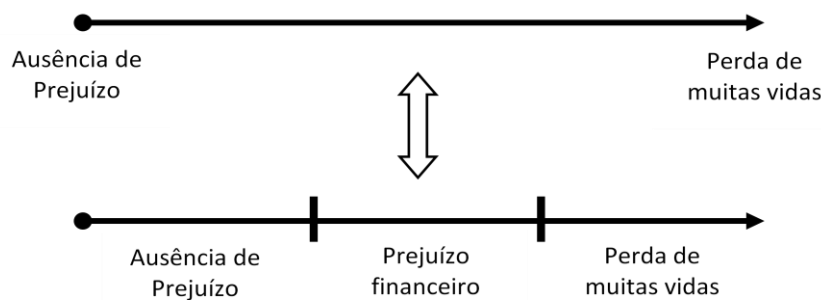


Figura 3.3 – Mapeamento dos intervalos da escala de perdas

- **Cultura / Maturidade em Processo**

Para quantificar o fator “Cultura / Maturidade em Processo”, é necessário analisar o contexto de desenvolvimento da equipe para saber que tipo de métodos e técnicas são utilizadas na estruturação e organização do desenvolvimento. Assim, é possível saber sobre a cultura da equipe e o quanto ela está habituada a trabalhar de forma organizada (Soares, 2007).

As perguntas 11, 12 e 13 do questionário gerencial mostrado no Apêndice A, são utilizadas para analisar que tipos de artifícios são utilizados no cotidiano da equipe para estruturar o desenvolvimento e a comunicação entre a equipe.

As três perguntas permitem que sejam marcadas desde nenhuma até todas as opções de respostas. As opções de resposta referentes às três perguntas foram associadas a pesos. Assim, cada resposta tem um peso respectivo e a soma dos pesos associados às opções marcadas viabilizam a obtenção de uma pontuação.

A atribuição de pesos às opções de respostas é mostrada no “Apêndice B – Atribuição de Peso Gerencial”. A pontuação pode variar de 0 até 24 (pontuação total). A pontuação total indica que a equipe analisada utiliza todos os tipos de recursos

(modelos, técnicas e processos) para estruturar o desenvolvimento e que passa por todas as etapas do ciclo de desenvolvimento de software reconhecidas pela área de Engenharia de Software. Uma equipe nesta situação está habituada a trabalhar com um alto nível de organização (Soares, 2007).

Isto significa que, mesmo que seus membros possuam alguma predileção por trabalharem um ambiente orientado pelo caos, cada um deles está preparado e condicionado a desenvolver software de forma disciplinada. Em função disto e do fato de ser difícil e subjetivo medir as predileções de cada profissional com relação ao caos ou à ordem, este trabalho considera que uma equipe que atinja a pontuação total tem condições de ser detentora de uma cultura fortemente orientada pela ordem, o que seria representado por um ponto no extremo mais externo do eixo Cultura do gráfico da Figura 3.2.

De acordo com a pontuação obtida por uma equipe, pode-se calcular que percentual esta pontuação representa frente à pontuação total; e assim estimar qual é o percentual que orienta a cultura dessa equipe pela ordem.

A partir do percentual obtido, pode-se categorizar a equipe de acordo com sua cultura. Este trabalho estabelece a seguinte classificação:

Tabela 3.11 – Classificação do fator Cultura / Maturidade m Processo

| Posição | Classificação | Descrição |
|----------------|---|---|
| a | Equipes maduras em processos (Acima de 66%) | Facilidade e predisposição para trabalhar de forma organizada |
| b | Maturidade moderada da equipe em processos (Entre 33% e 66%) | Características moderadas para trabalhar de forma organizada |
| c | Equipes imaturas em processos (Menores que 33%) | Dificuldade de trabalhar de forma organizada |

- **Previsibilidade Arquitetural**

O fator “Previsibilidade Arquitetural” é quantificado através da pergunta 16 do questionário gerencial mostrado no Apêndice A. Na pergunta 16A é extraído a previsibilidade em termos de consumo de recursos computacionais, o que aponta o quanto é possível conhecer previamente o projeto em termos de consumo de recursos. Na pergunta 16B é extraída a informação da previsibilidade em termos de manutenibilidade, o que aponta o quanto é possível conhecer previamente o projeto com relação ao impacto de adição de novas funcionalidades e/ou tecnologias. A pergunta

16C extrai a informação de previsibilidade em termos de tolerância à falha, ou seja, o quanto o projeto é flexível com relação à falhas de operação.

As opções de respostas para as perguntas são intervalos que já estão de acordo com os valores do eixo do gráfico (com três opções).

A classificação para o fator Previsibilidade Arquitetural está baseada no framework de customização de processos apresentado em (Royce, 1998), e foi estendida para conter uma classificação intermediária do mesmo.

A classificação do fator Previsibilidade Arquitetural se dará da seguinte forma:

Tabela 3.12 – Classificação do fator Previsibilidade Arquitetural

| Posição | Classificação | Descrição |
|----------------|---|--|
| a | Projetos Alta previsibilidade | Conhecimento prévio elevado da arquitetura necessária |
| b | Projetos previsibilidade moderada | Conhecimento prévio moderado da arquitetura necessária |
| c | Projetos com baixa previsibilidade | Conhecimento prévio pobre da arquitetura necessária |

A classificação final para Previsibilidade Arquitetural do projeto se dará através da “soma” das respostas das perguntas 16A, 16B e 16C sendo que as opções a, b e c de cada questão têm respectivamente os pesos 1, 2 e 3. Assim para soma inferior ou igual a 4 a posição no eixo será “a” (considerando no máximo um valor “b”), para valores entre 5 e 7 a posição será “b” (considerando no máximo dois valor “c” e um “a”) e para valores acima de 7 a posição será “c”.

- **Experiência no Domínio**

O fator “Experiência no Domínio” é quantificado através da pergunta 15 do questionário gerencial mostrado no Apêndice A. Na pergunta 15A é extraída a diversidade de áreas de atuação da empresa, o que aponta fortemente a capacitação da mesma no(s) domínio(s) do problema. Na pergunta 15B é extraída a informação da quantidade de projetos numa determinada área, tal informação visa balancear a informação da pergunta 15A em relação diversidade pela quantidade. A pergunta 15C extrai a informação com relação à repetição da equipe nos projetos, na tentativa de avaliar corretamente a quantidade e diversidade corporativa no domínio dos projetos relacionando a experiência corporativa e a experiência específica da equipe.

As opções de respostas para as perguntas são intervalos de valores que já estão de acordo com a escala do eixo escala do gráfico (com três opções).

A classificação para o fator Experiência no Domínio está baseada no framework de customização de processos apresentado em (Royce, 1998), e foi estendida para conter uma classificação intermediária do mesmo. Nesse caso é levada em consideração também a curva de aprendizado nos domínios dos projetos.

A classificação do fator Experiência no Domínio se dará da seguinte forma:

Tabela 3.13 – Classificação do fator Experiência no Domínio

| Posição | Classificação | Descrição |
|----------------|---|---|
| a | Empresas com alta experiência no domínio | Especialista em desenvolvimento de uma determinada área |
| b | Empresas com experiência no domínio moderada | Especialista em desenvolvimento em uma grande área do conhecimento |
| c | Empresas com baixa experiência no domínio | Desenvolvimento de projetos em qualquer grande área do conhecimento |

A classificação final para Experiência no Domínio do projeto se dará através da “soma” das respostas das perguntas 15A, 15B e 15C sendo que as opções a, b e c de cada questão têm respectivamente os pesos 1, 2 e 3. Assim para soma inferior ou igual a 4 a posição no eixo será “a” (considerando no máximo um valor “b”), para valores entre 5 e 7 a posição será “b” (considerando no máximo dois valor “c” e um “a”) e para valores acima de 7 a posição será “c”.

- **Competência Pessoal**

Para quantificar o fator “Competência Pessoal” é necessário, primeiramente, analisar o nível de habilidade referente a cada indivíduo, para depois estimar a proporção das habilidades dos membros da equipe.

As perguntas 1, 2, 3 e 4 do questionário individual mostrado no apêndice A, são utilizadas para classificar cada membro da equipe de acordo com os três níveis de habilidade mostrados na Tabela 3.14.

As opções de resposta para essas perguntas foram associadas a pesos. Assim, cada resposta tem um peso respectivo e o conjunto de todas as respostas implica na soma de seus pesos e obtenção de uma pontuação. A partir da pontuação obtida, pode-se classificar o indivíduo de acordo com os níveis da Tabela 3.14. A atribuição de pesos às

respostas é mostrada no “Apêndice B – Atribuição de Peso Individual”.

Os três níveis de entendimento apontados por (Cockburn, 2002) sobre métodos de software: 1, 2 e 3; que podem auxiliar na especificação do que pode ser feito pelas pessoas com vários níveis de habilidade.

Tabela 3.14 – Níveis de Competência Pessoal (Cockburn 2002)

| Nível | Características |
|-------|--|
| 3 | Capaz de revisar um método (quebrar suas regras) para ajustá-lo a uma nova situação sem precedentes |
| 2 | Capaz de construir um método que se ajuste a novas situações com precedentes. |
| 1 | Com treinamento, é capaz de seguir os passos de um método procedural (codificar um método simples, realizar refatorações simples, seguir padrões de codificação). Com experiência, pode atingir habilidades do nível 2 |

A extração de informações sobre as habilidades do profissional deve cobrir três aspectos: sua formação, sua experiência em atividades relacionadas ao desenvolvimento de software e seu conhecimento sobre técnicas e métodos de estruturação e melhoria do processo de desenvolvimento do software.

Formação do profissional: A pergunta 1 esclarece se o respondente é da área da Computação ou se é um profissional pertencente a outra área de atuação e que, ainda assim, desenvolve software. Foi atribuído o peso 2 para a resposta sim e peso 0 para a resposta não. A justificativa para essa distribuição é que um profissional da área da Computação adquire conceitos e treinamento específicos durante sua formação que um profissional de outra área não possui e talvez só chegue a alcançar depois de anos de experiência com desenvolvimento de software e estudos complementares (Soares, 2007). Isso contribui para que o profissional da área de Computação possua, naturalmente, um nível mais alto de habilidade de acordo com a Tabela 3.14.

Experiência do profissional: Outro fator que influencia o nível de habilidade do profissional é a sua experiência em atividades de desenvolvimento de software. Os itens A, C, D e F da pergunta 2 têm o propósito de analisar este fator. Esses itens destinam-se a estimar a experiência do profissional em cada uma das atividades pertencentes ao ciclo de desenvolvimento do software. As opções de resposta são as mesmas para estas seis perguntas. Quanto maior a experiência, maior é o peso atribuído à resposta. Foi atribuído peso zero às respostas dos itens B e E da pergunta 2 por não representarem atividades que aumentem significativamente o nível de habilidade de um desenvolvedor de software de acordo com os níveis da Tabela 3.14 (Soares, 2007)

Outros conhecimentos: Além da formação e experiência do profissional, outro fator importante é o seu conhecimento sobre técnicas e métodos de estruturação e melhoria do processo de desenvolvimento do software. As Perguntas 3 e 4 destinam-se a esclarecer qual é o nível de conhecimento do respondente sobre o CMM - Capability Maturity Model, MPS.BR – Melhoria do Processo de Software Brasileiro e sobre processos de desenvolvimento de software como RUP (Rational Unified Process), PSP (Personal Software Process) e TSP (Team Software Process). As respostas que indicam um conhecimento menor sobre esses itens são associadas a pesos de menor valor. O peso máximo é atribuído às respostas que indiquem que o desenvolvedor já trabalhou em um ambiente que utilize a técnica ou processo citado no enunciado da pergunta o que indica um diferencial significativo para aumentar o nível de habilidade do profissional de acordo com a Tabela 3.14 (Soares, 2007).

Classificação do nível de habilidade do respondente: Com o peso de cada uma das opções de resposta já estabelecido, é possível obter a pontuação respectiva a cada respondente e classificá-lo de acordo com os níveis de habilidade da Tabela 3.14 como é mostrado a seguir:

Tabela 3.15 – Pontuação necessária para cada nível de habilidade

| Nível de Habilidade | Características |
|-----------------------|------------------|
| Profissional 3 | ≥ 31 pontos |
| Profissional 2 | 25 a 30 pontos |
| Profissional 1 | ≤ 24 pontos |

A classificação final da Competência Pessoal da equipe se dará pela maioria de profissionais em cada nível. Em caso de empate fica definido que o nível adotado é o mais baixo por apresentar menor risco para proposição de práticas contrárias ao perfil.

Tabela 3.16 – Classificação do fator Competência Pessoal

| Posição | Classificação | Descrição |
|----------|---|---|
| a | Equipes experientes (Maioria “3”) | Equipe experiente, com capacidade para adequar o trabalho a qualquer situação sem precedentes |
| b | Equipes com experiência moderada (Maioria “2”) | Equipe com experiência moderada, com capacidade para adequar o trabalho a situações com precedentes |
| c | Equipes inexperientes (Maioria “1”) | Equipe inexperiente com capacidade apenas de seguir os passos de um método procedural |

4 CONJUNTO DE BOAS PRÁTICAS – BUILDING BLOCKS

4.1 Investigação de Práticas para nível G MPS.BR

4.1.1 Gerência de Projeto (Complexidade Gerencial)

A Gerência de Projetos (GPR) envolve várias atividades, como: desenvolver um plano geral de controle do projeto; obter o comprometimento e mantê-lo ao longo de toda a execução do projeto; e conhecer o progresso do projeto, de maneira que ações corretivas possam ser tomadas quando a execução do projeto desviar do planejado (MPS.BR, 2009).

O desenvolvimento do plano do projeto inclui: identificar e estimar o escopo, os produtos de trabalho e as tarefas do projeto; estabelecer recursos necessários; identificar e analisar riscos do projeto; estabelecer compromissos; e definir cronograma de execução baseado no ciclo de vida definido para o projeto. O plano do projeto estabelece a base de execução e controle para as atividades do projeto junto aos seus interessados (especialmente o cliente). Todos os interessados devem estar comprometidos com ele (MPS.BR, 2009).

O progresso da execução do projeto é determinado pela comparação dos atributos reais de produtos de trabalho e tarefas, esforço, custo e cronograma com o que foi planejado nos marcos ou em pontos de controle predefinidos no planejamento do projeto. Um marco é um ponto de revisão, por exemplo, o início ou o final de cada fase do projeto ou algumas atividades de fundamental importância para o seu sucesso. A revisão de início de fase de projeto tem por objetivo verificar se as condições para que uma fase seja iniciada estão atendidas. Pode ser que, mesmo que a fase anterior não esteja encerrada, seja possível iniciar a nova fase, nas condições atendidas e com prazos para o cumprimento de algumas outras condições. A revisão de fim de fase de projeto tem por objetivo verificar se todos os critérios de encerramento de fase foram cumpridos. As revisões em marcos podem ter um caráter formal, com participação de gerências superiores, representantes do cliente e outras partes interessadas no projeto. Sempre que necessário, deve-se realizar um replanejamento e uma nova análise de sua viabilidade (MPS.BR, 2009).

Pontos de controle representam pontos entre um marco e outro nos quais revisões são realizadas para avaliar o andamento do projeto, porém, não estão no caminho crítico

do projeto, ou seja, o projeto pode prosseguir mesmo que a revisão de um ponto de controle não tenha sido concluída. A visibilidade apropriada possibilita a tomada de ações corretivas quando o status do projeto se desvia significativamente do esperado. Tais ações podem exigir o replanejamento, para incluir a revisão do plano original, o estabelecimento de novos acordos ou atividades adicionais de mitigação de riscos no plano (MPS.BR, 2009).

O PMI (Project Management Institute), um dos mais conceituados e reconhecidos institutos na área de gerenciamento de projetos, é responsável pela publicação e atualização do PMBOK (Project Management Body of Knowledge) (PMI, 2008). O PMBOK é um guia em gerência de projetos. Ele agrupa o conhecimento em gerência de projetos que é amplamente reconhecido como as boas práticas deste tipo de gerenciamento.

O gerenciamento de projeto na visão do PMBOK (Project Management Body of Knowledge) (PMI, 2008) é a aplicação de conhecimento, habilidades, ferramentas e técnicas às atividades do projeto, a fim de atender aos seus requisitos. Gerenciar projeto envolve identificar as necessidades, estabelecer objetivos claros e viáveis e balancear as demandas conflitantes em termos de qualidade, escopo, tempo e custo. Um processo de gerenciamento de projeto identifica, estabelece, coordena e produz um produto, de acordo com seus requisitos.

O IEEE (Institute of Electrical and Electronics Engineers), em seu Glossário Padrão de Terminologias da Engenharia de Software (IEEE, 1990), diz que a gerência de projetos de software pode ser definida como a aplicação de planejamento, coordenação, medição, monitoramento, controle e divulgação de relatórios, com o intuito de garantir que o desenvolvimento e a manutenção de software sejam sistemáticos, disciplinados e qualificados. E, segundo a norma internacional ISO/IEC 12207, o propósito da gerência de projetos é identificar, estabelecer, coordenar e monitorar as atividades, tarefas e recursos que um projeto necessita para produzir um produto, no contexto dos requisitos e restrições do projeto (ISO/IEC, 2008).

Os resultados esperados para gerência de projeto no nível G do MPS.BR são (MPS.BR, 2009):

- ✓ GPR 1. O escopo do trabalho para o projeto é definido;
- ✓ GPR 2. As tarefas e os produtos de trabalho do projeto são dimensionados utilizando métodos apropriados;

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

4.1.2 Gerência de Requisitos (Complexidade Técnica)

O principal objetivo da Gerência de Requisitos é controlar a evolução dos requisitos. O processo Gerência de Requisitos (GRE) gerencia todos os requisitos recebidos ou gerados pelo projeto, incluindo requisitos funcionais e não-funcionais, bem como os requisitos impostos ao projeto pela organização (MPS.BR, 2009).

Para assegurar que o conjunto de requisitos acordados seja gerenciado e forneça apoio às necessidades de planejamento e execução do projeto, a organização deve executar um conjunto de passos definidos e apropriados. Quando um projeto recebe requisitos de um fornecedor de requisitos (pessoa autorizada a participar de sua definição e a solicitar modificação), estes devem ser revisados para resolver questões e prevenir o mau entendimento, antes que os requisitos sejam incorporados ao escopo do projeto. Quando o fornecedor de requisitos e a organização chegam a um acordo, é obtido um compromisso das demais partes interessadas sobre os requisitos (MPS.BR, 2009).

Outras atribuições do processo de Gerência de Requisitos são documentar as mudanças nos requisitos e suas justificativas, bem como manter a rastreabilidade bidirecional entre os requisitos e produtos de trabalho em geral e identificar inconsistências entre os requisitos, os planos do projeto e os produtos de trabalho do projeto (MPS.BR, 2009).

Uma boa comunicação com os fornecedores de requisitos é fundamental para assegurar um bom entendimento das necessidades do cliente e dos requisitos do projeto e, conseqüentemente, aumentar as chances de sucesso do projeto.

Existem diversos assuntos ligados a requisitos que devem ser tratados com os fornecedores de requisitos, como por exemplo: definição de requisitos, aprovação de requisitos, solicitação de mudança nos requisitos, dentre outros.

Segundo (Dorfmann e Thayer, 1990), requisito de software representa a capacidade requerida pelo usuário que deve ser encontrada ou possuída por um determinado produto ou componente de produto para resolver um problema ou alcançar um objetivo ou para satisfazer a um contrato, a um padrão, a uma especificação ou a outros documentos formalmente impostos.

A gerência de requisitos envolve identificar os requisitos do produto e dos componentes do produto do projeto, bem como estabelecer e manter um acordo entre o cliente e a equipe de projeto sobre esses requisitos. Também é objetivo da gerência de

requisitos controlar e tratar as mudanças nos requisitos ao longo do desenvolvimento.

Para apoiar o processo de mudança de requisito, é fundamental definir e manter a rastreabilidade dos requisitos. Rastreabilidade é o grau em que o relacionamento pode ser estabelecido entre dois ou mais produtos de desenvolvimento de software especialmente produtos que tenham uma relação de predecessor sucessor ou de mestre subordinado com outro; por exemplo, o grau em que requisitos e projeto (design) de um determinado componente de software combinam (IEEE, 1990).

Quando os requisitos são bem gerenciados, a rastreabilidade pode ser estabelecida desde um requisito fonte, passando por todos os níveis de decomposição do produto até seus requisitos de mais baixo nível e destes até o seu requisito fonte. Esta rastreabilidade bidirecional auxilia a determinar se todos os requisitos fonte foram completamente tratados e se todos os requisitos de mais baixo nível podem ser rastreados para uma fonte válida (SEI, 2006).

A rastreabilidade bidirecional deve acontecer tanto de forma horizontal quanto vertical. A rastreabilidade horizontal estabelece a dependência entre os requisitos ou produtos de trabalho em um mesmo nível, por exemplo, rastreabilidade dos requisitos entre si ou rastreabilidade entre códigos de unidades dependentes. A rastreabilidade vertical estabelece uma rastreabilidade bidirecional desde um requisito fonte, passando pelos seus requisitos de mais baixo nível, até o nível de decomposição mais baixo do produto, por exemplo, códigos de unidade ou módulos do software. Esse mecanismo deve permitir também rastrear itens do nível mais baixo de decomposição do produto até o(s) seu(s) requisito(s) fonte. A rastreabilidade vertical auxilia a determinar se todos os requisitos fonte foram completamente tratados e se todos os requisitos de mais baixo nível ou códigos de unidade podem ser rastreados para um requisito fonte válido. A rastreabilidade vertical bidirecional possibilita, então, rastrear requisitos e produtos de trabalho a códigos de unidade ou módulos do software implementados.

Esse mecanismo de rastreabilidade vertical é essencial para a realização da análise de impacto de mudanças de requisitos, por exemplo, para identificar de que forma uma mudança de requisito impacta nos planos do projeto que contêm as estimativas aprovadas de esforço e custo para os produtos de trabalho e tarefas, bem como os códigos de unidade ou módulos do software que necessitam ser modificados. Por essas análises, o responsável pela gerência do projeto é capaz de negociar com o cliente alterações nos planos do projeto para atender às solicitações de mudanças de requisitos e, ao mesmo tempo, minimizar os riscos do projeto, como por exemplo, desvios de

cronograma e de custos.

Os resultados esperados para gerência de requisitos no nível G do MPS.BR são (MPS.BR, 2009):

- ✓ GRE 1. Os requisitos são entendidos, avaliados e aceitos junto aos fornecedores de requisitos, utilizando critérios objetivos;
- ✓ GRE 2. O comprometimento da equipe técnica com os requisitos aprovados é obtido;
- ✓ GRE 3. A rastreabilidade bidirecional entre os requisitos e os produtos de trabalho é estabelecida e mantida;
- ✓ GRE 4. Revisões em planos e produtos de trabalho do projeto são realizadas visando identificar e corrigir inconsistências em relação aos requisitos;
- ✓ GRE 5. Mudanças nos requisitos são gerenciadas ao longo do projeto.

4.2 Investigação de Práticas CMMI

4.2.1 Gestão de Projetos

As áreas de processo de Gestão de Projeto tratam das atividades de gestão relacionadas a planejamento, monitoramento e controle de projeto (SEI, 2006).

As áreas de processo do CMMI de Gestão de Projeto são:

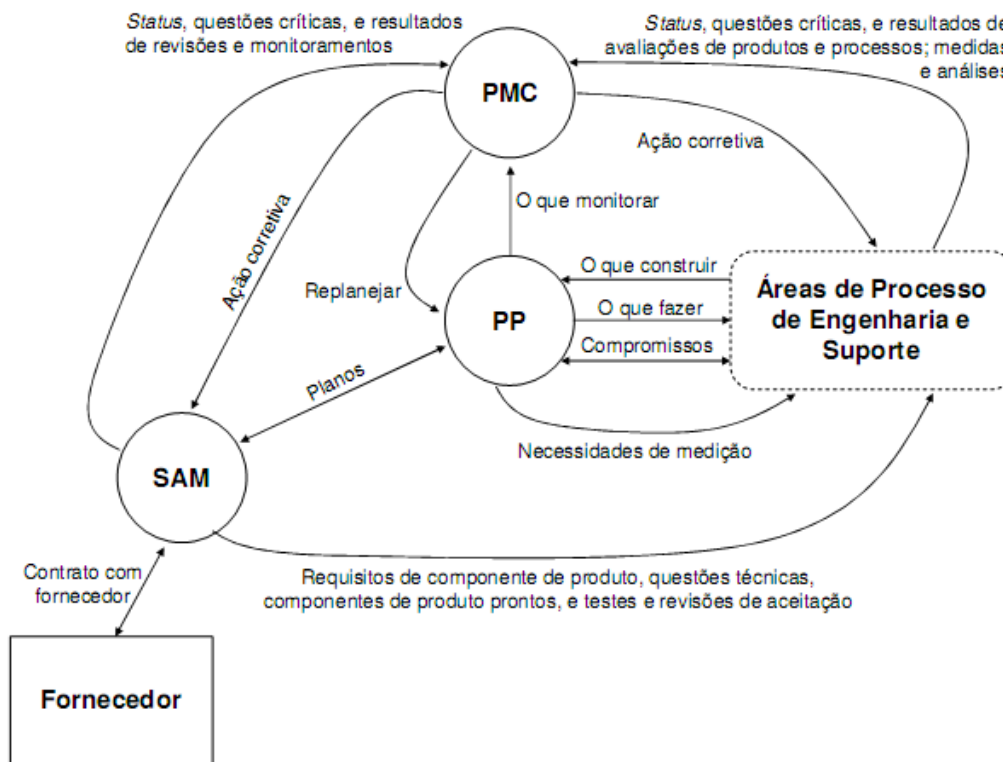
- ✓ Planejamento de Projeto.
- ✓ Monitoramento e Controle de Projeto.
- ✓ Gestão de Contrato com Fornecedores.
- ✓ Gestão Integrada de Projeto.
- ✓ Gestão de Riscos.
- ✓ Gestão Quantitativa de Projeto.

Áreas de Gestão de Projetos Básica

As áreas de processo de Gestão de Processo básicas tratam das atividades relacionadas ao estabelecimento e manutenção do plano de projeto, estabelecimento e manutenção de compromissos, monitoramento do progresso em relação ao plano, implementação de ações corretivas e gestão de contratos com fornecedores.

A Figura 4.1 apresenta uma visão panorâmica das interações entre as áreas de processo de Gestão de Projeto básicas e também com outras categorias de área de processo. A área de processo Planejamento de Projeto inclui a elaboração do plano de

projeto, o envolvimento apropriado das partes interessadas, a obtenção de comprometimento com o plano e sua manutenção. Quando se utiliza IPPD (Desenvolvimento Integrado de Processo e Produto), as partes interessadas não representam apenas a experiência técnica necessária para desenvolvimento de produtos e processos, mas também as implicações de negócio desses desenvolvimentos (SEI, 2006).



PMC – *Project Monitoring and Control* (Monitoramento e Controle de Projeto)

PP – *Project Planning* (Planejamento de Projeto)

SAM – *Supplier Agreement Management* (Gestão de Contrato com Fornecedores)

Figura 4.1 – Áreas de Gestão de Projetos Básica (adaptado: SEI, 2006)

O planejamento tem início com os requisitos que caracterizam o produto e o projeto (“O que construir”, na Figura 4.1). O plano de projeto cobre as várias atividades de gestão e desenvolvimento de projeto executadas no âmbito do projeto. O projeto revisa outros planos que o afetam, gerados por várias partes interessadas, e estabelece compromissos com elas a respeito de suas contribuições para o projeto. São exemplos os planos de gestão de configuração, de verificação, e de medição e análise.

A área de processo “Monitoramento e Controle de Projeto” inclui atividades de monitoramento e de implementação de ações corretivas. O plano de projeto especifica o nível apropriado de monitoramento, a frequência de revisões de progresso e as medidas

utilizadas para monitorar o progresso do projeto, o qual é basicamente determinado comparando-se o status do projeto com o plano. Implementam-se ações corretivas (incluindo replanejamento) conforme apropriado, quando o status do projeto desvia significativamente dos valores esperados (SEI, 2006).

A área de processo Gestão de Contrato com Fornecedores trata das necessidades de aquisição de partes do trabalho que são produzidas por fornecedores. As fontes de produtos utilizadas para satisfazer aos requisitos de projetos são identificadas proativamente. O fornecedor é selecionado, e é estabelecido um contrato para que se possa gerenciá-lo. O progresso e o desempenho do fornecedor são acompanhados por meio do monitoramento de processos e produtos de trabalho selecionados, e o contrato com o fornecedor é atualizado conforme apropriado. Realizam-se revisões e testes de aceitação nos componentes de produto gerados pelo fornecedor (SEI, 2006).

4.2.2 Engenharia no CMMI

As áreas de processo de “Engenharia” tratam de atividades de desenvolvimento e manutenção das diversas disciplinas de Engenharia (SEI, 2006).

As áreas de processo de Engenharia são escritas utilizando uma terminologia genérica de Engenharia, de modo que qualquer disciplina técnica envolvida no processo de desenvolvimento do produto (por exemplo, Engenharia de Software ou Engenharia Mecânica) possa utilizá-la para melhoria de processo (SEI, 2006).

As áreas de processo de Engenharia também integram os processos associados a diferentes disciplinas de Engenharia em um único processo de desenvolvimento de produto, apoiando uma estratégia de melhoria de processo orientada a produto. Essa estratégia está mais preocupada em alcançar objetivos estratégicos essenciais do que as disciplinas técnicas específicas. Tal abordagem para processos evita, de forma efetiva, a tendência em direção a um pensamento compartimentalizado das organizações.

As áreas de processo de Engenharia aplicam-se ao desenvolvimento de qualquer produto ou serviço no domínio de desenvolvimento (por exemplo: produtos de software, produtos de hardware, serviços ou processos).

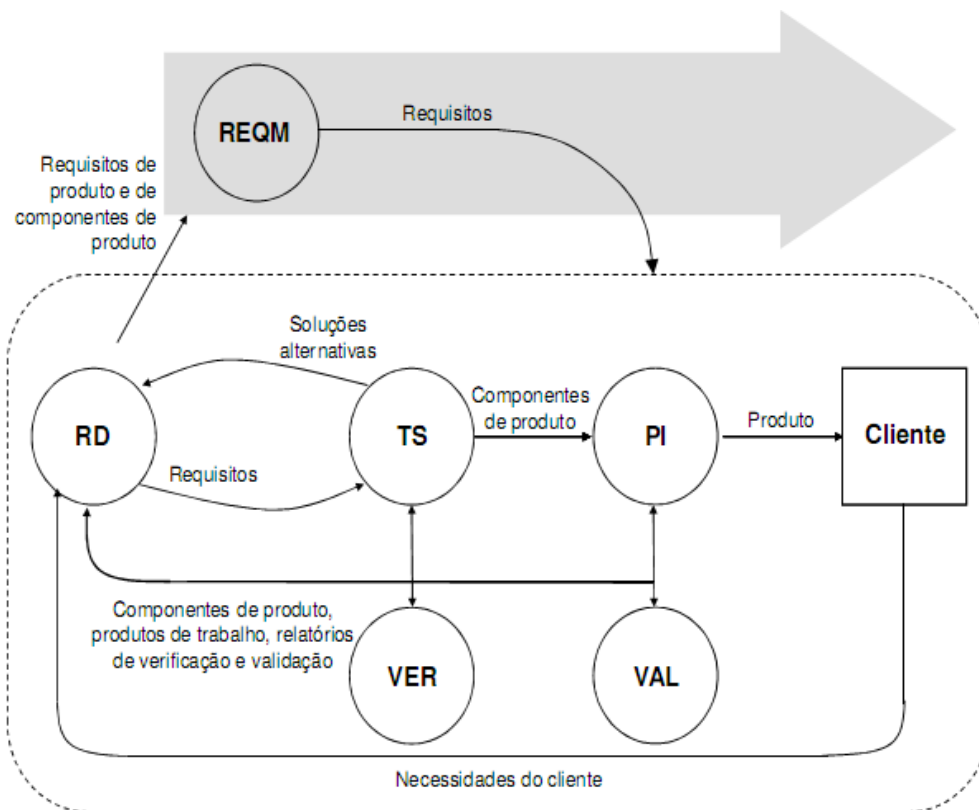
As bases técnicas para IPPD (Desenvolvimento Integrado de Processo e Produto) estão fundamentadas em uma abordagem robusta de Engenharia de Sistemas que engloba desenvolvimento no contexto das fases da vida do produto. As áreas de processo de Engenharia fornecem essas bases técnicas. Além disso, a implementação de IPPD é tratada por meio de extensões às práticas específicas das áreas de processo de

Engenharia que enfatizam o desenvolvimento concorrente (paralelo) e aplicam-se às fases da vida do produto (SEI, 2006).

As áreas de processo de Engenharia do CMMI são:

- ✓ Desenvolvimento de Requisitos.
- ✓ Gestão de Requisitos.
- ✓ Solução Técnica.
- ✓ Integração de Produto.
- ✓ Verificação.
- ✓ Validação.

A Figura 4.2 apresenta uma visão panorâmica das interações entre as seis Áreas de Processo de Engenharia.



REQM – *Requirements Management* (Gestão de Requisitos)

RD – *Requirements Development* (Desenvolvimento de Requisitos)

TS – *Technical Solution* (Solução Técnica)

PI – *Product Integration* (Integração de Produto)

VER – *Verification* (Verificação)

VAL – *Validation* (Validação)

Figura 4.2 – Áreas do Processo de Engenharia (adaptado: SEI, 2006)

A área de processo Desenvolvimento de Requisitos identifica as necessidades do cliente e traduz essas necessidades em requisitos de produto. O conjunto de requisitos de produto é analisado para gerar uma solução conceitual de alto nível. Esse conjunto de requisitos é então alocado para estabelecer um conjunto inicial de requisitos de produto. Outros requisitos que ajudam a definir o produto são derivados e alocados aos componentes de produto. Esse conjunto de requisitos de produto e de componentes de produto descreve claramente o desempenho do produto, suas características de design e seus requisitos de verificação, de forma que o desenvolvedor possa entendê-los e utilizá-los (SEI, 2006).

A área de processo Desenvolvimento de Requisitos fornece requisitos para a área de processo Solução Técnica, onde os requisitos são convertidos em arquitetura do produto, design de componentes de produto e no próprio componente de produto (por exemplo, código e fabricação). Os requisitos também são fornecidos à área de processo Integração de Produto, em que os componentes de produto são combinados e as interfaces são verificadas para assegurar que os requisitos de interface fornecidos pelo Desenvolvimento de Requisitos sejam atendidos (SEI, 2006).

A área de processo Gestão de Requisitos mantém os requisitos. Ela descreve atividades para obter e controlar mudanças de requisitos e assegurar que outros planos e dados relevantes se mantenham atualizados. Além disso, fornece rastreabilidade de requisitos, desde o cliente até o produto ou o componente de produto (SEI, 2006).

A Gestão de Requisitos assegura que as mudanças ocorridas nos requisitos sejam refletidas em planos, atividades e produtos de trabalho do projeto. Esse ciclo de mudanças pode afetar todas as outras áreas de processo de Engenharia. Assim, a gestão de requisitos é uma seqüência de eventos dinâmica e freqüentemente recursiva. A área de processo Gestão de Requisitos é fundamental para um processo de Engenharia controlado e gerenciado (SEI, 2006).

A área de processo Solução Técnica desenvolve pacotes de dados técnicos para componentes de produto que serão utilizados pela área de processo Integração de Produto ou pela área de processo Gestão de Contrato com Fornecedores. Soluções alternativas são examinadas a fim de escolher o design ótimo com base em critérios previamente estabelecidos. Esses critérios podem variar significativamente para os diversos produtos, dependendo do tipo, ambiente operacional, requisitos de desempenho, requisitos de suporte, e custo ou prazo de entrega do produto. A tarefa de escolha da solução final faz uso de práticas específicas da área de processo Análise e

Tomada de Decisões (SEI, 2006).

A área de processo Solução Técnica apóia-se nas práticas específicas da área de processo Verificação para realizar verificações de design e revisões por pares durante o design e antes da construção final. A área de processo Verificação assegura que produtos de trabalho selecionados satisfaçam aos seus requisitos especificados, selecionando métodos para sua verificação em relação aos requisitos especificados. Geralmente, a verificação é um processo incremental, iniciado com a verificação de componentes de produto e concluído com a verificação de produtos completos (SEI, 2006).

A verificação também envolve revisão por pares, que é um método comprovado para a remoção efetiva e antecipada de defeitos e proporciona um conhecimento valioso sobre os produtos de trabalho e componentes de produto que estão sendo desenvolvidos.

A área de processo Validação valida produtos, de forma incremental, com relação às necessidades do cliente. A validação pode ser realizada no ambiente real de operação ou em um ambiente operacional simulado. Um aspecto importante para esta área de processo é o alinhamento dos requisitos de validação com o cliente. O escopo da área de processo Validação engloba validação de produtos, componentes de produto, produtos de trabalho intermediários e processos. Frequentemente, esses elementos podem ter que ser verificados e validados novamente. Questões críticas encontradas durante a validação são normalmente solucionadas por meio da área de processo Desenvolvimento de Requisitos ou Solução Técnica (SEI, 2006).

A área de processo Integração de Produto contém as práticas específicas associadas à geração da melhor seqüência de integração possível, envolvendo a integração de componentes de produto e a entrega do produto ao cliente. A Integração de Produto utiliza práticas específicas das áreas de processo Verificação e Validação ao implementar o processo de integração de produto. As práticas de verificação possibilitam a verificação das interfaces e dos requisitos de interface de componentes de produto antes da integração do produto. Esse é um evento essencial no processo de integração. Durante a integração de produto no ambiente operacional, utilizam-se as práticas específicas da área de processo Validação (SEI, 2006).

4.3 Levantamento e Estrutura de Boas Práticas

Esse trabalho de pesquisa está direcionado à investigação do mapeamento e adoção de um conjunto de boas práticas de engenharia de software que esteja aderente à realidade de MPEs. A partir desse contexto, foi investigado o trabalho de (Leal, 2009) com

levantamento sobre boas práticas de desenvolvimento de software em ES e verificou-se que organizações bem sucedidas adotam essas práticas com a intenção de atuar de forma eficaz na predição de problemas com a produção de software.

As práticas, aqui denominadas boas práticas, reduzem as chances de insucesso nos projetos, pois tratam as causas dos problemas. (Kruchten, 2004) propõe as seguintes práticas:

- 1) **Desenvolvimento Iterativo:** utilizado para esclarecer mais cedo os desentendimentos com relação a requisitos, encoraja o retorno do usuário com relação aos requisitos levantados, permite que a equipe trabalhe em um nível de abstração mais adequado, facilita a elaboração e a aplicação de testes contínuos e iterativos permitindo a avaliação objetiva de status do projeto, auxilia na correção das inconsistências entre requisitos, projetos e implementação, permite a distribuição da carga de tarefas da equipe nas devidas fases do projeto, há um aprendizado progressivo da equipe com uma carga menor e mais objetiva de tarefas executadas distribuídas em diferentes fases.
- 2) **Gerência de Requisitos:** é possível melhorar a comunicação entre as partes envolvidas, uma vez que ela é feita com base em requisitos definidos e registrados. Os requisitos podem ser priorizados, filtrados e rastreados, é possível a avaliação objetiva das funcionalidades do projeto e do desempenho desejado, as inconsistências são encontradas em fases preliminares e corrigidas mais cedo. Com suporte adequado de ferramentas, é possível ter um registro centralizado de requisitos, atributos e rastreabilidade do sistema, com o suporte de documentação adequada e disponível.
- 3) **Arquitetura Baseada em Componentes:** auxilia na organização do software, na seleção de elementos estruturais que compõem o sistema e suas interfaces, permite definir o comportamento dos elementos estruturais especificados nas suas colaborações, permite a reutilização dos componentes em estruturas maiores. Além disso, os componentes ajudam a obter arquiteturas mais estáveis, a modularidade facilita a focalização em elementos mais sujeitos a mudança, os componentes são uma base natural para gestão de configurações.
- 4) **Modelagem Visual:** permite que casos de uso caracterizem os cenários e os modelos do projeto de software sem ambigüidades, auxiliando na percepção de inconsistências de forma mais eficaz, permite que alguns detalhes fiquem

escondidos quando necessário, permite garantir de forma preliminar a qualidade da aplicação, pois, facilita no desenvolvimento do projeto. E com ferramentas de modelagem visual pode-se melhor trabalhar os conceitos da *Unified Modeling Language* (UML).

- 5) **Verificação Contínua de Qualidade:** permite a avaliação de status do projeto uma vez que é feita objetivamente com base em resultados dos testes reais, e não documentos em papel, permite a correção de inconsistências em requisitos, projeto e implementação. Nessa atividade o foco dos testes se baseia em áreas de maior risco, minimizando as chances de ocorrências de problemas. Os defeitos são identificados mais cedo, trabalhando de forma eficaz na redução do custo de correção. A utilização de ferramentas automáticas de testes permite testar as funcionalidades, a confiabilidade e o desempenho do sistema produzido.
- 6) **Gestão de Alterações:** deverá ser pré-definida utilizando-se de mecanismos que descrevam o fluxo de trabalho em caso de mudança de requisitos. A requisição formalizada de mudanças facilita a comunicação e o entendimento do que se deve ser implementado. As equipes devem trabalhar com espaços bem definidos e isolados, reduzindo a interferência entre equipes trabalhando em paralelo na execução da linha base do projeto. Devem ser utilizadas de medições estatísticas para identificar a frequência das solicitações de alterações com o objetivo de se identificar o status do projeto e o motivo das mudanças, para que seja possível controlar a essas solicitações.

As boas práticas descritas por (Kruchten, 2004), são baseadas na proposta do processo moderno sugerido por (Royce, 1998), que propõe cinco princípios para o processo moderno de desenvolvimento de software. A Figura 4.3 apresenta esses princípios:

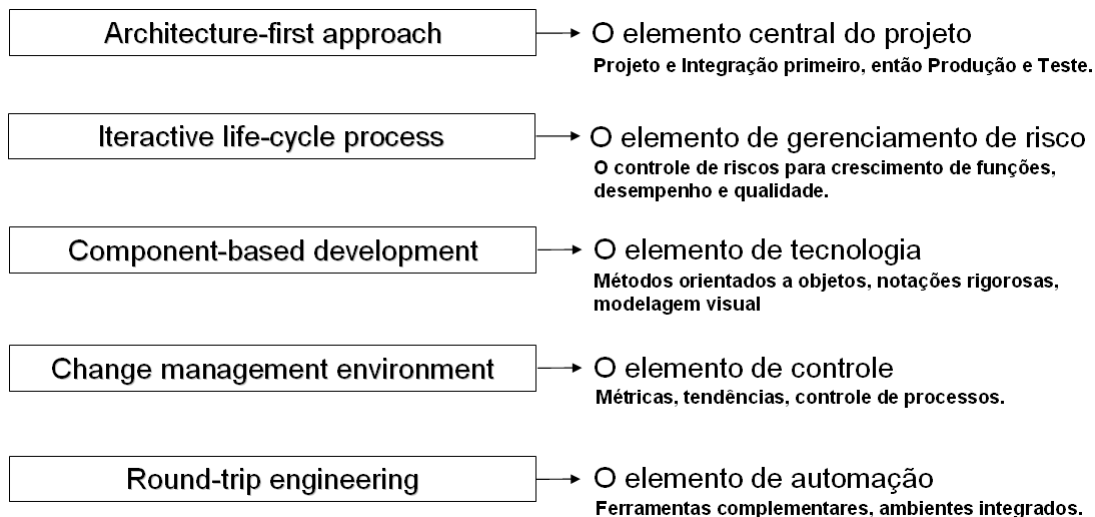


Figura 4.3 – Princípios de um processo moderno de software (Royce 1998)

- 1) *Architecture-first approach* (o elemento central do projeto): nesse princípio é importante o foco dos esforços no projeto de implementação e de integração do software e, só após isso, os esforços devem ser concentrados na produção e nos testes.
- 2) *Iterative life-cycle process* (o elemento de gerenciamento de risco): a abordagem nesse princípio deve ser concentrada no controle dos riscos a fim de suportar o crescimento das funcionalidades do software, suportar o crescimento e garantir o desempenho e a qualidade.
- 3) *Component-based development* (o elemento de tecnologia): este princípio sugere a adoção de modelos formais de documentação, de modelagem visual e métodos orientados a objetos para representar os requisitos que estão sendo especificados.
- 4) *Change management environment* (o elemento de controle): um processo de software moderno deve estar amparado por métricas, por indicadores de tendência que permita a tomada de decisão para a correção de desvios do planejado e de um controle organizado de todas as etapas do processo de desenvolvimento.
- 5) *Round-trip engineering* (o elemento da automação): no processo moderno, há a necessidade de se amparar o processo com ferramentas complementares de apoio ao desenvolvimento, o controle do projeto e a utilização de ambientes integrados.

Diante da apresentação das boas práticas propostas por (Kruchten, 2004) ou dos

princípios de (Royce, 1998), foram investigados na literatura técnica quais seriam os elementos que poderiam compor inicialmente e de forma organizada uma estrutura hierárquica que representasse técnicas, artefatos, ferramentas e métodos que pudessem dar amparo a aplicação das boas práticas para o desenvolvimento de software em MPEs. Os elementos foram escolhidos com base na sua facilidade de implementação e aderência às características das MPEs.

Uma estrutura sugerida de elementos pode ser vista na Figura 4.4 proposta por (Leal, 2009) a partir estudos de propostas de autores de ES como (Pfleeger, 2004), (Pressman, 2001), (Sommerville, 2003), (Wazlawick, 2004), (Pádua Filho, 2003), (Yourdon, 1997), (Dennis & Wixom, 2005) e também da literatura especializada em gerenciamento de projetos baseado no (PMI, 2008). A estrutura não pretende esgotar as possibilidades apresentadas entre as diversas ferramentas, técnicas e métodos de ES, mas é utilizada no trabalho como um direcionamento para o pensamento sistematizado a respeito do domínio aqui apresentado.

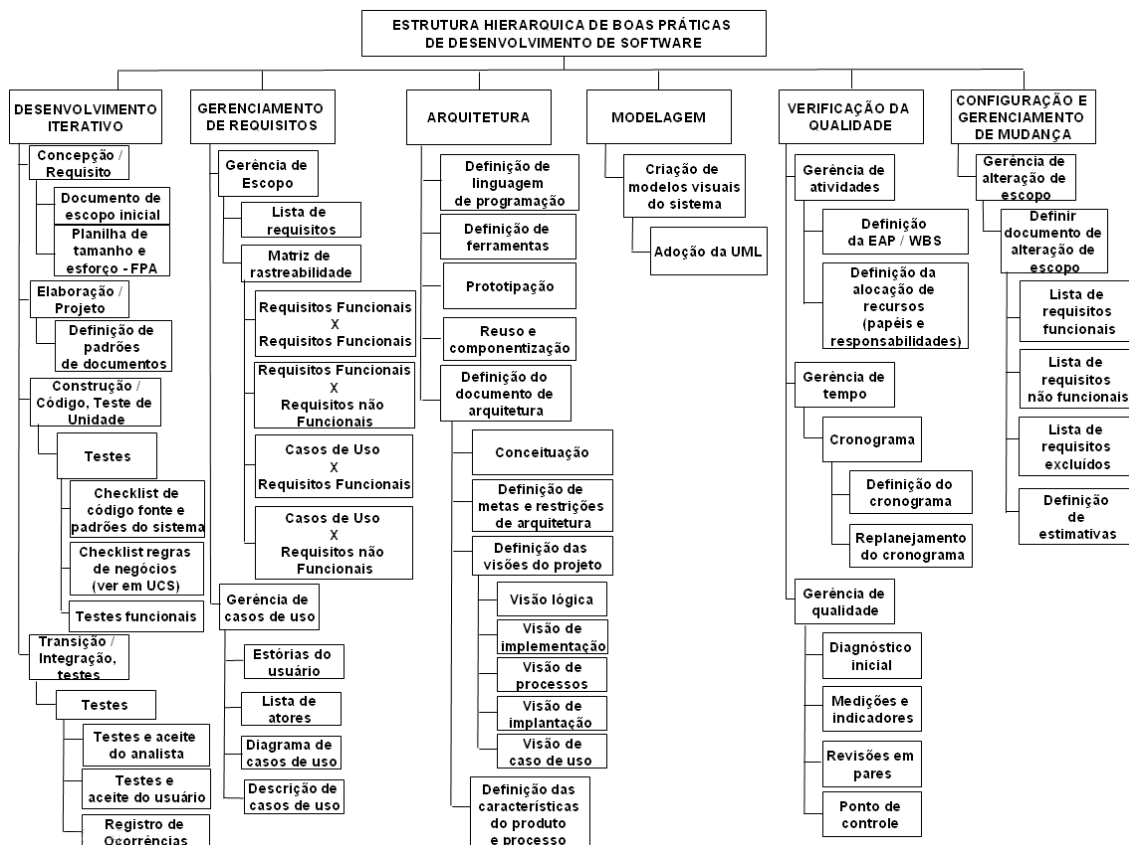


Figura 4.4 – Elementos para apoio às boas práticas de desenvolvimento (adaptado: Leal, 2009)

A estrutura hierárquica apresentada na Figura 4.4 é a base para seleção da lista de boas práticas explicitadas na Seção 4.4.

4.4 Lista de Boas Práticas – *Building Blocks*

Foi realizado um estudo e relacionamento das práticas de engenharia de software para o nível G do MPS.BR (Gerência de Projetos e Gerência de Requisitos) apresentada na Seção 4.1, das práticas de gestão de projetos e engenharia para CMMI apresentada na Seção 4.2 e do levantamento da estrutura de boas práticas apresentada na Seção 4.3. O conjunto de boas práticas foi então selecionado mesclando práticas dessas três fontes. Tais práticas compõem a lista que será utilizada no mapeamento tratado no Capítulo 5.

As práticas foram chamadas de building blocks (blocos de construção) para demonstrar o caráter de independência entre elas. Além disso, foram separadas em dois conjuntos: Blocos de Práticas Gerenciais e Blocos de Práticas Técnicas.

Os blocos de práticas estão listados a seguir:

Blocos de Práticas Gerenciais

- ✓ Definição inicial de escopo (Documento de visão)
- ✓ Representação do escopo através de Estrutura Analítica do Projeto (EAP)
- ✓ Dimensionamento do projeto sob o ponto de vista do usuário (Planilha de APF)
- ✓ Definição de um modelo e as fases do ciclo de vida do projeto (Iterativo)
- ✓ Dimensionamento de esforço e custo do projeto (Relação entre tamanho, produtividade e valor)
- ✓ Cronograma simples com datas e responsáveis pelas atividades
- ✓ Cronograma com dependências entre as atividades, marcos de controle e produtos de trabalho gerados
- ✓ Identificação e monitoramento formal de riscos do projeto, com relação de impacto e probabilidade
- ✓ Gerência formal de riscos incluindo além de identificação, o planejamento de resposta aos riscos
- ✓ Planejamento de recursos humanos para o projeto, determinando funções, responsabilidades, relações hierárquicas
- ✓ Planejamento de recursos e ambiente necessários, incluindo, por exemplo, equipamentos, ferramentas, serviços, componentes, viagens, etc.
- ✓ Controle formal de orçamento do projeto (Planilha de orçamento)
- ✓ Gerência de configuração do projeto, com identificação, coleta, armazenamento e distribuição das informações incluindo regras de segurança e confidencialidade

- ✓ Integração dos planejamentos realizados garantindo que a dependência entre eles seja monitorada
- ✓ Realização de estudos de viabilidade em marcos específicos para garantir que o projeto está sendo realizado de forma viável
- ✓ Acompanhamento incluindo revisão e obtenção de compromisso dos planos do projeto com os interessados
- ✓ Comunicação formal das atividades do projeto com todos os interessados
- ✓ Identificação e monitoramento formal de problemas do projeto
- ✓ Gerência formal de problemas incluindo além da identificação o registro de ações para correção de desvios

Blocos de Práticas Técnicas

- ✓ Identificação formal dos requisitos (Lista de Requisitos funcionais e não funcionais, Estórias de Usuários)
- ✓ Identificação formal dos fornecedores de requisitos do projeto
- ✓ Definição de critérios objetivos para validação dos requisitos (*Checklist* de requisitos)
- ✓ Comunicação e comprometimento formal dos requisitos com a equipe técnica
- ✓ Rastreabilidade bidirecional entre os requisitos e os produtos de trabalho (Requisitos x Requisitos, Requisitos x Casos de Uso, Casos de Uso x Casos de Uso)
- ✓ Revisões em planos e produtos de trabalho para identificar e corrigir inconsistências em relação aos requisitos
- ✓ Registro formal de solicitações de alteração e/ou inclusão de requisitos, incluindo análise de impacto para elas
- ✓ Definição de padrão de codificação para minimizar impactos de manutenção
- ✓ Realização de testes funcionais
- ✓ Realização de testes de aceitação do analista
- ✓ Realização de testes de aceitação dos usuários
- ✓ Utilização de prototipação
- ✓ Formalização de reuso e componentização
- ✓ Descrição de Casos de Uso
- ✓ Modelagem visual do projeto através de UML

De forma alguma o presente trabalho visa a esgotar todas as práticas de engenharia de software ou de desenvolvimento de projetos de software. A listagem apresentada apenas ilustra o conjunto de boas práticas consideradas no método de mapeamento proposto.

4.4.1 Relação das práticas com nível G do MPS.BR

Para tornar explícita a aderência das boas práticas selecionadas com os quesitos e resultados esperados no nível G no MPS.BR foi construída a Tabela 4.1 contendo a relação entre os GPRs desse nível e as práticas gerenciais que o atendem largamente ou totalmente e a Tabela 4.2 contendo a relação entre os GREs do mesmo nível e as práticas técnicas que o atendem largamente ou totalmente.

Tabela 4.1 – Relação entre GPRs do nível G e práticas selecionadas

| | |
|--|---|
| GPR 1. O escopo do trabalho para o projeto é definido; | <ul style="list-style-type: none"> Definição inicial de escopo (Documento de visão) Representação do escopo através de Estrutura Analítica do Projeto (EAP) |
| GPR 2. As tarefas e os produtos de trabalho do projeto são dimensionados utilizando métodos apropriados; | <ul style="list-style-type: none"> Dimensionamento do projeto sob o ponto de vista do usuário (Planilha de APF) |
| GPR 3. O modelo e as fases do ciclo de vida do projeto são definidos; | <ul style="list-style-type: none"> Definição de um modelo e as fases do ciclo de vida do projeto (Iterativo) |
| GPR 4. O esforço e o custo para a execução das tarefas e dos produtos de trabalho são estimados com base em dados históricos ou referências técnicas; | <ul style="list-style-type: none"> Dimensionamento de esforço e custo do projeto (Relação entre tamanho, produtividade e valor) Controle formal de orçamento do projeto (Planilha de orçamento) |
| GPR 5. O orçamento e o cronograma do projeto, incluindo a definição de marcos e pontos de controle, são estabelecidos e mantidos; | <ul style="list-style-type: none"> Controle formal de orçamento do projeto (Planilha de orçamento) Cronograma com dependências entre as atividades, marcos de controle e produtos de trabalho gerados |
| GPR 6. Os riscos do projeto são identificados e o seu impacto, probabilidade de ocorrência e prioridade de tratamento são determinados e documentados; | <ul style="list-style-type: none"> Identificação e monitoramento formal de riscos do projeto, com relação de impacto e probabilidade Gerência formal de riscos incluindo além de identificação, o planejamento de resposta aos riscos |
| GPR 7. Os recursos humanos para o projeto são planejados considerando o perfil e o conhecimento necessários para executá-lo; | <ul style="list-style-type: none"> Planejamento de recursos humanos para o projeto, determinando funções, responsabilidades, relações hierárquicas |
| GPR 8. Os recursos e o ambiente de trabalho necessários para executar o projeto são planejados; | <ul style="list-style-type: none"> Planejamento de recursos e ambiente necessários, incluindo, por exemplo, equipamentos, ferramentas, serviços, componentes, viagens, etc. |

| | |
|--|--|
| GPR 9. Os dados relevantes do projeto são identificados e planejados quanto à forma de coleta, armazenamento e distribuição. Um mecanismo é estabelecido para acessá-los, incluindo, se pertinente, questões de privacidade e segurança; | <ul style="list-style-type: none"> • Gerência de configuração do projeto, com identificação, coleta, armazenamento e distribuição das informações incluindo regras de segurança e confidencialidade |
| GPR 10. Um plano geral para a execução do projeto é estabelecido com a integração de planos específicos; | <ul style="list-style-type: none"> • Integração dos planejamentos realizados garantindo que a dependência entre eles seja monitorada |
| GPR 11. A viabilidade de atingir as metas do projeto, considerando as restrições e os recursos disponíveis, é avaliada. Se necessário, ajustes são realizados; | <ul style="list-style-type: none"> • Realização de estudos de viabilidade em marcos específicos para garantir que o projeto está sendo realizado de forma viável |
| GPR 12. O Plano do Projeto é revisado com todos os interessados e o compromisso com ele é obtido; | <ul style="list-style-type: none"> • Acompanhamento incluindo revisão e obtenção de compromisso dos planos do projeto com os interessados |
| GPR 13. O projeto é gerenciado utilizando-se o Plano do Projeto e outros planos que afetam o projeto e os resultados são documentados; | <ul style="list-style-type: none"> • Cronograma com dependências entre as atividades, marcos de controle e produtos de trabalho gerados • Acompanhamento incluindo revisão e obtenção de compromisso dos planos do projeto com os interessados |
| GPR 14. O envolvimento das partes interessadas no projeto é gerenciado; | <ul style="list-style-type: none"> • Comunicação formal das atividades do projeto com todos os interessados |
| GPR 15. Revisões são realizadas em marcos do projeto e conforme estabelecido no planejamento; | <ul style="list-style-type: none"> • Cronograma com dependências entre as atividades, marcos de controle e produtos de trabalho gerados |
| GPR 16. Registros de problemas identificados e o resultado da análise de questões pertinentes, incluindo dependências críticas, são estabelecidos e tratados com as partes interessadas; | <ul style="list-style-type: none"> • Identificação e monitoramento formal de problemas do projeto |
| GPR 17. Ações para corrigir desvios em relação ao planejado e para prevenir a repetição dos problemas identificados são estabelecidas implementadas e acompanhadas até a sua conclusão; | <ul style="list-style-type: none"> • Gerência formal de problemas incluindo além da identificação o registro de ações para correção de desvios |

Tabela 4.2– Relação entre GREs do nível G e práticas selecionadas

| | |
|--|--|
| GRE 1. Os requisitos são entendidos, avaliados e aceitos junto aos fornecedores de requisitos, utilizando critérios objetivos; | <ul style="list-style-type: none"> • Identificação formal dos requisitos (Lista de Requisitos funcionais e não funcionais, Estórias de Usuários) • Identificação formal dos fornecedores de requisitos do projeto • Definição de critérios objetivos para |
|--|--|

| | |
|--|---|
| | validação dos requisitos (Checklist de requisitos) <ul style="list-style-type: none"> • Descrição de Casos de Uso |
| GRE 2. O comprometimento da equipe técnica com os requisitos aprovados é obtido; | <ul style="list-style-type: none"> • Comunicação e comprometimento formal dos requisitos com a equipe técnica • Utilização de prototipação • Modelagem visual do projeto através de UML |
| GRE 3. A rastreabilidade bidirecional entre os requisitos e os produtos de trabalho é estabelecida e mantida; | <ul style="list-style-type: none"> • Rastreabilidade bidirecional entre os requisitos e os produtos de trabalho (Requisitos x Requisitos, Requisitos x Casos de Uso, Casos de Uso x Casos de Uso) |
| GRE 4. Revisões em planos e produtos de trabalho do projeto são realizadas visando identificar e corrigir inconsistências em relação aos requisitos; | <ul style="list-style-type: none"> • Revisões em planos e produtos de trabalho para identificar e corrigir inconsistências em relação aos requisitos • Realização de testes funcionais • Realização de testes de aceitação do analista • Realização de testes de aceitação dos usuários |
| GRE 5. Mudanças nos requisitos são gerenciadas ao longo do projeto. | <ul style="list-style-type: none"> • Registro formal de solicitações de alteração e/ou inclusão de requisitos, incluindo análise de impacto para elas • Definição de padrão de codificação para minimizar impactos de manutenção • Formalização de reuso e componentização |

No capítulo seguinte será apresentado o mapeamento propriamente dito entre os perfis de complexidade do ambiente e as práticas de engenharia de software propostas para os mesmos.

5 SELEÇÃO DE PRÁTICAS BASEADO EM PARÂMETROS DO PROBLEMA

5.1 Visão Geral

Considerando que duas organizações são diferentes entre si e, além disso, que dois projetos dentro de uma mesma organização podem também ter diferentes características, um processo que tenha sido bem sucedido em um projeto pode não ser adequado em outro projeto. Assim, um processo precisa ser definido de acordo com os requisitos e o contexto específico de um projeto em particular (Pedreira, 2007).

Dessa forma, não existe um processo de desenvolvimento ideal e universal. Ao invés disso, cada projeto de uma organização deveria ser conduzido por um processo de desenvolvimento especificamente modelado para o contexto próprio do projeto. Adaptação de processos de software (do inglês, *software process tailoring*) significa partir de um modelo de processo e adaptá-lo para aplicação em um projeto específico a ser conduzido em uma organização.

Deve-se destacar que o conceito de adaptação de processos é tratado pelo CMM (Paulk, 1994), nível 3, que requer que um conjunto de processos bem compreendido seja definido e que a cada projeto, a organização seja capaz de selecionar e adaptar um processo do conjunto, de acordo com as características do projeto. Assim, uma atividade que deve preceder a adaptação é a seleção (e recuperação), a partir do repositório da organização, do modelo de processo mais adequado (contexto mais similar às características do novo projeto) para servir como entrada a atividade de adaptação.

5.2 Relacionamento com as Primitivas do Processo de Software

(Royce, 1998), propõe um conjunto de Primitivas de Processo que são relacionadas aos discriminantes para customização de processo na mesma obra. As Primitivas de Processo são:

- i. **Fases do ciclo de vida** – referente à atenção demandada para modelagem e transição entre as fases do ciclo de vida do processo.
- ii. **Formalidade dos artefatos** – referente à formalidade e necessidade de artefatos e documentação formal.
- iii. **Alocação do esforço no processo** – referente à alocação de esforço requerido em cada etapa do processo.

iv. **Pontos formais de controle** – referente à formalidade de delimitação dos marcos formais de controle.

v. **Disciplina Gerencial** – referente à atenção necessária nos aspectos gerenciais do projeto.

vi. **Disciplina Técnica** – referente à atenção necessária nos aspectos técnicos do projeto.

O relacionamento das primitivas com os discriminantes para customização de processo proposto por (Royce, 1998), foi estendida para contemplar os sete fatores de definição de complexidade do ambiente propostos no trabalho. A principal contribuição observada nesse ponto é a inclusão considerações intermediárias ao relacionamento dicotômico descrito por Royce para cada primitiva. Tal relacionamento é a base do mapeamento do perfil do ambiente nos quadrantes de complexidade técnica e gerencial, bem como para a proposição de práticas de engenharia de software mais adequada para cada perfil.

As Tabelas 5.1, 5.2, 5.3, 5.4, 5.5 e 5.6 foram adaptadas de (Royce, 1998) e a Tabela 5.7 foi construída considerando elementos contidos na mesma obra. Nas tabelas em questão será utilizada a seguinte legenda para facilitar o entendimento:

- ✓ **Tópicos em negrito foram adicionados às tabelas;**
- ✓ *Tópicos em itálico foram pequenas alterações em relação às tabelas originais;*
- ✓ Tópicos com fonte normal foram mantidos inalterados.

Essas tabelas são apresentadas a seguir:

Tabela 5.1 – Relação das primitivas de processo com o fator Escala (adaptado de Royce, 1998)

| <i>Primitivas de Processo</i> | <i>Projetos Pequenos</i> | <i>Projetos Médios</i> | <i>Projetos Grandes</i> |
|----------------------------------|--|--|---|
| <i>Fases do ciclo de vida</i> | ✓ Fronteira entre as fases sem muita delimitação | ✓ Definição de fases num nível macro | ✓ Transições bem definidas entre as fases para sincronizar o progresso entre atividades concorrentes |
| <i>Formalidade dos artefatos</i> | <ul style="list-style-type: none"> ✓ Foco nos artefatos técnicos ✓ Baseline discreta ✓ Poucos artefatos gerenciais requeridos | <ul style="list-style-type: none"> ✓ Utilização moderada de artefatos técnicos ✓ Utilização moderada de artefatos gerenciais | <ul style="list-style-type: none"> ✓ Gerência de mudanças nos artefatos técnicos, o que pode gerar muitas baselines ✓ <i>Artefatos gerenciais necessários</i> |

| | | | |
|--|--|--|--|
| <i>Alocação do esforço no processo</i> | ✓ Maior necessidade de generalistas, pessoas que desempenham diversos papéis | ✓ Necessidade de equipes mescladas com generalistas e especialistas | ✓ Maior necessidade de especialistas ✓ Mais pessoas e equipes focadas em etapas do processo específicas |
| <i>Pontos de controle</i> | ✓ Poucos eventos informais para auxiliar a consistência técnica ✓ <i>Não precisa de cronograma muito definido</i> | ✓ Eventos formais para auxiliar a consistência técnica ✓ Cronograma com formalização de marcos gerenciais | ✓ <i>Eventos formais para garantir a consistência técnica</i> ✓ Sincronização entre as equipes que pode ser demorada |
| <i>Disciplina Gerencial</i> | ✓ Planejamento, controle e organização informais são suficientes | ✓ Planejamento formal ✓ Necessidade de alguma forma de acompanhamento | ✓ Planejamento, controle e organização formais são necessários |
| <i>Disciplina Técnica</i> | ✓ Ambiente <i>ad hoc</i> administrado pelos indivíduos | ✓ Organização do ambiente e estrutura de trabalho ✓ Ferramentas de versionamento requeridas | ✓ Estrutura para garantir ambientes consistentes e atualizados por todas as equipes ✓ Ferramenta de integração para auxiliar o controle do projeto e o controle de mudanças |

Tabela 5.2 – Relação das primitivas de processo com o fator Dinamismo (adaptado de Royce, 1998)

| <i>Primitivas de Processo</i> | <i>Projetos pouco dinâmicos</i> | <i>Projetos com dinamismo moderado</i> | <i>Projetos muito dinâmicos</i> |
|----------------------------------|--|---|--|
| <i>Fases do ciclo de vida</i> | ✓ Poucas fases formais ✓ Possibilidade de utilização das fases no modelo cascata | ✓ Recomendada utilização das fases no modelo iterativo ✓ Recomendada utilização de fases de prototipação | ✓ <i>Utilização das fases no modelo iterativo incremental</i> |
| <i>Formalidade dos artefatos</i> | ✓ Poucos artefatos gerenciais não muito detalhados ✓ Poucos artefatos técnicos de controle de requisitos | ✓ Alguns artefatos gerenciais ✓ Registro formal dos requisitos com flexibilidade de alteração | ✓ Artefatos gerenciais detalhados ✓ <i>Artefatos para gerenciamento de requisitos e de solicitação de mudança fortemente recomendados</i> |

| | | | |
|--|--|--|--|
| <i>Alocação do esforço no processo</i> | <ul style="list-style-type: none"> ✓ Menor retrabalho das estimativas ✓ Concentração da análise e projeto nas etapas iniciais | <ul style="list-style-type: none"> ✓ Esforço concentrado em prototipação ✓ Esforço razoável em análises de impacto | <ul style="list-style-type: none"> ✓ <i>Muito esforço demandado para estimativas</i> ✓ <i>Análises de impacto recorrentes</i> |
| <i>Pontos de controle</i> | <ul style="list-style-type: none"> ✓ Alguns eventos informais | <ul style="list-style-type: none"> ✓ Controle formal das alterações quando ocorrerem | <ul style="list-style-type: none"> ✓ <i>Validações formais em entregas periódicas</i> ✓ Revisões de planejamento formal para garantir o alinhamento |
| <i>Disciplina Gerencial</i> | <ul style="list-style-type: none"> ✓ Plano de projeto estável | <ul style="list-style-type: none"> ✓ Planejamento dinâmico ✓ Controle moderado da comunicação dos stakeholders | <ul style="list-style-type: none"> ✓ Plano de projeto dinâmico com escopo dinâmico ✓ Organização, controle e planejamento formal ✓ Controle de comunicação formal |
| <i>Disciplina Técnica</i> | <ul style="list-style-type: none"> ✓ Não aplicado | <ul style="list-style-type: none"> ✓ Controle de versão | <ul style="list-style-type: none"> ✓ Ambiente dos stakeholders on-line |

Tabela 5.3 - Relação das primitivas de processo com o fator Criticalidade/Flexibilidade (adaptado de Royce, 1998)

| <i>Primitivas de Processo</i> | <i>Projetos com baixa criticalidade e alta flexibilidade</i> | <i>Projetos criticalidade e flexibilidade moderada</i> | <i>Projetos com alta criticalidade e baixa flexibilidade</i> |
|--|--|---|--|
| <i>Fases do ciclo de vida</i> | <ul style="list-style-type: none"> ✓ Tolerância na etapa de comprometimento | <ul style="list-style-type: none"> ✓ Etapa inicial com compromisso moderado | <ul style="list-style-type: none"> ✓ Compromisso elevado com a etapa inicial de definição |
| <i>Formalidade dos artefatos</i> | <ul style="list-style-type: none"> ✓ Visão e caso de negócio alteráveis | <ul style="list-style-type: none"> ✓ Casos de negócio com controle moderado | <ul style="list-style-type: none"> ✓ Controle criterioso com documentos de visão e casos de negócio ✓ Aprovação formal de alteração dos artefatos |
| <i>Alocação do esforço no processo</i> | <ul style="list-style-type: none"> ✓ Não se aplica | <ul style="list-style-type: none"> ✓ Análise formal requerida | <ul style="list-style-type: none"> ✓ Elevado nível de esforço nos fluxos de gerência e estimativas |

| | | | |
|-----------------------------|--|---|--|
| Pontos de controle | ✓ Alguns pontos informais para manter a consistência técnica | ✓ Aprovação informal dos stakeholders | ✓ <i>Aprovação formal dos stakeholders dos requisitos</i> ✓ Testes formais com relatório de execução |
| Disciplina Gerencial | ✓ Planejamento e controle informais do projeto | ✓ Planejamento e controle moderados para o projeto | ✓ Grande atenção para planejamento e controle do projeto |
| Disciplina Técnica | ✓ Não se aplica | ✓ Não se aplica | ✓ Grande atenção ao desenvolvimento ✓ Revisões de código periódicas |

Tabela 5.4 - Relação das primitivas de processo com o fator Cultura/Maturidade em Processo
(adaptado de Royce, 1998)

| Primitivas de Processo | Equipes maduras em processos | Maturidade moderada da equipe em processos | Equipes imaturas em processos |
|--|--|--|---|
| Fases do ciclo de vida | ✓ Critérios bem definidos delimitando as fases | ✓ Fases estabelecidas sem formalismo | ✓ Não há delimitação entre as fases do ciclo de vida |
| Formalidade dos artefatos | ✓ Artefatos com formato, conteúdo e métodos de produção bem definidos | ✓ Construção moderada de artefatos com formato e conteúdo variáveis | ✓ Formulários livres |
| Alocação do esforço no processo | ✓ Não se aplica | ✓ Não se aplica | ✓ Não se aplica |
| Pontos de controle | ✓ Combinação bem definida entre eventos formais e informais | ✓ Respeito aos eventos formais de controle | ✓ Sem pontos de controle formais |
| Disciplina Gerencial | ✓ Planejamento previsível ✓ Avaliações de status objetivas | ✓ Planejamento respeitado | ✓ Planejamento e controle informais |
| Disciplina Técnica | ✓ Autos níveis de automação da engenharia, gerência de alterações e instrumentação do processo | ✓ Automação moderada do processo | ✓ Pequena automação ou automação em ilhas desconectadas |

Tabela 5.5 - Relação das primitivas de processo com o fator Previsibilidade Arquitetural (adaptado de Royce, 1998)

| <i>Primitivas de Processo</i> | <i>Projetos Alta previsibilidade</i> | <i>Projetos previsibilidade moderada</i> | <i>Projetos com baixa previsibilidade</i> |
|--|--|---|--|
| <i>Fases do ciclo de vida</i> | ✓ Mais iterações nas fases de iniciação e elaboração | ✓ Iterações divididas moderadamente nas fases de elaboração e construção | ✓ Poucas iterações iniciais ✓ Mais iterações de construção |
| <i>Formalidade dos artefatos</i> | ✓ <i>Artefatos técnicos construídos mais profundamente desde o início</i> | ✓ Não se aplica | ✓ Artefatos de fácil modificação recomendados |
| <i>Alocação do esforço no processo</i> | ✓ Concentração do esforço na etapa de projeto ✓ Menor esforço nas etapas de implementação e homologação | ✓ Esforço moderado nas etapas de projeto e implementação | ✓ <i>Concentração do esforço nas etapas de implementação e homologação, com grande nível de retrabalho</i> |
| <i>Pontos de controle</i> | ✓ Maior ênfase em demonstrações executáveis | ✓ Balanceamento entre demonstrações executáveis e simulações | ✓ <i>Maior ênfase nos guias, documentos e simulações</i> |
| <i>Disciplina Gerencial</i> | ✓ Pouca ênfase na gerência de alterações | ✓ Ênfase moderada na gerência de alterações | ✓ Grande ênfase na gerência de alterações |
| <i>Disciplina Técnica</i> | ✓ Mais recursos de ambiente requeridos mais cedo no ciclo de vida | ✓ Não se aplica | ✓ Menos recursos de ambiente requeridos no início do ciclo de vida |

Tabela 5.6 - Relação das primitivas de processo com o fator Experiência no Domínio

| <i>Primitivas de Processo</i> | <i>Projetos com alta experiência no domínio</i> | <i>Projetos com no domínio moderada</i> | <i>Projetos com baixa experiência no domínio</i> |
|----------------------------------|---|--|---|
| <i>Fases do ciclo de vida</i> | ✓ Curto estágio de engenharia | ✓ Estágios moderados de engenharia | ✓ Longos estágios de engenharia |
| <i>Formalidade dos artefatos</i> | ✓ Menos retrabalho nos artefatos de análise e projeto | ✓ Considerações e alterações moderadas nos artefatos de análise e projeto | ✓ Mais retrabalho e refinamentos nos artefatos de análise e projeto |
| <i>Alocação do</i> | ✓ Menor alocação | ✓ Alocação | ✓ Maior alocação nos |

| | | | |
|-----------------------------|--|--|---|
| <i>esforço no processo</i> | nos requisitos e projeto da solução | moderada nos requisitos e projeto da solução | requisitos e projeto da solução |
| <i>Pontos de controle</i> | ✓ Não se aplica | ✓ Não se aplica | ✓ Não se aplica |
| <i>Disciplina Gerencial</i> | ✓ Menor ênfase na gerência de risco ✓ Avaliações de status necessárias com baixa frequência | ✓ Avaliações periódicas de status ✓ Planejamento flexível recomendado | ✓ Maior ênfase na gerência de risco ✓ Avaliações de status necessárias com alta frequência |
| <i>Disciplina Técnica</i> | ✓ Não se aplica | ✓ Não se aplica | ✓ Não se aplica |

Tabela 5.7 - Relação das primitivas de processo com o fator Competência Pessoal

| <i>Primitivas de Processo</i> | <i>Equipes inexperientes</i> | <i>Equipes com experiência moderada</i> | <i>Equipes experientes</i> |
|--|--|--|---|
| <i>Fases do ciclo de vida</i> | ✓ Não se aplica | ✓ Não se aplica | ✓ Não se aplica |
| <i>Formalidade dos artefatos</i> | ✓ Necessidade de templates bem definidos ✓ Poucos artefatos formais | ✓ Artefatos básicos gerados com problemas de interpretação por alguns membros da equipe | ✓ Artefatos bem definidos e difundidos |
| <i>Alocação do esforço no processo</i> | ✓ Maior esforço nas etapas de análise e projeto | ✓ Esforço moderado em análise e projeto | ✓ Distribuído uniformemente em todas as etapas |
| <i>Pontos de controle</i> | ✓ Eventos informais | ✓ Alguns eventos formais | ✓ Acompanhamento de eventos formais |
| <i>Disciplina Gerencial</i> | ✓ Grande necessidade de gerência formal | ✓ Planejamento e controle moderados | ✓ Equipes auto-gerenciadas |
| <i>Disciplina Técnica</i> | ✓ Pouca utilização de ferramentas de engenharia | ✓ Utilização moderada de ferramentas de engenharia | ✓ Grande utilização de ferramentas de engenharia |

5.3 Mapeamento de Complexidade Técnica e Gerencial

5.3.1 Divisão dos Quadrantes de Complexidade

Os quatro quadrantes considerados por (Royce, 1998), descrevem regiões divididas em Complexidade Técnica e Complexidade Gerencial, como visto na Seção 2.3.2. O presente trabalho nomeia cada um dos quadrantes em função do perfil de Complexidade do Ambiente considerado em cada um.

Os quadrantes apresentados na Figura 5.1 são os seguintes:

- ✓ Complexidade do Ambiente Baixa
- ✓ Complexidade do Ambiente Média Gerencial
- ✓ Complexidade do Ambiente Média Técnica
- ✓ Complexidade do Ambiente Alta

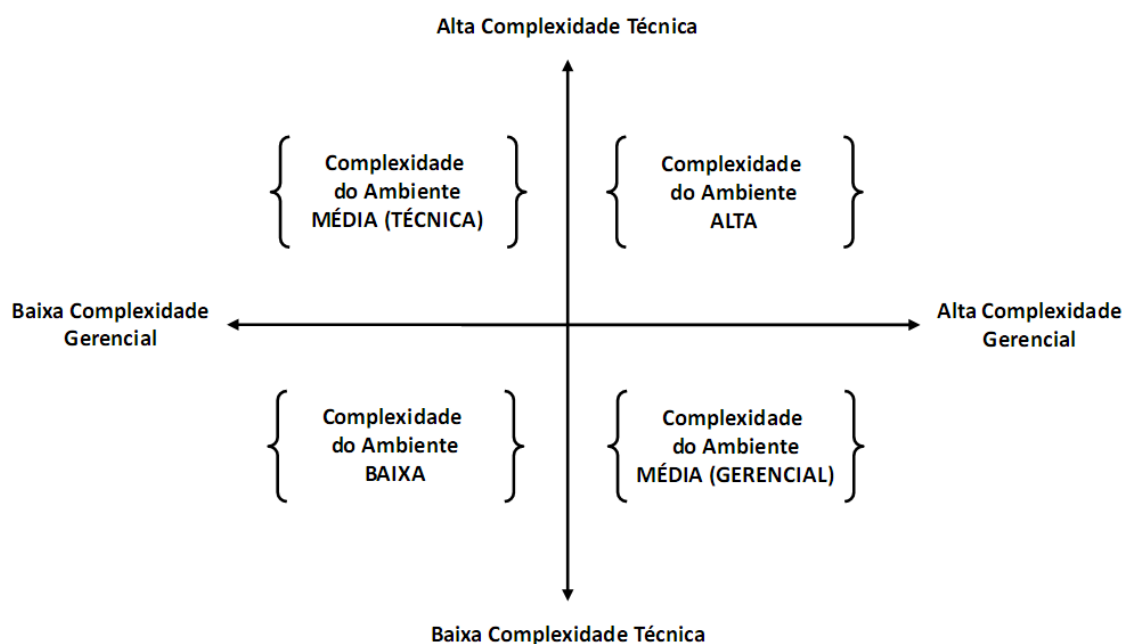


Figura 5.1 – Quadrantes de complexidade do ambiente

5.3.2 Posicionamento do Perfil de Complexidade nos Quadrantes

Como visto no Capítulo 3, foi proposto um perfil de complexidade do ambiente construído a partir da investigação de sete fatores: Escala, Dinamismo, Criticalidade / Flexibilidade, Previsibilidade Arquitetural, Experiência no Domínio, Cultura / Maturidade em Processo e Competência pessoal. É um gráfico radial contendo os sete fatores.

Cada eixo do gráfico possui 3 posições, “a”, “b”, e “c”, baixa, média e alta respectivamente. Para cada posição foi atribuído um peso:

- ✓ “a” - 1 (um)
- ✓ “b” - 2 (dois)
- ✓ “c” - 3 (três)

Assim a pontuação mínima para o perfil é 7 (sete) e a pontuação máxima para o perfil é 21 (vinte e um). E essa pontuação é utilizada para classificar o perfil de complexidade do ambiente estudado.

O total de pontos é dado da seguinte forma:

$$Total\ de\ Pontos = PEE + \sum_{i=1}^3 PEG_i + \sum_{i=1}^3 PET_i$$

Onde:

PEE são Pontos do Eixo Escala

PEG₁, PEG₂ e PEG₃ são Pontos dos Eixos Gerenciais

PET₁, PET₂ e PET₃ são os Pontos dos Eixos Técnicos

Tal classificação é dada por intervalos de pontos pela seguinte regra:

- ✓ Complexidade Baixa – Entre 7 (sete) e 10 (dez) pontos (Máximo 3 “b” ou 1 “c”)
- ✓ Complexidade Média – Entre 11 (onze) e 17 (dezesete) pontos (Máximo 5 “c”)
- ✓ Complexidade Alta – Entre 18 (dezoito) e 21 (vinte e um) pontos

Essa divisão de intervalos respeita a proporção de que para ser considerado como perfil de complexidade baixa o ambiente pode ter no máximo três dos sete fatores como médios ou um como alto, e para ser considerado médio pode ter no máximo cinco dos sete fatores como altos (desde que os outros dois sejam baixos), qualquer pontuação acima disso implica em um ambiente com complexidade alta.

A divisão do perfil de complexidade média entre Gerencial e Técnica é dado pela soma dos pontos dos eixos classificados como predominantemente técnicos e predominantemente gerenciais.

- ✓ Complexidade Média Gerencial – Soma dos fatores “gerenciais” é maior que a soma dos fatores “técnicos”

$$\sum_{i=1}^3 PEG_i > \sum_{i=1}^3 PET_i$$

- ✓ Complexidade Média Técnica – Soma dos fatores “técnicos” é maior que a soma dos fatores “gerenciais”

$$\sum_{i=1}^3 PEG_i < \sum_{i=1}^3 PET_i$$

Caso as somas dos fatores “gerenciais” e a soma dos fatores “técnicos” sejam iguais o perfil de complexidade do ambiente será classificado apenas como “Médio” tornando recomendada a aplicação tanto das práticas do quadrante de complexidade média gerencial quanto as práticas do quadrante de complexidade média gerencial.

- ✓ Complexidade Média (geral) – Soma dos fatores “técnicos” é igual a soma dos fatores “gerenciais”

$$\sum_{i=1}^3 PEGi = \sum_{i=1}^3 PETi$$

A Figura 5.2 apresenta os quadrantes de complexidade do ambiente divididos em termos das pontuações consideradas no trabalho.

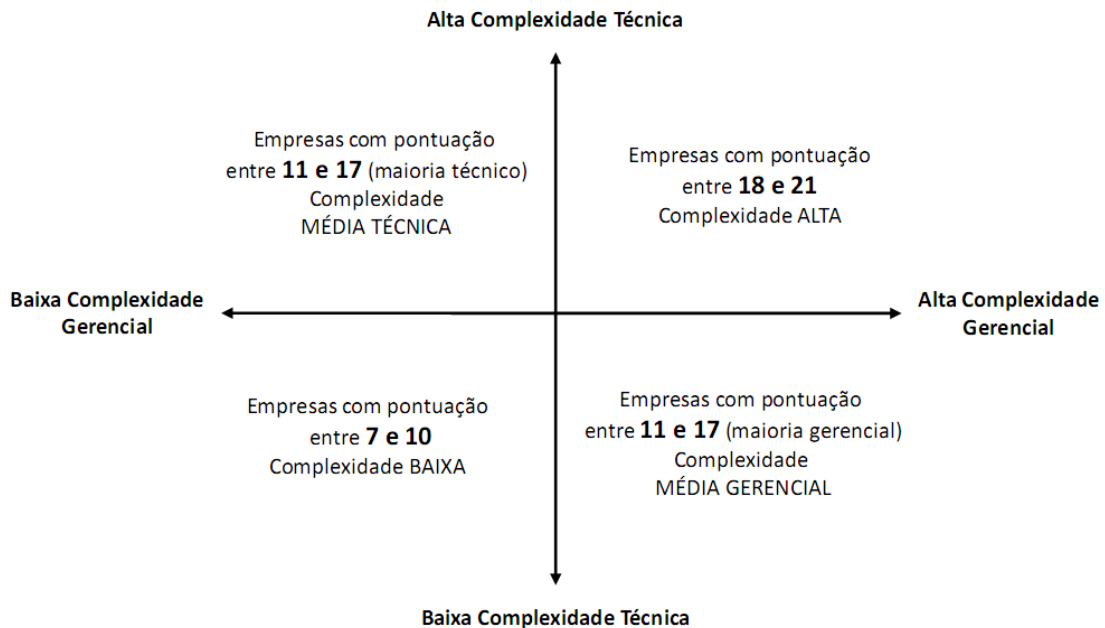


Figura 5.2 – Quadrantes de complexidade do ambiente em termos das pontuações

Para tornar explícita a relação de características relacionadas ao ambiente das empresas nos respectivos quadrantes são apresentadas as listas a seguir, construídas através da compilação das Tabelas 5.1 a 5.7.

Características de empresas com complexidade baixa do ambiente:

- ✓ Fronteira entre as fases sem muita delimitação
- ✓ Foco nos artefatos técnicos
- ✓ Maior necessidade de generalistas, pessoas que desempenham diversos papéis

- ✓ Poucos eventos informais para auxiliar a consistência técnica
- ✓ Sem necessidade de cronograma muito definido
- ✓ Ambiente *ad hoc* administrado pelos indivíduos
- ✓ Poucas fases formais
- ✓ Possibilidade de utilização das fases no modelo cascata
- ✓ Poucos artefatos gerenciais não muito detalhados
- ✓ Poucos artefatos técnicos de controle de requisitos
- ✓ Menor retrabalho das estimativas
- ✓ Concentração da análise e projeto nas etapas iniciais
- ✓ Plano de projeto estável
- ✓ Tolerância na etapa de comprometimento
- ✓ Visão e caso de negócio alteráveis
- ✓ Mais iterações nas fases de iniciação e elaboração
- ✓ Menor esforço nas etapas de implementação e homologação
- ✓ Maior ênfase em demonstrações executáveis
- ✓ Pouca ênfase na gerência de alterações
- ✓ Mais recursos de ambiente requeridos mais cedo no ciclo de vida
- ✓ Menor ênfase na gerência de risco
- ✓ Critérios bem definidos delimitando as fases
- ✓ Artefatos com formato, conteúdo e métodos de produção bem definidos
- ✓ Combinação bem definida entre eventos formais e informais
- ✓ Planejamento previsível
- ✓ Necessidade de templates bem definidos
- ✓ Poucos artefatos formais
- ✓ Pouca utilização de ferramentas de engenharia

Características das empresas com complexidade média gerencial do ambiente:

- ✓ Recomendada utilização das fases no modelo iterativo
- ✓ Recomendada utilização de fases de prototipação
- ✓ Alguns artefatos gerenciais
- ✓ Registro formal dos requisitos com flexibilidade de alteração
- ✓ Esforço razoável em análises de impacto
- ✓ Controle formal das alterações quando ocorrerem
- ✓ Planejamento dinâmico

- ✓ Controle moderado da comunicação dos stakeholders
- ✓ Controle de versão
- ✓ Etapa inicial com compromisso moderado
- ✓ Casos de negócio com controle moderado
- ✓ Análise formal requerida
- ✓ Aprovação informal dos stakeholders
- ✓ Planejamento e controle moderados para o projeto
- ✓ Fases estabelecidas sem formalismo
- ✓ Construção moderada de artefatos com formato e conteúdo variáveis
- ✓ Respeito aos eventos formais de controle
- ✓ Planejamento respeitado
- ✓ Automação moderada do processo

Características das empresas com complexidade média técnica do ambiente:

- ✓ Iterações divididas moderadamente nas fases de elaboração e construção
- ✓ Esforço moderado nas etapas de projeto e implementação
- ✓ Balanceamento entre demonstrações executáveis e simulações
- ✓ Ênfase moderada na gerência de alterações
- ✓ Estágios moderados de engenharia
- ✓ Considerações e alterações moderadas nos artefatos de análise e projeto
- ✓ Alocação moderada nos requisitos e projeto da solução
- ✓ Avaliações periódicas de status
- ✓ Planejamento flexível recomendado
- ✓ Artefatos básicos gerados com problemas de interpretação por alguns membros da equipe
- ✓ Esforço moderado em análise e projeto
- ✓ Alguns eventos formais
- ✓ Planejamento e controle moderados
- ✓ Utilização moderada de ferramentas de engenharia

Características das empresas com complexidade alta do ambiente:

- ✓ Transições bem definidas entre as fases para sincronizar o progresso entre atividades concorrentes
- ✓ Artefatos gerenciais necessários
- ✓ Maior necessidade de especialistas

- ✓ Mais pessoas e equipes focadas em etapas do processo específicas
- ✓ Eventos formais para garantir a consistência técnica
- ✓ Planejamento, controle e organização formais são necessários
- ✓ Estrutura para garantir ambientes consistentes e atualizados por todas as equipes
- ✓ Ferramenta de integração para auxiliar o controle do projeto e o controle de mudanças
- ✓ Utilização das fases no modelo iterativo incremental
- ✓ Artefatos gerenciais detalhados
- ✓ Artefatos para gerenciamento de requisitos e de solicitação de mudança fortemente recomendados
- ✓ Muito esforço demandado para estimativas
- ✓ Análises de impacto recorrentes
- ✓ Validações formais em entregas periódicas
- ✓ Plano de projeto dinâmico com escopo dinâmico
- ✓ Controle de comunicação formal
- ✓ Compromisso elevado com a etapa inicial de definição
- ✓ Controle criterioso com documentos de visão e casos de negócio
- ✓ Aprovação formal de alteração dos artefatos
- ✓ Aprovação formal dos stakeholders dos requisitos
- ✓ Testes formais com relatório de execução
- ✓ Grande atenção ao desenvolvimento
- ✓ Revisões de código periódicas
- ✓ Maior ênfase nos guias, documentos e simulações
- ✓ Grande ênfase na gerência de alterações
- ✓ Menos recursos de ambiente requeridos no início do ciclo de vida
- ✓ Maior alocação nos requisitos e projeto da solução
- ✓ Maior ênfase na gerência de risco
- ✓ Grande utilização de ferramentas de engenharia

5.4 Proposição do Conjunto de Práticas

O conjunto de práticas foi distribuído nos quadrantes de complexidade do ambiente através do relacionamento dos mesmos com as **primitivas de processo** apresentadas na Seção 5.1 e a **lista de características das empresas de cada quadrante** apresentada na Seção 5.3.2.

As boas práticas listadas, como descrito anteriormente, são independentes entre si, contudo, a proposição das mesmas deve ter caráter cumulativo entre os quadrantes. Assim, as práticas tanto gerenciais como técnicas relacionadas para o quadrante de Complexidade do Ambiente Baixa, serão práticas básicas propostas para MPEs que se aplicam a todos os casos. Isso implica que para ambiente com perfil de Complexidade do Ambiente Média (técnica ou gerencial) sempre serão propostas tanto práticas gerenciais e técnicas. E para o perfil de Complexidade do Ambiente Alta são propostas todas as boas práticas relacionadas.

Como tratado na seção anterior os quadrantes de Complexidade do Ambiente Média, estão divididos em dois: Complexidade Média Técnica e Complexidade Média Gerencial, o que não impede que as práticas do quadrante oposto sejam propostas, a divisão se dá principalmente na consideração de prioridade de implantação das práticas.

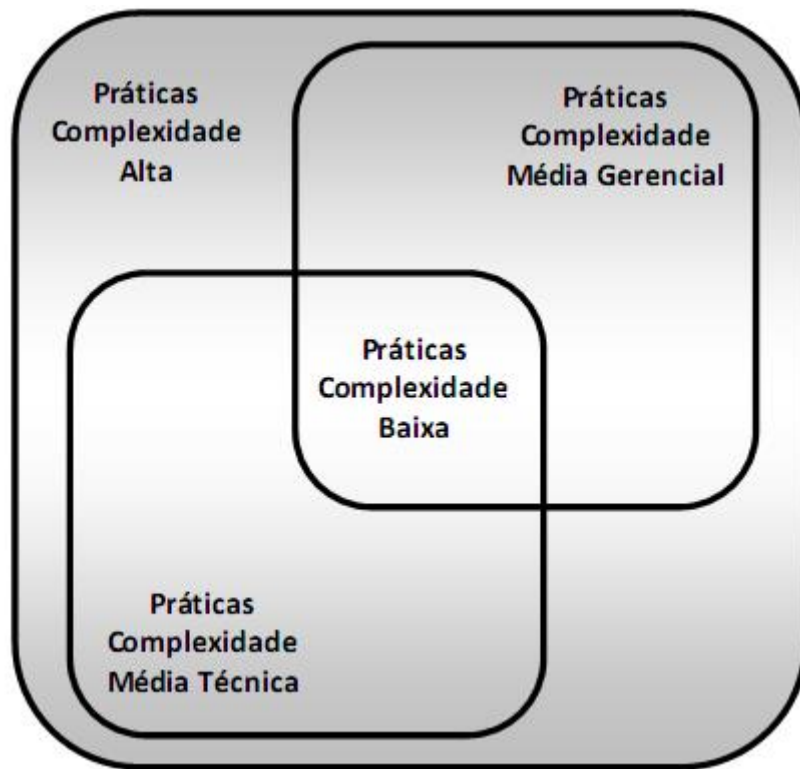


Figura 5.3 – Organização do conjunto de práticas propostas para cada perfil identificado

5.4.1 Práticas para Complexidade Baixa do Ambiente

Nas Tabelas 5.8 e 5.9 é apresentado o conjunto de boas práticas propostas para o quadrante de complexidade baixa do ambiente e a(s) respectiva(s) justificativa(s) para a escolha, conforme as características das empresas em cada quadrante listadas na Seção 5.3.2.

Tabela 5.8 – Práticas gerenciais para o quadrante de complexidade baixa do ambiente

| Boas práticas propostas | Justificativas |
|--|---|
| ✓ Definição inicial de escopo (Documento de visão) | A definição do escopo do projeto é muito importante em qualquer contexto, mesmo em ambientes com características de planejamento previsíveis. |
| ✓ Cronograma simples com datas e responsáveis pelas atividades | Empresas relacionadas nesse quadrante não apresentam necessidade de artefatos gerenciais muito detalhados, contudo a definição de um cronograma com informações mínimas de datas e responsáveis pelas atividades é uma prática gerencial básica a ser adotada. |
| ✓ Planejamento de recursos humanos para o projeto, determinando funções, responsabilidades, relações hierárquicas | Mesmo em ambientes de baixa complexidade, e com planejamento geral muito previsível, uma prática gerencial indicada é a formalização do planejamento de recursos humanos, pois se trata de uma premissa básica de qualquer projeto que o seu plano preveja os recursos básicos necessários. |
| ✓ Planejamento de recursos e ambiente necessários, incluindo, por exemplo, equipamentos, ferramentas, serviços, componentes, viagens, etc. | Pelo mesmo motivo do planejamento de recursos humanos citado anteriormente, o planejamento de recursos gerais e de ambiente também é uma prática gerencial pertinente independente do contexto de complexidade do projeto. |
| ✓ Acompanhamento incluindo revisão e obtenção de compromisso dos planos do projeto com os interessados | Projetos situados em ambiente de baixa complexidade tendem a necessitar de poucas alterações ao longo do ciclo de vida, contudo, acompanhamento e revisão do plano traçado para o projeto e a obtenção de compromisso com o plano e escopo do mesmo é uma prática gerencial de grande valia em qualquer contexto. |

Tabela 5.9 – Práticas técnicas para o quadrante de complexidade baixa do ambiente

| Boas práticas propostas | Justificativas |
|--|---|
| ✓ Identificação formal dos requisitos (Lista de Requisitos funcionais e não funcionais, Estórias de Usuários) | A identificação e listagem formal dos requisitos é uma boa prática que se aplica independente do nível de complexidade. No caso de empresas com baixa complexidade do ambiente, o foco tende a ser nos artefatos técnicos e o documento de requisitos formal deve ser um dos artefatos centrais nesse contexto. |
| ✓ Registro formal de solicitações de alteração e/ou inclusão de requisitos, incluindo análise de impacto para elas | Empresas desse perfil tendem a ter poucas alterações de requisitos, contudo, a formalização das solicitações é uma boa prática para administrar corretamente o escopo do projeto |

| | |
|--|---|
| | através da análise de impacto das inclusões e/ou modificações do mesmo. |
| ✓ Definição de padrão de codificação para minimizar impactos de manutenção | Uma prática técnica muito útil no contexto de empresas com complexidade baixa é a definição organizacional de um padrão de codificação. Esse ambiente tende a ter grande número de profissionais “generalistas” o que implica que diversas pessoas podem trabalhar diretamente na codificação, e um padrão bem estabelecido facilita o entendimento e manutenção do código. |
| ✓ Realização de testes funcionais | Formalização de uma etapa de testes funcionais é uma importante forma de institucionalizar políticas iniciais de qualidade, e pelo mesmo motivo citado para definição do padrão de codificação, testes bem realizados minimizam impactos negativos relacionados a diversas pessoas trabalhando num mesmo código. |
| ✓ Utilização de prototipação | Como dentro desse perfil o foco tende a ser no desenvolvimento propriamente dito, para auxiliar a etapa de análise realizada, uma boa prática é a utilização de prototipação, seja para obtenção de entendimento do cliente, seja para facilitar a comunicação interna dos profissionais envolvidos. |

5.4.2 Práticas para Complexidade Média Gerencial do Ambiente

Na Tabela 5.10 é apresentado o conjunto de boas práticas propostas para o quadrante de complexidade média gerencial do ambiente e a(s) respectiva(s) justificativa(s) para a escolha, conforme as características das empresas em cada quadrante listadas na Seção 5.3.2.

Além das práticas listadas na Tabela 5.10, as práticas gerenciais e técnicas propostas para o quadrante de complexidade baixa do ambiente, devem também ser consideradas para esse quadrante.

Tabela 5.10 – Práticas gerenciais para o quadrante de complexidade média gerencial do ambiente

| Boas práticas propostas | Justificativas |
|--|---|
| ✓ Representação do escopo através de Estrutura Analítica do Projeto (EAP) | O planejamento do projeto em empresas com esse perfil de complexidade tendem a ter um dinamismo moderado, e a representação visual do escopo através da EAP, auxilia a manutenção do escopo do projeto. |
| ✓ Dimensionamento do projeto sob o ponto de vista do usuário (Planilha de APF) | A partir do perfil de complexidade média as empresas já devem possuir um rigor de |

| | |
|--|---|
| | planejamento dos projetos e uma boa prática é estimar o tamanho do projeto de forma quantitativa, e análise de pontos de função ou similar torna-se importante. |
| ✓ Definição de um modelo e as fases do ciclo de vida do projeto (Iterativo) | Para gerenciar projetos com complexidade gerencial moderada torna-se importante um mínimo de definição com relação ao ciclo de vida do projeto. Nesse caso, sugere-se mais especificamente algum tipo de ciclo de vida iterativo que tende a minimizar riscos em ambientes dinâmicos. |
| ✓ Cronograma com dependências entre as atividades, marcos de controle e produtos de trabalho gerados | Para que possa haver um controle mais direto e um grau satisfatório de acompanhamento dos projetos com esse perfil de complexidade, sugere-se a extensão do cronograma proposto para ambientes de complexidade baixa, para contemplar além das atividades e responsáveis, os marcos de controle e produtos de trabalho gerados. |
| ✓ Identificação e monitoramento formal de riscos do projeto, com relação de impacto e probabilidade | À medida que os projetos vão caminhando para maior complexidade gerencial, os riscos envolvidos no projeto devem ser tratados de maneira formal. Para esse nível de complexidade do ambiente, sugere-se a identificação e monitoramento dos riscos. |
| ✓ Controle formal de orçamento do projeto (Planilha de orçamento) | Devido ao dinamismo envolvido no planejamento de projetos com esse grau de complexidade, controle formal do orçamento de projeto é fortemente recomendado, impedindo que a viabilidade do projeto seja violada. |
| ✓ Integração dos planejamentos realizados garantindo que a dependência entre eles seja monitorada | Empresas com esse perfil de complexidade exigem um planejamento mais formal como descrito anteriormente nas práticas sugeridas. Uma prática recomendada nesse contexto é a integração dos planos, impedindo que eles fiquem inconsistentes entre eles ou que a atenção seja concentrada em apenas um fator do plano, avaliando sempre em conjunto: prazos, riscos, orçamento e etc. |
| ✓ Comunicação formal das atividades do projeto com todos os interessados | Para fortalecer o monitoramento dos planos do projeto, bem como para obter forte comprometimento dos envolvidos, sugere-se nesse nível de complexidade que os interessados sejam comunicados formalmente sobre todas as atividades do projeto, através de atas, termos ou outra forma de comunicação. |
| ✓ Identificação e monitoramento formal de problemas do projeto | Problemas encontrados tanto no planejamento como na execução das atividades devem ser formalmente identificados e monitorados para tentar diminuir a reincidência dos mesmos. |

5.4.3 Práticas para Complexidade Média Técnica do Ambiente

Na Tabela 5.11 é apresentado o conjunto de boas práticas propostas para o quadrante de complexidade média técnica do ambiente e a(s) respectiva(s) justificativa(s) para a escolha, conforme as características das empresas em cada quadrante listadas na Seção 5.3.2.

Além das práticas listadas na Tabela 5.11, as práticas gerenciais e técnicas propostas para o quadrante de complexidade baixa do ambiente, devem também ser consideradas para esse quadrante.

Tabela 5.11– Práticas técnicas para o quadrante de complexidade média técnica do ambiente

| Boas práticas propostas | Justificativas |
|--|--|
| ✓ Identificação formal dos fornecedores de requisitos do projeto | A identificação formal dos fornecedores de requisito tendem a diminuir o retrabalho de análise gerado pela falta de coesão entre os representantes do cliente. |
| ✓ Comunicação e comprometimento formal dos requisitos com a equipe técnica | Obter comprometimento formal entre a equipe técnica e os requisitos auxilia o acompanhamento do projeto, não permitindo que funcionalidades sem a devida análise sejam incorporadas ao projeto. |
| ✓ Rastreabilidade bidirecional entre os requisitos e os produtos de trabalho (Requisitos x Requisitos, Requisitos x Casos de Uso, Casos de Uso x Casos de Uso) | Empresas com o perfil de complexidade média técnica, devem possuir forte rastreabilidade em todas as direções do projeto, e uma maneira de conseguir essa característica é definir formalmente através de uma matriz (ou através de ferramenta dedicada) o relacionamento entre os requisitos e produtos de trabalho. Essa é uma maneira de tornar possível avaliar o impacto de alterações que por ventura surjam no projeto. |
| ✓ Realização de testes de aceitação dos usuários | A partir de certo grau de complexidade técnica relacionado ao contexto dos projetos, tornam-se importantes validações periódicas dos usuários do sistema. Esses testes de aceitação servem para monitorar a aderência entre o entendimento da equipe de desenvolvimento do projeto e a real necessidade do usuário final. |
| ✓ Descrição de Casos de Uso | Artefatos técnicos mais detalhados, através da transformação de requisitos em casos de uso são importantes dentro desse contexto, para aumentar o entendimento e uniformidade dos envolvidos em torno do problema. |
| ✓ Modelagem visual do projeto através de UML | A comunicação visual de projetos com complexidade técnica moderada é mais uma maneira de facilitar o entendimento do problema pelas partes envolvidas. |

5.4.4 Práticas para Complexidade Alta do Ambiente

Nas Tabelas 5.12 e 5.13 é apresentado o conjunto de boas práticas propostas para o quadrante de complexidade alta do ambiente e a(s) respectiva(s) justificativa(s) para a escolha, conforme as características das empresas em cada quadrante listadas na Seção 5.3.2.

Além das práticas listadas nas Tabelas 5.12 e 5.13, as práticas gerenciais e técnicas propostas para os quadrantes de complexidade baixa, média gerencial e média técnica do ambiente, devem também ser consideradas para esse quadrante.

Tabela 5.12 – Práticas gerenciais para o quadrante de complexidade alta do ambiente

| Boas práticas propostas | Justificativas |
|--|---|
| ✓ Dimensionamento de esforço e custo do projeto (Relação entre tamanho, produtividade e valor) | Uma prática que é importante em ambientes muito complexos é a definição de relação entre esforço e custo, para poder obter planejamentos mais precisos. |
| ✓ Gerência formal de riscos incluindo além de identificação, o planejamento de resposta aos riscos | Além da identificação dos riscos proposta para complexidade gerencial média, nesse contexto o planejamento de resposta a eles também torna-se muito importante, uma vez que projetos desse contexto possuem alto grau de criticalidade. |
| ✓ Gerência de configuração do projeto, com identificação, coleta, armazenamento e distribuição das informações incluindo regras de segurança e confidencialidade | Projetos muito complexos exigem planejamento rigoroso em todas as suas nuances, e uma gerência de configuração madura é importante para garantir quesitos relacionados, sobretudo, a segurança. |
| ✓ Realização de estudos de viabilidade em marcos específicos para garantir que o projeto está sendo realizado de forma viável | Projetos com alto grau de complexidade, devem possuir marcos para avaliação da viabilidade, permitindo que alterações nesse sentido sejam discutidos tão logo sejam identificados. |
| ✓ Gerência formal de problemas incluindo além da identificação o registro de ações para correção de desvios | Assim como a gerência formal dos riscos, a gerência formal dos problemas sugere que projetos nesse nível devem estar preparados para responder a qualquer eventual problema. |

Tabela 5.13 – Práticas técnicas para o quadrante de complexidade alta do ambiente

| Boas práticas propostas | Justificativas |
|--|--|
| ✓ Definição de critérios objetivos para validação dos requisitos (Checklist de requisitos) | Assim como o rigor gerencial estabelecido para projetos nesse contexto, o rigor técnico do mesmo deve ser avaliado baseado em critérios objetivos. |
| ✓ Revisões em planos e produtos de trabalho para identificar e corrigir inconsistências em | Tão logo sejam identificados problemas de inconsistência entre produtos de trabalho e |

| | |
|---|---|
| relação aos requisitos | requisitos os mesmos devem ser ajustados, assim revisões periódicas são importantes. |
| ✓ Realização de testes de aceitação do analista | Para esse contexto, sugere-se mais um nível de validação entre os testes funcionais e os testes de aceitação dos usuários, que são os testes de aceitação do analista que projetou a solução para garantir que o que foi implementado é exatamente o que foi projetado. |
| ✓ Formalização de reuso e componentização | A reutilização e componentização realizadas com maturidade auxiliam muito a validação, uma vez que os componentes reutilizados tendem a terem sido exhaustivamente testados em projetos anteriores. |

6 ESTUDO DE CASO

Foram realizados estudos de casos para comprovar na prática a proposta deste trabalho. A metodologia proposta foi aplicada em duas empresas diferentes obtendo-se o perfil de complexidade do ambiente de desenvolvimento inerente a cada empresa.

As empresas analisadas pertencem ao Arranjo Produtivo Local de Tecnologia da Informação de Viçosa – MG (APL-TI Viçosa). Elas apresentam similaridade entre si. São equipes pequenas onde o usuário final do sistema tem um perfil especializado, assim como o próprio desenvolvedor. O software desenvolvido visa resolver problemas bem definidos e específicos, geralmente ligados a objetivos acadêmicos ou de pesquisa, não envolvendo múltiplos papéis de usuários externos. Parte dos desenvolvedores não tem como área de atuação principal a Ciência da Computação, mas têm formação especializada em suas respectivas áreas.

O desenvolvimento de software para esse perfil de problema requer um processo mais adaptado às suas características específicas, que escapam ao convencional das demais áreas, notadamente das áreas organizacionais.

Os estudos de casos são detalhados nas próximas seções. As respostas obtidas através da aplicação dos questionários gerencial e individual estão disponíveis no Apêndice B.

6.1 Formato de Apresentação dos Estudos de Caso

Para padronizar a exposição dos resultados, cada estudo de caso será descrito através do seguinte formato:

- ✓ **Nome:** nome da empresa.
- ✓ **Descrição:** breve descrição da equipe.
- ✓ **Composição da equipe:** informações sobre a composição da equipe, o perfil de seus integrantes e o papel de cada um deles dentro da equipe.
- ✓ **Aplicação do questionário:** informações sobre a aplicação do questionário.
- ✓ **Tabela de classificação:** classificação simplificada da equipe em relação aos sete fatores de complexidade do ambiente. Realizada a partir dos dados coletados e das interpretações associadas ao questionário descritas na Seção 3.3.
- ✓ **Gráfico Radial:** representação gráfica do perfil de complexidade do ambiente, obtido através dos dados coletados e das interpretações descritas na Seção 3.3.

- ✓ **Pontuação e mapeamento:** Quantidade de pontos computada para o perfil de complexidade do ambiente traçado no gráfico radial. E quadrante mapeado para empresa.
- ✓ **Listagem de práticas de Engenharia de Software propostas:** sugestões a serem consideradas na customização do processo para a equipe analisada.
- ✓ **Observações:** informações complementares que enriqueçam o estudo de caso (quando existirem).

6.2 Empresa 1

- ✓ **Nome:** Cientec – Consultoria e Desenvolvimento de Sistema Ltda.
- ✓ **Descrição:** A Cientec é uma empresa graduada pela Incubadora de Empresas de Base Tecnológica CENTEV/UFV e atua na área de consultoria e desenvolvimento de softwares. Estando inserida no contexto da Universidade Federal de Viçosa - UFV, um pólo de referência nacional de pesquisas científicas e desenvolvimento tecnológico, a Cientec tem como missão oferecer tecnologias inovadoras e desenvolver softwares de alto conteúdo científico e tecnológico, visando adquirir, aplicar e difundir conhecimentos, promover o crescimento e superar as expectativas dos clientes, da equipe e dos parceiros, através da excelência na qualidade dos produtos e serviços prestados.
- ✓ **Composição da equipe:** A equipe de desenvolvimento é formada por oito pessoas, sendo que algumas delas são substituídas ocasionalmente. Os integrantes da equipe estão classificados nos seguintes papéis: um gerente de projetos, dois analistas de sistemas, três desenvolvedores, um homologador e um analista de negócio (engenheiro florestal).
- ✓ **Aplicação do questionário:** Foi realizado um contato prévio com a equipe através de reuniões e conversas informais. Esse contato foi realizado antes da aplicação dos questionários e teve o objetivo de esclarecer o papel de cada indivíduo dentro da equipe e identificar qual membro estava mais apto a responder o questionário gerencial. O questionário gerencial foi respondido pelo Gerente de Projetos. Ele foi o selecionado por pertencer à equipe há mais tempo e concentrar as atividades gerenciais. O questionário individual foi respondido por todos os membros, inclusive pelo respondente do questionário gerencial, pois este profissional também desempenha atividades ligadas diretamente ao desenvolvimento.

✓ **Tabela de classificação:**

Tabela 6.1 – Classificação da empresa 1 nos sete fatores de complexidade do ambiente

| | |
|---|--|
| Escala | Até 10 profissionais, Até 50 casos de uso, Até 700 pontos de função |
| Dinamismo | Até 15% de alterações por mês |
| Criticalidade / Flexibilidade | Risco de prejuízo financeiro e rigor contratual moderado |
| Cultura / Maturidade em Processo | Características organizacionais moderadas para trabalhar de forma organizada |
| Previsibilidade Arquitetural | Conhecimento prévio pobre da arquitetura necessária |
| Experiência no Domínio | Especialista em desenvolvimento de uma determinada área |
| Competência Pessoal | Equipe inexperiente para trabalhar orientada pela organização formal |

✓ **Gráfico Radial:**

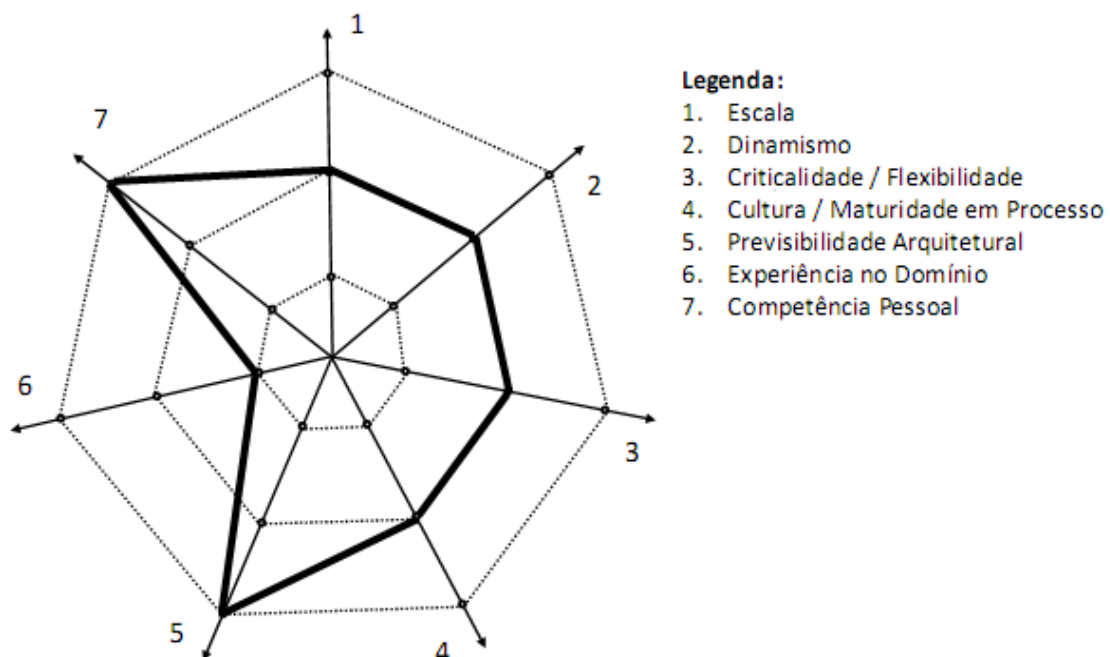


Figura 6.1 – Gráfico radial traçado com o perfil de complexidade do ambiente da empresa 1

✓ **Pontuação e mapeamento:**

Tabela 6.2 – Pontuação do perfil de complexidade do ambiente da empresa 1

| Eixo | Valor |
|-------------------------------------|--------------|
| 1. Escala | 2 |
| 2. Dinamismo | 2 |
| 3. Criticalidade / Flexibilidade | 2 |
| 4. Cultura / Maturidade em Processo | 2 |
| 5. Previsibilidade Arquitetural | 3 |
| 6. Experiência no Domínio | 1 |
| 7. Competência Pessoal | 3 |
| Total Geral | 15 |

O perfil de complexidade do ambiente para empresa 1 é classificado como médio por obter pontuação entre 11 e 17 no total geral.

A empresa foi mapeada no quadrante de **Complexidade Média Técnica** devido à pontuação dos eixos com características predominantemente técnicas (eixos 4, 5 e 7) ser maior que a pontuação dos eixos com características predominantemente gerenciais (eixos 2, 3 e 6).

✓ **Listagem de práticas de Engenharia de Software propostas:**

Blocos de Práticas Gerenciais

- Definição inicial de escopo (Documento de visão)
- Cronograma simples com datas e responsáveis pelas atividades
- Planejamento de recursos humanos para o projeto, determinando funções, responsabilidades, relações hierárquicas
- Planejamento de recursos e ambiente necessários, incluindo, por exemplo, equipamentos, ferramentas, serviços, componentes, viagens, etc.
- Acompanhamento incluindo revisão e obtenção de compromisso dos planos do projeto com os interessados

Blocos de Práticas Técnicas

- Identificação formal dos requisitos (Lista de Requisitos funcionais e não funcionais, Estórias de Usuários)
- Registro formal de solicitações de alteração e/ou inclusão de requisitos, incluindo análise de impacto para elas
- Definição de padrão de codificação para minimizar impactos de manutenção

- Realização de testes funcionais
- Utilização de prototipação
- Identificação formal dos fornecedores de requisitos do projeto
- Comunicação e comprometimento formal dos requisitos com a equipe técnica
- Rastreabilidade bidirecional entre os requisitos e os produtos de trabalho (Requisitos x Requisitos, Requisitos x Casos de Uso, Casos de Uso x Casos de Uso)
- Realização de testes de aceitação dos usuários
- Descrição de Casos de Uso
- Diagramas de Casos de Uso
- Modelagem visual do projeto através de UML
- ✓ **Observações:** A empresa possui grande experiência em desenvolvimento de software no setor de Agronegócio.

6.3 Empresa 2

- ✓ **Nome:** Jungle Digital Games.
- ✓ **Descrição:** A Jungle Digital Games é uma empresa inserida na Incubadora de Base Tecnológica da Universidade Federal de Viçosa que atua no setor de entretenimento educacional. Localizada em um pólo de tecnologia da informação e inserida no contexto de uma das mais renomadas universidades brasileiras, a Jungle vêm atuando desde 2007 no desenvolvimento de objetos digitais educacionais. Seus produtos e serviços, em especial os jogos digitais educacionais destacam-se como ferramentas complementares ao processo de ensino-aprendizagem trazendo melhoria à qualidade de ensino das instituições e contribuindo assim para o desenvolvimento dos indivíduos e por conseqüência da sociedade. Focada no desenvolvimento de jogos educacionais digitais, a empresa espera atender as aspirações de estudantes e professores no que tange à ferramenta de aprendizagem e aspectos motivacionais utilizados em sala de aula.
- ✓ **Composição da equipe:** O setor produtivo da Jungle é formada por 9 (nove) pessoas que se dividem entre a produção artística e a programação propriamente dita. Os integrantes da equipe estão classificados nos seguintes papéis: um gerente de produção, um gerente de projetos, um analista de sistema, um programador, um desenhista, um designer gráfico, um modelador 3D, e dois analistas de negócios (pedagogos).

✓ **Aplicação do questionário:** Assim como na empresa 1, foi realizado um contato prévio com a equipe através de reuniões e conversas informais. Esse contato foi realizado antes da aplicação dos questionários e teve o objetivo de esclarecer o papel de cada indivíduo dentro da equipe e identificar qual membro estava mais apto a responder o questionário gerencial. O questionário gerencial foi respondido pelo Diretor de Projetos. Ele foi o selecionado por ser um dos sócios fundadores e concentrar as atividades de gestão. O questionário individual foi respondido por todos os membros, inclusive pelo respondente do questionário gerencial, pois este profissional também desempenha atividades ligadas diretamente ao desenvolvimento.

✓ **Tabela de classificação:**

Tabela 6.3 – Classificação da empresa 2 nos sete fatores de complexidade do ambiente

| | |
|---|---|
| Escala | Até 3 profissionais, Até 20 casos de uso, Até 200 pontos de função |
| Dinamismo | Até 15% de alterações por mês |
| Criticalidade / Flexibilidade | Risco de prejuízo financeiro e rigor contratual moderado |
| Cultura / Maturidade em Processo | Características moderadas para trabalhar de forma organizada |
| Previsibilidade Arquitetural | Conhecimento prévio moderado da arquitetura necessária |
| Experiência no Domínio | Especialista em desenvolvimento de uma determinada área |
| Competência Pessoal | Equipe com experiência moderada, com capacidade para adequar o trabalho a situações com precedentes |

✓ **Gráfico Radial:**

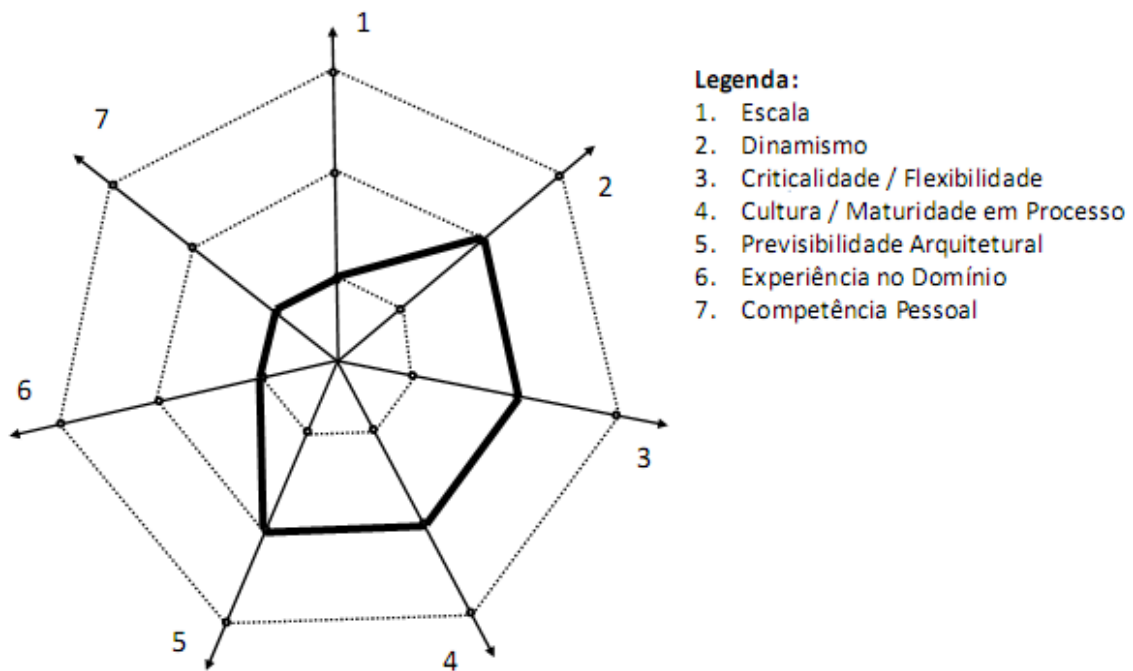


Figura 6.2 – Gráfico radial traçado com o perfil de complexidade do ambiente da empresa 1

✓ **Pontuação e mapeamento:**

Tabela 6.4 – Pontuação do perfil de complexidade do ambiente da empresa 2

| Eixo | Valor |
|-------------------------------------|--------------|
| 1. Escala | 1 |
| 2. Dinamismo | 2 |
| 3. Criticalidade / Flexibilidade | 2 |
| 4. Cultura / Maturidade em Processo | 2 |
| 5. Previsibilidade Arquitetural | 2 |
| 6. Experiência no Domínio | 1 |
| 7. Competência Pessoal | 1 |
| Total Geral | 11 |

O perfil de complexidade do ambiente para empresa 2 é classificado como médio por obter pontuação entre 11 e 17 no total geral.

A empresa foi mapeada no quadrante de **Complexidade Média Gerencial** devido à pontuação dos eixos com características predominantemente gerenciais (eixos 2, 3 e 6) ser maior que à pontuação dos eixos com características predominantemente técnicas (eixos 4, 5 e 7).

✓ **Listagem de práticas de Engenharia de Software propostas:**

Blocos de Práticas Gerenciais

- Definição inicial de escopo (Documento de visão)
- Planejamento de recursos humanos para o projeto, determinando funções, responsabilidades, relações hierárquicas
- Planejamento de recursos e ambiente necessários, incluindo, por exemplo, equipamentos, ferramentas, serviços, componentes, viagens, etc.
- Acompanhamento incluindo revisão e obtenção de compromisso dos planos do projeto com os interessados
- Representação do escopo através de Estrutura Analítica do Projeto (EAP)
- Dimensionamento do projeto sob o ponto de vista do usuário (Planilha de APF)
- Definição de um modelo e as fases do ciclo de vida do projeto (Iterativo)
- Cronograma com dependências entre as atividades, marcos de controle e produtos de trabalho gerados
- Identificação e monitoramento formal de riscos do projeto, com relação de impacto e probabilidade
- Controle formal de orçamento do projeto (Planilha de orçamento)
- Integração dos planejamentos realizados garantindo que a dependência entre eles seja monitorada
- Comunicação formal das atividades do projeto com todos os interessados
- Identificação e monitoramento formal de problemas do projeto

Blocos de Práticas Técnicas

- Identificação formal dos requisitos (Lista de Requisitos funcionais e não funcionais, Estórias de Usuários)
- Registro formal de solicitações de alteração e/ou inclusão de requisitos, incluindo análise de impacto para elas
- Definição de padrão de codificação para minimizar impactos de manutenção
- Realização de testes funcionais
- Utilização de prototipação

7 CONCLUSÕES E PERSPECTIVAS

Este trabalho demonstrou que é possível fundamentar uma metodologia para seleção de boas práticas isoladas de engenharia de software a partir da extração de informações e características do ambiente do problema.

A abordagem GQM adotada no diagnóstico do perfil mostrou-se muito satisfatória nos estudos de caso realizados, e apesar de contemplar razoavelmente todos os fatores investigados no levantamento do perfil, a mesma possui um caráter de independência na metodologia proposta no trabalho, podendo ser evoluída (através da evolução dos questionários) sem impactar o restante do trabalho.

A seleção e aplicação de práticas independentes foi sugerida para atender as limitações das micro e pequenas empresas que têm dificuldades, em sua grande maioria, em se adaptar às exigências do mercado em relação aos modelos de qualidade de processos organizacionais existentes. A carência de mão-de-obra e limitação de recursos para investimento tornam a implantação de tais processos inviáveis em tais empresas. O presente trabalho não descarta em seu escopo a importância dos modelos de qualidade atualmente no mercado, muito pelo contrário, a proposição de práticas isoladas visa preparar tais empresas para conseguir alcançar o quanto antes uma maturidade em termos de processo de desenvolvimento tornando-as aptas a almejar certificações CMMI e MPS.BR.

O conjunto de boas práticas selecionadas no trabalho foi associado aos quesitos do nível G do MPS.BR no Capítulo 4, demonstrando que as mesmas atendem largamente ou totalmente a grande parte deles, o que indica aderência do trabalho ao modelo de melhoria de processos em questão.

Os principais objetivos propostos inicialmente pelo trabalho foram alcançados, uma vez que o conjunto de boas práticas de engenharia foi levantado através dos principais modelos de qualidade de processo (CMMI E MPS.BR), sem é claro almejar esgotar todas as práticas e sim selecionar um subconjunto amplamente aceito por ambos modelos, o que apresenta mais uma indicação que o trabalho não se contrapõe a tais modelos e sim sugere uma etapa pré-implantação dos mesmos. Além disso, os fatores e características definidos para traçar o perfil de complexidade do ambiente foram selecionados através de relacionamento entre dois modelos da literatura para customização de processos. A relação de mapeamento foi baseada em critérios objetivos

para cada um dos sete fatores escolhidos, fazendo o método totalmente repetível, inclusive ilustrado através de estudos de caso em duas empresas.

Apesar dos estudos de caso terem sido realizados manualmente com apoio apenas de uma planilha eletrônica para auxiliar na atribuição de pesos das respostas, está sendo desenvolvida uma ferramenta web para aplicação da metodologia proposta no trabalho. A ferramenta para apoio ao mapeamento está sendo desenvolvida através de um trabalho de iniciação científica por um aluno de graduação em Ciência da Computação da UFV e contará com automatização da aplicação dos questionários bem como a construção do gráfico radial e apresentação de relatório com o conjunto de práticas sugeridas pelo método.

É importante citar que o conjunto de boas práticas propostas pelo método pode ser utilizado como blocos independentes e que a empresa pode combinar a elas outras práticas aplicadas com sucesso em projetos anteriores.

7.1 Principais Contribuições

As principais contribuições do trabalho foram:

- i. Uma forma de diagnosticar o perfil de complexidade do ambiente de desenvolvimento de MPEs, através da extensão dos questionários propostos por (Soares, 2007); da definição dos sete fatores para compor o perfil, mesclando os trabalhos de (Royce, 1998) e (Boehm and Turner, 2004a); da extensão da interpretação dos questionários proposta por (Soares, 2007) e da adaptação do gráfico polar proposto por (Boehm and Turner, 2004a).
- ii. A definição de um conjunto de boas práticas de engenharia de software, baseado nos modelos de melhoria de processos MPS.BR e CMMI e adaptado da estrutura de boas práticas proposta por (Leal, 2009), além da categorização das práticas em práticas técnicas e práticas gerenciais.
- iii. Uma metodologia para seleção de boas práticas baseado em parâmetros extraídos do ambiente do problema através da: ampliação do relacionamento das primitivas de processo listadas por (Royce, 1998) e os fatores que compõem o perfil de complexidade; definição de uma classificação para os quadrantes apresentados por (Royce, 1998) no que se refere à complexidade técnica e complexidade gerencial; e definição de um mapeamento entre os quadrantes e o conjunto de práticas técnicas e gerenciais correspondentes.

7.2 Trabalhos Futuros

Como trabalho futuro, sugere-se o acompanhamento da aplicação e implantação das práticas sugeridas para as empresas relacionadas no estudo de caso desse trabalho, bem como uma análise qualitativa e quantitativa dos resultados alcançados pela aplicação do método. Para isso, podem ser estabelecidas métricas que permitam a realização desse tipo de análise.

Outra sugestão para trabalhos futuros consiste na melhor utilização do gráfico radial traçado para levar em consideração a área do mesmo associada à soma dos fatores, a fim de dar um direcionamento e priorização das práticas a serem adotadas, análogo ao framework de adoção de práticas baseado em riscos para direcionar a customização de modelos de processos ágeis ou dirigidos por planejamento apresentado na Seção 2.3.1.

Utilização de técnicas de inteligência computacional, como por exemplo, funções *fuzzy* para ampliar os horizontes da pesquisa, permitindo um mapeamento ainda mais específico para realidade da empresa.

A evolução natural de um conjunto de práticas de engenharia de software adotadas por uma empresa é a materialização de um processo de desenvolvimento de software que as contemple. Sugere-se, portanto, a evolução da metodologia em estágios, onde o próximo estágio seria a indicação de um processo de desenvolvimento condizente com as práticas apontadas bem como com o perfil diagnosticado para empresa.

APÊNDICE A

A seguir são apresentados os questionários gerencial e individual que foram aplicados no trabalho para investigar os sete fatores que compõem o perfil de complexidade do ambiente.

QUESTIONÁRIO GERENCIAL

Abaixo, são detalhadas as perguntas presentes no questionário gerencial:

1 - Qual é o tamanho médio da equipe de desenvolvimento por projeto (marque apenas uma opção)?

- a) até 3 pessoas
- b) entre 4 e 10 pessoas
- c) entre 11 e 30 pessoas
- d) entre 31 e 100 pessoas
- e) entre 101 e 300 pessoas
- f) mais de 300

2 - Em geral, um indivíduo se dedica a quantos projetos simultaneamente (marque apenas uma opção)?

- a) 1
- b) 2
- c) 3
- d) 4
- e) 5
- f) mais de 5

3 - Qual é a quantidade média de Casos de Uso por projeto (marque apenas uma opção)?

- a) 10
- b) 20
- c) 30
- d) 40
- e) 50
- f) mais de 50

4 - Caso seja utilizada a técnica de estimativa de projetos por Pontos de Função, qual é a quantidade média de Pontos de Função por projeto (marque apenas uma

opção)?

- a) 100
- b) 200
- c) 400
- d) 600
- e) 1000
- f) mais de 1000

5 - Durante o ciclo de desenvolvimento de software, podem ocorrer mudanças nos requisitos já especificados. Em alguns casos, essas mudanças ocorrem em função de alterações sofridas pelo contexto do problema a ser resolvido pelo desenvolvimento de software. Outra fonte de mudanças nos requisitos seria a visualização de partes prontas do sistema o que pode levar o usuário a desejar outras funcionalidades ainda não especificadas, ou ainda o levantamento incorreto de requisitos que ocasiona mudanças de requisitos em estágios posteriores do desenvolvimento. De acordo com sua realidade, as mudanças de requisitos no sistema:

a- Devem-se a alterações sofridas pelo contexto do problema a ser resolvido (marque apenas uma opção).

- a) 100% das vezes
- b) 80% a 99% das vezes
- c) 51% a 79% das vezes
- d) 50% das vezes
- e) 30% a 49% das vezes
- f) 0% a 29% das vezes
- g) nunca

b - Devem-se a visualização de partes prontas do sistema (marque apenas uma opção).

- a) 100% das vezes
- b) 80% a 99% das vezes
- c) 51% a 79% das vezes
- d) 50% das vezes
- e) 30% a 49% das vezes
- f) 0% a 29% das vezes
- g) nunca

c -Devem-se ao levantamento incorreto de requisitos (marque apenas uma opção).

- a) 100% das vezes
- b) 80% a 99% das vezes
- c) 51% a 79% das vezes
- d) 50% das vezes
- e) 30% a 49% das vezes
- f) 0% a 29% das vezes
- g) nunca

6 - Com que frequência ocorre mudanças de requisitos no sistema (marque apenas uma opção)?

- a) nunca
- b) diariamente
- c) semanalmente
- d) mensalmente
- e) bimestralmente
- f) trimestralmente
- g) semestralmente
- h) anualmente

7 - Aproximadamente, cada uma dessas mudanças afeta que % do total de requisitos estabelecidos até o momento da mudança (marque apenas uma opção).

- a) 0% a 20%
- b) 21% a 40%
- c) 41% a 60%
- d) 61% a 80%
- e) 81% a 100%

8 - Com que frequência ocorre a substituição de membros da equipe? (marque apenas uma opção)

- a) semanalmente
- b) mensalmente
- c) bimestralmente
- d) trimestralmente
- e) semestralmente
- f) anualmente
- g) com menos frequência

h) nunca

9 - Qual é o tempo médio em que um membro permanece na equipe? (marque apenas uma opção)

a) 1 semana

b) 1 mês

c) 2 meses

d) 3 meses

e) 6 meses

f) de 6 a 12 meses

g) de 1 a 2 anos

h) de 2 a 3 anos

i) mais de 3 anos

10 - Quando ocorre a saída de um membro da equipe, qual é o tempo médio que esse indivíduo permanece na equipe transmitindo sua experiência adquirida para os membros que ficam e para os novos membros antes que ocorra a sua saída? (marque apenas uma opção)

a) 1 semana

b) 15 dias

c) 1 mês

d) de 1 a 6 meses

e) de 6 a 12 meses

f) mais de 1 ano

g) Não ocorre a permanência na equipe com esse propósito

11 - Quando ocorre a saída de membros da equipe, como a experiência adquirida pelos membros que saem é transmitida para os membros que ficam e para os novos membros? (assinale a(s) opção(ões) utilizada(s))

a) verbalmente

b) através de diagramas/modelos documentados

c) através de documentação sobre a comunicação entre os membros da equipe

d) através de documentação sobre o fluxo de trabalho e tarefas

desempenhadas por cada membro

12 - A partir das opções abaixo, marque aquelas que são empregadas no desenvolvimento de software? (assinale a(s) opção(ões) empregada(s))

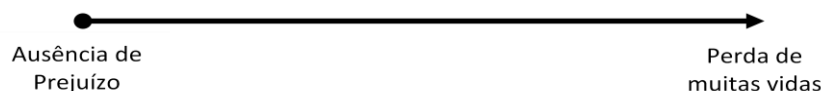
a) diagramas e/ou modelos pré-estabelecidos como padrão para toda a equipe

- b) padrão para a codificação
- c) documentação sobre o desenvolvimento dos projetos de software
- d) documentos e/ou ferramentas para documentar a comunicação e as decisões da equipe
- e) padrão para a documentação sobre o desenvolvimento do projeto
- f) algum tipo de processo de desenvolvimento de software definido

13 - A partir das fases listadas abaixo, quais estão incluídas no ciclo de desenvolvimento de software (assinale a(s) opção(ões) incluída(s))?

- a) análise de requisitos
- b) codificação
- c) especificação de requisitos
- d) teste
- e) projeto

14 - Possíveis falhas nos software desenvolvido podem ocasionar vários tipos de prejuízo. Nos casos mais graves, as falhas podem implicar em perda de uma ou mais vidas. Em outras situações, as falhas podem ocasionar prejuízos financeiros mais ou menos graves ou não ocasionar prejuízos substanciais. Considerando o tipo de software desenvolvido, Assinale um ponto na escala abaixo que represente a gravidade de possíveis falhas no software. Observe que os extremos representam o melhor caso (ausência de prejuízos) e o pior caso (perda de várias vidas).



15 – A Experiência no Domínio do projeto é dada pela capacidade da empresa em desenvolver de forma rápida e com qualidade projetos em uma determinada área do conhecimento. Para medir o nível de experiência no domínio dos projetos de uma empresa vamos analisar:

a – Quantidade de áreas de conhecimento que a empresa atua (marque apenas uma opção)?

- a) Uma
- b) Até 3
- c) Mais de 3

b – Quantidade média de projetos em cada área de conhecimento (Marque apenas uma opção)?

- a) Mais de 7
- b) Até 7
- c) 3

c – Quantidade média de projetos da mesma área de conhecimento desenvolvido pela mesma equipe (marque apenas uma opção)?

- a) 1
- b) Até 3
- c) Mais de 3

16 – A Previsibilidade Arquitetural do projeto é dada pelo grau de visibilidade técnica em termos de desempenho, robustez e confiabilidade. Para medir o nível de previsibilidade arquitetural dos projetos de desenvolvimento de uma empresa vamos analisar:

a – Porcentagem de projetos em que é possível conhecer previamente a necessidade de consumo de recursos computacionais, tempo de resposta ou taxa de transmissão de dados (marque apenas uma opção)?

- a) Mais de 50%
- b) Entre 10% e 50%
- c) Até 10%

b – Porcentagem de projetos em que é possível conhecer previamente o impacto de adição de novas funcionalidades ou novas tecnologias (marque apenas uma opção)?

- a) Mais de 50%
- b) Entre 10% e 50%
- c) Até 10%

c – Porcentagem de projetos em que é possível conhecer previamente sua tolerância a falhas (marque apenas uma opção)?

- a) Mais de 50%
- b) Entre 10% e 50%
- c) Até 10%

QUESTIONÁRIO INDIVIDUAL

Abaixo são detalhadas as perguntas presentes na versão final do questionário individual:

1 - A sua área de atuação principal é a área de Ciência da Computação?

- a) sim
- b) não

2 - Há quanto tempo você atua (ou já atuou) nas atividades relacionadas abaixo?

a - Atividades de Análise de requisitos(marque apenas uma opção)

- a) 0 a 6 meses
- b) 6 a 12 meses
- c) 1 a 3 anos
- d) 3 a 5 anos
- e) mais de 5 anos
- f) não atuo/ não atuei nessa atividade

b - Atividades de Especificação de Requisitos (marque apenas uma opção)

- a) 0 a 6 meses
- b) 6 a 12 meses
- c) 1 a 3 anos
- d) 3 a 5 anos
- e) mais de 5 anos
- f) não atuo/ não atuei nessa atividade

c - Atividades de Projeto(marque apenas uma opção)

- a) 0 a 6 meses
- b) 6 a 12 meses
- c) 1 a 3 anos
- d) 3 a 5 anos
- e) mais de 5 anos
- f) não atuo/ não atuei nessa atividade

d - Atividades de Codificação (marque apenas uma opção)

- a) 0 a 6 meses
- b) 6 a 12 meses
- c) 1 a 3 anos
- d) 3 a 5 anos
- e) mais de 5 anos

f) não atuou/ não atuei nessa atividade

e - Atividades de Teste (marque apenas uma opção)

a) 0 a 6 meses

b) 6 a 12 meses

c) 1 a 3 anos

d) 3 a 5 anos

e) mais de 5 anos

f) não atuou/ não atuei nessa atividade

f - Atividades Gerenciais (marque apenas uma opção)

a) 0 a 6 meses

b) 6 a 12 meses

c) 1 a 3 anos

d) 3 a 5 anos

e) mais de 5 anos

f) não atuou/ não atuei nessa atividade

3 - Com relação ao CMM (*Capability Maturity Model*) e/ou MPS.BR (Melhoria do Processo de Software Brasileiro), você (marque apenas uma opção):

a) desconhece

b) já ouviu falar

c) conhece

d) já trabalhou em um ambiente que utilizasse CMM/MPS.BR

4 - Com relação a processos tais como RUP (*Rational Unified Process*), PSP (*Personal Software Process*) e TSP (*Team Software Process*), você (marque apenas uma opção):

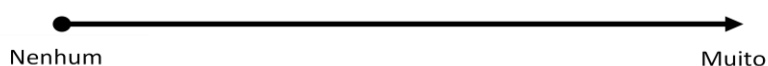
a) desconhece

b) já ouviu falar

c) conhece

d) já trabalhou em um ambiente que utilizasse algum desses processos

5 - Assinale um ponto na escala abaixo que reflita o seu nível de desconforto quando recebe uma solicitação de mudança nos requisitos?



APÊNDICE B

Atribuição de Pesos Gerencial

Na Tabela B.1 é mostrada a atribuição de pesos às respostas referentes às questões 11, 12 e 13 do questionário gerencial. Estas questões são utilizadas para quantificar o fator crítico cultura. A soma dos pesos é utilizada para se alcançar um percentual que indique o quanto a equipe é orientada pela ordem e pelo caos.

Tabela B.0.1 – Pesos das respostas do questionário gerencial

| Pergunta | Resposta | Peso |
|---|---|------|
| 11 - Quando ocorre a saída de membros da equipe, como a experiência adquirida pelos membros que saem é transmitida para os membros que ficam e para os novos membros? (assinale a(s) opção(ões) utilizada(s)) | verbalmente | 0 |
| | através de documentação sobre a comunicação entre os membros da equipe | 1 |
| | através de diagramas / modelos documentados | 1 |
| | Através de documentação sobre o fluxo de trabalho e tarefas desempenhadas por cada membro | 1 |
| 12 – A partir das opções abaixo, marque aquelas que são empregadas no desenvolvimento de software. (Assinale a(s) opção(ões) empregada(s)) | diagramas e/ou modelos pré-estabelecidos como padrão para toda a equipe | 1 |
| | padrão para a codificação | 1 |
| | documentação sobre o desenvolvimento dos projetos de software | 1 |
| | Documentos e/ou ferramentas para documentar a comunicação e as decisões da equipe | 2 |
| | padrão para a documentação sobre o desenvolvimento do projeto | 2 |
| | algum tipo de processo de desenvolvimento de software definido | 3 |
| 13 - A partir das fases listadas abaixo, quais estão incluídas no ciclo de desenvolvimento de software (assinale a(s) opção(ões) incluída(s)). | análise de requisitos | 3 |
| | codificação | 1 |
| | especificação de requisitos | 3 |
| | teste | 1 |
| | projeto | 3 |

Atribuição de Pesos Individual

Na Tabela B.2 é mostrada a atribuição de pesos às respostas referentes às questões 1, 2, 3 e 4 do questionário individual. Estas questões são utilizadas para quantificar o fator crítico competência pessoal. A soma dos pesos é utilizada para se alcançar uma pontuação para cada membro da equipe de desenvolvimento.

Tabela B.0.2 – Pesos das respostas do questionário individual

| Pergunta | Resposta | Peso |
|---|------------------------------------|------|
| 1 - A sua área de atuação principal é a área de Ciência da Computação? | Sim | 2 |
| | Não | 0 |
| 2 - Há quanto tempo você atua (ou já atuou) nas atividades relacionadas abaixo? | | |
| A) Atividades de Análise de requisitos (marque apenas uma opção) | 0 a 6 meses | 1 |
| | 6 a 12 meses | 2 |
| | 1 a 3 anos | 3 |
| | 3 a 5 anos | 4 |
| | mais de 5 anos | 5 |
| | não atuo/ atuei nessa atividade | 0 |
| B) Atividades de Especificação de Requisitos (marque apenas uma opção) | 0 a 6 meses | 1 |
| | 6 a 12 meses | 2 |
| | 1 a 3 anos | 3 |
| | 3 a 5 anos | 4 |
| | mais de 5 anos | 5 |
| | não atuo/ atuei nessa atividade | 0 |
| C) Atividades de Projeto (marque apenas uma opção) | 0 a 6 meses | 1 |
| | 6 a 12 meses | 2 |
| | 1 a 3 anos | 3 |
| | 3 a 5 anos | 4 |
| | mais de 5 anos | 5 |
| | não atuo/ atuei nessa atividade | 0 |
| D) Atividades de Codificação (marque apenas uma opção) | 0 a 6 meses | 1 |
| | 6 a 12 meses | 2 |
| | 1 a 3 anos | 3 |
| | 3 a 5 anos | 4 |
| | mais de 5 anos | 5 |
| | não atuo/ atuei nessa atividade | 0 |
| E) Atividades de Teste (marque apenas uma opção) | 0 a 6 meses | 1 |
| | 6 a 12 meses | 2 |
| | 1 a 3 anos | 3 |
| | 3 a 5 anos | 4 |
| | mais de 5 anos | 5 |
| | não atuo/ atuei nessa atividade | 0 |

| | | |
|---|---|---|
| F) Atividades Gerenciais (marque apenas uma opção) | 0 a 6 meses | 1 |
| | 6 a 12 meses | 2 |
| | 1 a 3 anos | 3 |
| | 3 a 5 anos | 4 |
| | mais de 5 anos | 5 |
| | não atuo/ atuei nessa atividade | 0 |
| 3 - Com relação ao CMM e/ou MPS.BR, você (marque apenas uma opção): | desconhece | 0 |
| | já ouviu falar | 2 |
| | Conhece | 4 |
| | já trabalhou em um ambiente que utilizasse algum desses processos | 6 |
| 4 - Com relação a processos tais como RUP, PSP e TSP, você (marque apenas uma opção): | desconhece | 0 |
| | já ouviu falar | 2 |
| | Conhece | 4 |
| | já trabalhou em um ambiente que utilizasse algum desses processos | 6 |

APÊNDICE C

Respostas dos questionários aplicados no Estudo de Caso

Nas tabelas a seguir são mostradas as respostas obtidas através da aplicação dos questionários gerencial e individual às duas empresas descritas na Seção 6 do trabalho.

Empresa 1: Cientec – Consultoria e Desenvolvimento de Sistemas Ltda.

Tabela C.0.1 – Respostas do questionário gerencial da empresa 1

| Questionário Gerencial | | | | | | | | | | |
|-------------------------------|-----------|-------------|-------------------|-----------|------------|------------|------------|------------|------------|------------|
| 1 | 2 | 3 | 4 | 5a | 5b | 5c | 6 | 7 | 8 | 9 |
| B | B | D | D | B | C | D | C | B | F | H |
| 10 | 11 | 12 | 13 | 14 | 15a | 15b | 15c | 16a | 16b | 16c |
| C | A-B- D | B-C- E-F | A-B- C-D- E | 5,5 cm | A | A | B | C | C | C |

Tabela C.0.2 – Respostas do questionário individual da empresa 1

| Questionário Individual | | | | | | | | | | |
|--------------------------------|----------|-----------|-----------|-----------|-----------|-----------|-----------|----------|----------|----------|
| Respondente | 1 | 2a | 2b | 2c | 2d | 2e | 2f | 3 | 4 | 5 |
| 1 | A | C | B | B | C | F | C | D | C | A |
| 2 | A | F | F | A | C | A | A | B | C | B |
| 3 | A | B | B | B | E | A | B | C | C | B |
| 4 | A | B | B | B | A | C | A | C | B | A |
| 5 | A | A | A | B | C | A | A | C | C | B |
| 6 | A | B | B | B | B | C | F | A | A | B |
| 7 | A | C | C | C | C | C | F | C | B | B |
| 8 | B | F | C | F | F | C | F | B | A | B |

Empresa 2: Jungle – Digital Games.

Tabela C.0.3 – Respostas do questionário gerencial da empresa 2

| Questionário Gerencial | | | | | | | | | | |
|-------------------------------|-----------|-----------|-------------|-----------|------------|------------|------------|------------|------------|------------|
| 1 | 2 | 3 | 4 | 5a | 5b | 5c | 6 | 7 | 8 | 9 |
| B | A | A | B | E | C | B | D | B | F | G |
| 10 | 11 | 12 | 13 | 14 | 15a | 15b | 15c | 16a | 16b | 16c |
| C | A | A-B- C | A-B- C-D | 3,9 cm | B | C | B | A | A | A |

Tabela C.0.4 – Respostas do questionário individual da empresa 2

| Questionário Individual | | | | | | | | | | |
|--------------------------------|----------|-----------|-----------|-----------|-----------|-----------|-----------|----------|----------|----------|
| Respondente | 1 | 2a | 2b | 2c | 2d | 2e | 2f | 3 | 4 | 5 |
| 1 | B | F | F | F | F | F | F | A | A | A |
| 2 | B | A | A | F | F | F | F | A | A | B |
| 3 | B | F | F | C | F | C | C | B | A | B |
| 4 | A | E | E | E | E | E | E | D | D | B |
| 5 | A | D | D | D | E | D | C | C | C | A |
| 6 | A | C | C | C | E | B | D | C | C | B |
| 7 | A | C | C | E | E | C | F | B | A | A |
| 8 | A | A | F | F | F | F | F | A | A | A |

REFERÊNCIAS BIBLIOGRÁFICAS

ABNT. (2001). Associação Brasileira de Normas Técnicas. “**Série ISO 9000:2000: Sistemas de Gestão da Qualidade**”. ABNT.

Advanced Development Methods – ADM. (2010). **Controlled chaos: living on the edge**, <http://www.controlchaos.com/>, Acessado em Abril 2010.

Agerfalk, P. J. and Fitzgerald, B. (2006). **Flexible and distributed software processes: old petunias in new bowls?** Commun. ACM, 49(10): páginas 26–34.

Aggarwal, R. (1998). **Small Firm Survival and Technological Activity**. Small Business Economics 11.

Alliance, A. (2001). **Manifesto for agile software development**. <http://agilemanifesto.org/>. Acessado em 04/01/2009.

Alliance, A. (2007). **Agile alliance**. <http://www.agilealliance.org>. Acessado em 04/01/2009.

Ambler, S. W. (2002). **Agile modeling effective practices for eXtreme programming and the unified process**. Wiley, New York.

Anderson, D. J. (2005). **Stretching Agile to fit CMMI Level 3**, presented at Agile 2005 Conference, Dever, http://www.agilemanagement.net/Articles/Papers/Agile_2005_Paper_DJA_v1_5.pdf, 2005.

Anderson, D. J. (2003). **Agile Management for Software Engineering - Applying the Theory of Constraints for Business Results**, Prentice Hall.

Associação Brasileira das Empresas de Software (Abes). (2004). "**Mercado Brasileiro de Software - Panorama e Tendências**".

Associação Brasileira das Empresas de Software (Abes). (2009). "**Mercado Brasileiro de Software - Panorama e Tendências**".

Basili, V. (1992). **Software Modeling and Measurement: The Goal / Question / Metric Paradigm**. University of Maryland, CS-TR-2956, UMIACSTR páginas 92-96.

Basili, V. R., Caldiera, G., and Rombach, D. (1994). **Goal Question Metric Approach**. Encyclopedia of Software Engineering, páginas 528–532.

Beck, K. (2000). **Extreme Programming Explained: Embrace Change**. Reading, MA: Addison Wesley Longman, Inc.

Belgamo, A.; Fabbri, S. (2004). **Um Estudo sobre a Influência da Sistematização da Construção de Modelos de Casos de Uso na Contagem dos Pontos de Casos de Uso**. III Simpósio Brasileiro de Qualidade de Software. Brasília.

Berander, P. and Jönsson, P. (2006). **A goal question metric based approach for efficient measurement framework definition**. In ISESE '06: Proceedings of the 2006 ACM/IEEE international symposium on International symposium on empirical software engineering, páginas 316–325, New York, NY, USA. ACM Press.

Boehm, B. (2002). **Get ready for agile methods, with care**. IEEE Computer, 35(1): páginas 64–69.

Boehm, B. and Turner, R. (2004a). **Balancing Agility and Discipline: A Guide for the Perplexed**. Addison-Wesley, Boston.

Boehm, B. W. (1998). **A Spiral Model of Software Development and Enhancement**, IEEE Computer, Maio de 1998, páginas - 61-72

Brooks, F. P. (1987). **No silver bullet: essences and accidents of Software Engineering**. IEEE. Computer, v. 20, n. 4.

CMU (2002). CARNEGIE MELLON UNIVERSITY. **Capability Maturity Model Integration (CMMI), Version 1.1: CMMI for Software Engineering (CMMI-SW, V1.1) - Staged Representation**. Pittsburg.

Cockburn, A. (2000). **Selecting a project's methodology**. IEEE Software, 17(4): páginas 64–71.

Cockburn, A. (2002). **Agile Software Development**. Addison-Wesley, Boston.

Cockburn, A. and Highsmith, J. (2001). **Agile software development, the people factor**. Computer, 34(11): páginas 131–133.

Dennis, A., Wixom, B. H. (2005) **Análise e Projeto de Sistemas**. Rio de Janeiro: LTC.

Dorfmann, M. e Thayer, R., (1990). **Standards, Guidelines, and Examples of System and Software Requirements Engineering**. Los Alamitos, CA: IEEE Computer Society Press,.

Feiler, P. H., H. (1993). **Software process development and enactment: Concepts and definitions**. IEEE Computer Society Press.

Filho, W. P. (2003). **Engenharia de Software: Fundamentos, Métodos e Padrões**. LTC, Rio de Janeiro, segunda edição.

Gloger, B., **The Zen of Scrum**, (2007). <http://www.glogerconsulting.de>, Acessado em Fevereiro 2010.

Gruhn, V. (2002). **Process-centered software engineering environments a brief history and future challenges**. Annals of Software Engineering, 14: páginas 363–382.

Highsmith, J. and Cockburn, A. (2001). **Agile software development: the business of innovation**. Computer, 34(9): páginas 120–127.

Humphrey, W. S. (2000a). **The Personal Software Process (PSP)**. Technical report, CMU/SEI-2000-TR-022. ESC-TR-2000-022.

Humphrey, W. S. (2000b). **The Team Software Process (TSP)**. Technical report, CMU/SEI-2000-TR-023. ESC-TR-2000-023.

Humphrey, Watts S. (2000) **The Personal Software Process**. TECHNICAL REPORT. CMU/SEI-2000-TR-022.

IEEE, (1990) Std 610.12 - **IEEE Standard Glossary of Software Engineering Terminology, Institute of Electrical and Electronics Engineers**.

IEEE, (1998) Std 830-1998 - **IEEE recommended practice for software requirements specifications, Institute of Electrical and Electronics Engineers**.

ISO/IEC (1998) International Organization for Standardization. ISO/IEC 9126-1.2, **Information Technology - Software product quality - part 1: Quality model**.

ISO/IEC 15504. (2003) International Organization for Standardization. **“ISO/IEC 15504: Information Technology – Process Assessment, Part 1 to Part 5”**. ISO/IEC Intermediate Report.

ISO/IEC, (2008) International Organization for Standardization. ISO/IEC 12207:2008 **Systems and software engineering — Software life cycle processes**, Geneve: ISO.

Jones, C. (1996). **Patterns of Software Systems Failure and Success**. Londres: International Thompson Computer Press.

Kruchten, P. (2001). **The nature of software: what’s so special about software engineering?** The Rational Edge.

Kruchten, P. (2004). **Introdução ao RUP – Rational Unified Process**. Trad. Deborah Rudiger, Addison-Wesley, Boston.

Larman C. (2004). **Agile & Iterative Development, A Manager’s Guide**, Addison-Wesley.

Leal, A. L. C. (2009). **Uma proposta de taxonomia de boas práticas em desenvolvimento de software**. Dissertação de Mestrado – DPI/UFV - Viçosa-MG.

Lin, S., Liu, F., Loe, S. (2003): **Building A Knowledge Base of IEEE/EAI 12207 and CMMI with Ontology**. Sixth International Protégé Workshop, Manchester, England.

MCT. (2005) Ministério da Ciência e Tecnologia. **Qualidade e Produtividade no Setor de Software Brasileiro, Resultados da Pesquisa 2005**. Disponível em: <<http://www.mct.gov.br/index.php/content/view/3253.html>>.

MCT. (2009) Ministério da Ciência e Tecnologia. **Qualidade e Produtividade no Setor de Software Brasileiro, Resultados da Pesquisa 2001**. Disponível em: <http://www.mct.gov.br/upd_blob/0210/210931.pdf>.

Methods, A. D. (2007). **Scrum**. <http://www.controlchaos.com/>. Acessado em 04/01/2009.

MIT Project, Softex, W-Class. (2002). **"The Software Industry in Brazil 2002: Strengthening The Economy of Knowledge"**.

MPS.BR (2009). **MPS.BR – Melhoria do Processo de Software Brasileiro – Guia Geral** – Softex. MPS.BR – 2009.

Paulk, M. (1994). **The Capability Maturity Model: Guidelines for Improving the Software Process**. Carnegie Mellon University, Software Engineering Institute. Addison-Wesley.

Paulk, M. C.; Curtis, B.; Chrissis, M. B.; Webber, C. V. (1993) **Capability maturity model sm for software, version 1.1. Technical report CMU/SEI-93-TR-024**, SEI Joint Program Office Hanscom AFB, MA 01731-2116, 1993.

Pedreira, Oscar et al. (2007). **A Systematic Review of Software Process Tailoring**. ACM SIGSOFT Software Engineering Notes. v.32, n.3.

Pfleeger, S. L. (2004). **Engenharia de software: teoria e prática**. 2.Ed. São Paulo: Prentice Hall.

PMI - PROJECT MANAGEMENT INSTITUTE. (2008). **A Guide To The Project Management Body of Knowledge**. 4. ed. Newton Square: PMI Publications. Disponível em: <www.pmi.org>.

Pressman, R. S. (2001). **Software Engineering: A Practitioner's Approach**, volume 5. McGraw-Hill.

Rovere L., Renata L., Erber, Fabio S., e Hasenclever L. (2000). **Industrial and Technology Policies and Local Economic Development: enhancing and supporting clusters**. Third Triple Helix International Conference, 26 a 29 de abril de 2000, Hotel Glória, Rio de Janeiro. Disponível em CDRom (PEP/COPPE/UFRJ)

Royce, W. (1998) **Software Project Management – A Unified Framework**, Addison-Wesley, Boston.

Schwaber, K. (2004). **Agile Project Management With Scrum**, Microsoft.

Schwaber, K. and Beedle, M. (2002). **Agile Software Development With Scrum**. NJ: Prentence Hall.

SEBRAE (2004). Serviço Brasileiro de Apoio às Micro e Pequenas Empresas. **Fatores Condicionantes e Taxa de Mortalidade de Empresas no Brasil**. Relatório de Pesquisa. Brasília.

Segal, J. (2005). **When software engineers met research scientists: a case study**. **Empirical Software Engineering**, 10: páginas: 517–536.

SEI, (2006) SOFTWARE ENGINEERING INSTITUTE. **CMMI for Development (CMMI-DEV), Version 1.2, Technical Report CMU/SEI-2006-TR-008**. Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University.

SEI. (2002) SOFTWARE ENGINEERING INSTITUTE. **CMMI for Systems Engineering/Software Engineering (CMMI-SE/SW), Staged Representation, Version 1.1, Technical report CMU/SEI-2002-TR-02**. Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University.

Silva, R. A. C., Braga, J. L., Silva, C. H. O., and Soares, L. S. (2006). **Pspm na melhoria da competência pessoal no desenvolvimento de software**. In Proceedings of JIISIC'06: 5ta. Jornada Iberoamericana de Ingeniería del Software e Ingeniería del Conocimiento, Puebla, Puebla, Mexico.

Soares, L. S. (2007). **Obtenção de requisitos para customização de processos de desenvolvimento de software**. Dissertação de Mestrado – DPI/UFV - Viçosa-MG.

Solingen, V., Basili, V. R., Caldiera, G., and Rombach, D. (2002). **Goal Question Metric (GQM) approach**. Encyclopedia of Software Engineering.

Sommerville, I. (2003). **Engenharia de Software** / Ian Sommerville; tradução André Maurício de Andrade Ribeiro; revisão técnica Kechi Hirama. São Paulo: Addison Wesley.

Sommerville, I. (2004). **Engenharia de Software**. 6ª edição. São Paulo: Addison Wesley.

SWEBoK. (2004) **Software Engineering Body of Knowledge**. IEEE - Computer Society Version 2004.

Tavares, Helena Cristina A. B.; Carvalho Ana Elizabete S.; Castro Jaelson F. B. (2004). **Medição de Pontos por Função a Partir da Especificação de Requisitos**. Departamento de Informática - UFPE.

Vos, Jan-Peter, Keizer, Jimme, E Halman, Joop M. (1998). **Diagnosing Constraints in Knowledge of SMEs**. Technological Forecasting and Social Change.

Wazlawick, R. S. (2004). **Análise e projeto de sistemas de informação orientados a objetos**. Rio de Janeiro: Elsevier.

Yourdone E. (1997). **Death March: Managing Mission Impossible Projects**. Upper Saddle River, NJ: Prentice-Hall.