

LIZIANE SANTOS SOARES

OBTENÇÃO DE REQUISITOS PARA CUSTOMIZAÇÃO DE
PROCESSO DE DESENVOLVIMENTO DE SOFTWARE

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
2007

LIZIANE SANTOS SOARES

**OBTENÇÃO DE REQUISITOS PARA CUSTOMIZAÇÃO DE PROCESSO DE
DESENVOLVIMENTO DE SOFTWARE**

**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*.**

APROVADO: 25 de julho de 2007

Prof. Alcione de Paiva Oliveira
(Co-orientador)

Prof. Carlos Henrique Osório Silva
(Co-orientador)

Prof. Ricardo dos Santos Ferreira

Prof. Clarindo Isaías P. da S. de Pádua

Prof. José Luis Braga
(Orientador)

Aos meus pais Wilson e Adelaide,
ao meu orientador José Luis Braga
e em especial, ao meu companheiro Marcos Paulo.

AGRADECIMENTOS

- Aos meus pais Adelaide e Wilson, pelo apoio e amor incondicionais, sem eles eu não teria chegado até aqui.
- Às minhas irmãs Fernanda e Juliana minhas melhores amigas.
- À minha Avó Maria por todas as lições de vida ensinadas.
- À minha Madrinha Judite que sempre foi uma segunda mãe.
- Ao Amigo e prof. José Luis Braga pela amizade e orientação neste trabalho, os quais foram muito importantes para o meu processo de crescimento pessoal.
- Ao meu companheiro Marcos Paulo pelo seu amor, apoio e dedicação.
- À Ireninha e Guilherme - *Família Buscapé* - pelo carinho, amizade e alegria com que me receberam.
- À amiga Eliana.
- Aos grupos BIOMA e GPRH.
- Aos meus conselheiros Carlos Henrique Osório Silva e Alcione de Paiva Oliveira.
- A todos os professores do DPI pelos anos de aprendizado e convívio.
- A todos os funcionários do DPI, sempre solícitos e gentis.
- À Capes pelo apoio financeiro.
- A todos aqueles que contribuíram de alguma forma para a realização desse trabalho.

BIOGRAFIA

Liziane Santos Soares, filha de Wilson Soares Pereira e Maria Adelaide Santos Soares, brasileira, nascida em 28 de maio de 1979 no município de Juiz de Fora, no Estado de Minas Gerais.

No ano 2002 concluiu o curso de Ciência da Computação na Universidade Federal de Viçosa - UFV e trabalhou como Analista de Sistemas em Belo Horizonte - MG durante dois anos. Em 2004 retornou à cidade de Viçosa e passou no concurso para professora substituta do Departamento de Informática - DPI, permanecendo no cargo por dois anos. Em 2005 foi aceita no curso de mestrado em Ciência da Computação na Universidade Federal de Viçosa - UFV, onde se tornou mestre, defendendo sua dissertação em julho de 2007.

SUMÁRIO

LISTA DE TABELAS	vii
LISTA DE FIGURAS	viii
RESUMO	ix
ABSTRACT	x
1 Introdução	1
2 Revisão Bibliográfica	4
2.1 Qualidade de Software baseada em Processo	4
2.2 Métodos dirigidos por planejamento	6
2.3 Métodos Ágeis	7
2.4 Métodos Híbridos	8
2.5 Goal Question Metric (GQM)	9
2.6 Trabalhos Relacionados	11
3 Metodologia	13
3.1 Aplicação da abordagem GQM	21
3.1.1 Projeto da estrutura da aplicação do GQM	21
3.1.2 Elaboração do questionário	26
3.1.3 Validação do questionário	27
3.1.4 Questionário	29
3.1.5 Análise e Interpretações associadas ao questionário	36
4 Resultados e Discussões	46
4.1 Estudos de caso	46
4.1.1 Formato de apresentação dos estudos de casos	47
4.1.2 Equipe1	48

4.1.3	Equipe2	51
4.1.4	Equipe3	53
4.1.5	Resultados comparativos entre as equipes	56
4.2	Análise complementar dos fatores	56
4.2.1	Escala do eixo competência pessoal no gráfico polar	57
4.2.2	Interferência entre os fatores críticos	59
4.2.3	Fatores determinantes	62
5	Conclusões e perspectivas	65
5.1	Trabalhos Futuros	66
A	Atribuições de pesos	67
B	Atribuições de pesos	69
C	Respostas dos questionários	71
	Referências Bibliográficas	73

LISTA DE TABELAS

3.1	Riscos associados ao projeto.	16
3.2	Classificação dos níveis de habilidade dos indivíduos.	18
3.3	Aplicação de GQM na quantificação do fator crítico <i>tamanho</i>	22
3.4	Aplicação de GQM na quantificação do fator crítico <i>dinamismo</i>	23
3.5	Aplicação de GQM na quantificação do fator crítico <i>cultura</i>	24
3.6	Aplicação de GQM na quantificação do fator crítico <i>criticalidade</i>	25
3.7	Aplicação de GQM na quantificação do fator crítico <i>competência pessoal</i>	26
3.8	<i>Pontuação</i> necessária para cada nível de habilidade.	44
4.1	Classificação da equipe analisada.	47
4.2	Classificação da Equipe 1 segundo os cinco fatores críticos.	50
4.3	Classificação da Equipe 2 segundo os cinco fatores críticos.	52
4.4	Classificação da Equipe 3 segundo os cinco fatores críticos.	55
4.5	Classificação de todas as equipes analisadas.	56
A.1	Pesos associados às respostas às <i>questões 9, 10 e 11</i> do questionário gerencial	68
B.1	Pesos associados às respostas às <i>questões 1, 2A, 2B, 2C, 2D, 2E, 2F, 3 e 4</i> do questionário individual	70
C.1	Respostas fornecidas pela equipe1	71
C.2	Respostas fornecidas pela equipe2	72
C.3	Respostas fornecidas pela equipe3	72

LISTA DE FIGURAS

2.1	Estrutura Hierárquica da Abordagem GQM	11
3.1	Fluxograma das etapas de desenvolvimento desse trabalho.	14
3.2	Resumo do método proposto por Boehm and Turner (2004a).	16
3.3	Gráfico polar com a representação dos fatores críticos.	19
3.4	Gráfico polar referente à SupplyChain.com.	20
3.5	Mapeamento dos intervalos da escala para as possíveis situações de perdas	41
4.1	Gráfico Polar	48
4.2	Gráfico polar referente ao perfil de desenvolvimento da Equipe 1	50
4.3	Gráfico polar referente ao perfil de desenvolvimento da Equipe 2	53
4.4	Gráfico polar referente ao perfil de desenvolvimento da Equipe 3	55
4.5	Proposta de alteração da escala do eixo <i>competência pessoal</i> com o objetivo de torná-la mais abrangente.	58
4.6	Representação da interferência entre o fator <i>Criticalidade</i> e o fator <i>Cultura</i>	59
4.7	Representação da interferência entre os fatores críticos <i>Tamanho</i> e <i>Cul-</i> <i>tura</i>	60
4.8	Representação da interferência entre os fatores críticos <i>Dinamismo</i> e <i>Cultura</i>	61
4.9	Representação da interferência entre o fator <i>Competência Pessoal</i> e o fator <i>Cultura</i>	62
4.10	Representação da interferência entre o fator <i>Cultura</i> e o fator <i>Com-</i> <i>petência Pessoal</i>	63

RESUMO

SOARES, Liziane Santos, M.Sc., Universidade Federal de Viçosa, julho de 2007.
Obtenção de requisitos para customização de processo de desenvolvimento de software. Orientador: José Luis Braga. Co-Orientadores: Carlos Henrique Osório e Alcione de Paiva Oliveira.

Desenvolver software com qualidade é, sem dúvida, o principal objetivo da Engenharia de Software. A qualidade do software está diretamente relacionada à qualidade do processo empregado em sua produção. Várias abordagens podem ser utilizadas durante a aplicação de um processo. Mas mesmo com tantas técnicas e métodos disponíveis, a escolha da técnica ou método a ser usado em cada tipo de problema ainda é feita de maneira empírica, sem bases científicas que justifiquem a escolha e procure a aderência do método ou técnica ao tipo de problema em questão. Este trabalho apresenta um método para obter requisitos que viabilizem o delineamento de um perfil composto pelas características inerentes à equipe, ao ambiente e aos problemas resolvidos no desenvolvimento. Esse perfil torna-se a base para a escolha do tipo de processo ou ainda a obtenção de um processo híbrido. O desenvolvimento deste trabalho envolve a aplicação da abordagem GQM na definição dos requisitos necessários ao delineamento do perfil da equipe perante fatores de riscos que caracterizam a equipe e o contexto de desenvolvimento. A partir da definição dos requisitos, é elaborado e testado um questionário através do qual os requisitos definidos são medidos para cada equipe. Como parte final do método proposto, é apresentada uma análise orientada para os dados coletados onde são esclarecidas as possíveis interpretações e classificações para os dados. São realizados estudos de caso onde o método proposto é aplicado a três equipes diferentes com o objetivo de analisar o perfil de cada uma delas.

ABSTRACT

SOARES, Liziane Santos, M.Sc., Universidade Federal de Viçosa, July of 2007.
An approach to critical factors measurement to guide software process customization. Adviser: José Luis Braga. Co-Advisers: Carlos Henrique Osório and Alcione de Paiva Oliveira.

The main goal of Software Engineering is developing quality software. Software quality is straightly related to the quality of the process employed in software production. Process specification is still an art, and it does not rely entirely on parameters that establish the relationship between context and process. There are many software process choices available, and in most cases a process that is not adequate to the kind of problems an organization is focused on is used. Choices are made empirically, based on past experience of members of the development team, and in general they do not rely on a technical approach based on more scientific bases and fundamentals. This work presents a method (procedure) that allows obtaining requirements to draw a profile composed of software development context characteristics. The determination of those requirements is done by the application of the GQM approach for their definition, extraction and analysis, thus allowing a more accurate quantification and subsequently the construction of a profile that forms a base for a more accurate choice of software development process for an organization or team. From the requirement definition, a questionnaire is developed and tested. The questionnaire is used to measure the defined requirement for each team. As the final step of the proposed method, an oriented data analysis is presented and the possible data interpretations are shown.

Case studies are carried out so the proposed method is applied to three different teams aiming to analyze the profile of each one.

Capítulo 1

Introdução

"In the absence of meaningful standards, a new industry like software comes to depend instead on folklore."

Tom DeMarco

Desenvolver software com qualidade é, sem dúvida, o principal objetivo da Engenharia de Software. A ubiquidade do software faz desse objetivo uma necessidade. A estrutura da maioria das organizações depende do suporte fornecido por esses sistemas e por isso, eles se tornam um elemento estratégico na diferenciação de produtos e serviços (Pressman, 2005).

O desenvolvimento de software pode ser visto como um conjunto de atividades que transforma os requisitos do usuário em software (Humphrey, 1989). Essas atividades devem envolver a descoberta das necessidades e expectativas do usuário, a formalização das mesmas e a aplicação de ferramentas e metodologias que produzam uma aplicação final confiável e de qualidade.

Um software obtém êxito quando atende às necessidades e expectativas de seus usuários. A obtenção de softwares de qualidade requer disciplina, justificando assim a necessidade de um enfoque de engenharia (Pressman, 2005).

A produção de software deve primar pela qualidade. Um dos desafios levantados em (SBC, 2006) refere-se justamente à pesquisa de métodos e técnicas para o desenvolvimento de software que contribuam para um sistema final que alcance qualidade, disponibilidade, corretude, segurança, escalabilidade e persistência.

A qualidade do software, em geral, está relacionada à qualidade do processo empregado em sua produção. Nos últimos anos, foram observados muitos esforços direcionados para a obtenção de tecnologias de processo de software que propiciem mecanismos de gerenciamento automatizado para o seu desenvolvimento.

Várias abordagens podem ser utilizadas durante a aplicação de um processo. Mas mesmo com tantas técnicas e métodos disponíveis, a escolha da técnica ou método a ser usado em cada tipo de problema ainda é feita de maneira empírica, sem bases científicas que justifiquem a escolha e procure a aderência do método ou técnica ao tipo de problema em questão. É importante que seja traçado um perfil baseado nas características inerentes à equipe, ao ambiente e aos problemas resolvidos no desenvolvimento, e que a escolha do processo tenha esse perfil como meta.

O problema que se propõe resolver nesse trabalho é apresentar um método cuja finalidade seja a obtenção de requisitos para estimar características inerentes à equipe e ao desenvolvimento de software e que possam ser a base para a seleção ou customização de processo de desenvolvimento de software.

As características consideradas neste trabalho são aquelas denominadas por Boehm and Turner (2004a) como sendo fatores críticos inerentes ao desenvolvimento e que são utilizados na adequação de Métodos Ágeis (MA) ou Métodos Dirigidos por Planejamento (MDP) a projetos de software. Essa adequação pode ser realizada pelo método proposto em (Boehm and Turner, 2004a) objetivando um processo híbrido que se adeque melhor a um contexto específico de desenvolvimento.

O desenvolvimento deste trabalho envolve a aplicação da abordagem GQM para determinar quais são os requisitos necessários ao delineamento do perfil da equipe perante certos fatores de riscos que, segundo Boehm and Turner (2004a), podem caracterizar a equipe e o contexto de desenvolvimento.

A partir da definição dos requisitos, passa-se para um próximo passo constituído pela elaboração de um questionário através do qual os requisitos definidos são medidos para cada equipe analisada.

Como parte final do método proposto, é apresentada uma análise orientada para os dados coletados onde são esclarecidas as possíveis interpretações e classifi-

cações para os dados. Para finalizar o trabalho, são realizados estudos de caso onde o método proposto é aplicado a três equipes diferentes com o objetivo de analisar o perfil de cada uma delas.

Este trabalho apresenta a seguinte estrutura: no Capítulo 2, são apresentados os conceitos chaves presentes na literatura e que embasam esse trabalho, são eles: qualidade de software baseada em processo, métodos dirigidos por planejamento, métodos ágeis, métodos híbridos, abordagem GQM, além de trabalhos relacionados. No Capítulo 3 são detalhadas as atividades realizadas para o desenvolvimento do método proposto por este trabalho. O Capítulo 4 descreve a realização de estudos de caso onde o método proposto é aplicado a três equipes diferentes e o resultado de cada aplicação é analisado. Além disso, são apresentadas outras discussões derivadas de observações feitas durante a realização dos estudos de caso. No Capítulo 5 são apresentadas as considerações finais sobre o trabalho e perspectiva para trabalhos futuros. Os Apêndices A, B e C apresentam dados complementares a serem utilizados durante a leitura do trabalho.

Capítulo 2

Revisão Bibliográfica

"If it is developed thoughtlessly and applied mindlessly, process can become the death of common sense."

Philip K. Howard

2.1 Qualidade de Software baseada em Processo

A qualidade de software tem sido um objetivo buscado por gerentes e desenvolvedores de software, os quais também têm que conciliar o cumprimento de prazos, orçamentos e a satisfação do cliente. Pode-se definir qualidade de software como sendo a conformidade com requisitos funcionais e de desempenho, documentados de forma explícita, além de características implícitas que são desejáveis em todo software (Pressman, 2005).

No desenvolvimento de software, a qualidade pode ser constituída por um conjunto de atributos desejáveis para que o produto atenda às necessidades de seus usuários. Esses atributos servem como guia para a descrição e avaliação da qualidade Rocha et al. (2001).

O modelo de qualidade, definido na norma ISO/IEC (1998), sintetiza esses atributos em seis características principais:

- *Funcionalidade*: refere-se à capacidade do software de oferecer funcionalidades que satisfaçam às necessidades explícitas e implícitas dos usuários. Suas subca-

racterísticas são: adequação, precisão, interoperabilidade, segurança de acesso e conformidade.

- *Confiabilidade*: refere-se à capacidade do software de manter seu nível de desempenho quando utilizado sob condições específicas. Suas subcaracterísticas são: maturidade, tolerância a falhas, recuperabilidade e conformidade.
- *Usabilidade*: refere-se à capacidade que o produto possui de ser entendido, aprendido, utilizado e ser atraente para o usuário; quando utilizado sob condições específicas. Suas subcaracterísticas são: inteligibilidade, apreensibilidade, operacionalidade, atratividade e conformidade.
- *Eficiência*: refere-se à capacidade do software de apresentar um desempenho apropriado à quantidade de recursos utilizados sob condições estabelecidas. Suas subcaracterísticas são: comportamento em relação ao tempo, comportamento em relação aos recursos e conformidade.
- *Manutenibilidade*: refere-se à facilidade de se fazer alterações no software. Suas subcaracterísticas são: analisabilidade, mutabilidade, estabilidade, testabilidade e conformidade.
- *Portabilidade*: refere-se à capacidade do software de ser transportado entre ambientes diferentes. Suas subcaracterísticas são: adaptabilidade, capacidade para ser instalado, coexistência, permutabilidade e conformidade.

No entanto, é importante que seja devotada atenção não só à qualidade do produto de software, como também à qualidade do processo de produção do software. É necessário que as visões de qualidade de processo e qualidade de produto caminhem juntas, pois o processo de produção é fundamental para a obtenção de produtos de qualidade (Rocha et al., 2001).

Um processo é um conjunto de passos através dos qual uma meta deve ser atingida (Paula Filho, 2003). Pode ser composto por vários elementos, tais como atividades, métodos, ferramentas e artefatos. Para se definir um processo é necessário

documentar o produto desse processo, os passos para atingi-lo, quantos são esses passos e por quem serão executados, além da especificação do que deve ser produzido a cada passo e o que será necessário para cada um deles (Paula Filho, 2003).

O processo de software é formado por um conjunto de passos parcialmente ordenados relacionados com conjuntos de artefatos, pessoas, recursos, estruturas organizacionais e restrições, tendo como objetivo produzir e manter os produtos de software requeridos (Lonchamp, 1993; Dowson et al., 1991). Informalmente, o processo de software pode ser compreendido como o conjunto de todas as atividades necessárias para transformar os requisitos do usuário em software (Humphrey, 1989; Osterweil, 1987).

Sua aplicação favorece aspectos ligados ao desenvolvimento do software, ao gerenciamento da qualidade e ao gerenciamento do projeto (Gruhn, 2002). A extração de dados sobre o processo e a transformação desses dados em conhecimento sobre o processo de produção permitem que o processo seja monitorado e melhorado, o que contribui para a melhoria da qualidade do processo e conseqüentemente para a melhoria do software produzido.

A Tecnologia de Processo de Software surgiu no final da década de 1980 e representa um importante passo em direção à melhoria da qualidade de software através de mecanismos que proporcionam o gerenciamento automatizado do desenvolvimento de software (Feiler and Humphrey, 1993).

2.2 Métodos dirigidos por planejamento

Os Métodos Dirigidos por Planejamento (MDP) são focados na disciplina. Utilizam uma abordagem sistemática onde o software é construído a partir dos requisitos até o código final passando por uma série de fases, todas muito bem documentadas (Boehm and Turner, 2004a). É devotada atenção às atividade de planejamento como forma de minimizar os erros de projeto na fase de implementação e como forma de melhorar a qualidade dos artefatos gerados durante o processo de desenvolvimento (Rundle and Dewar, 2006; Germain and Robillard, 2005).

Os MDP podem ser caracterizados pelos seguintes conceitos: 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 and Turner, 2004a). 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).

Em geral, os MDP estão associados à melhoria e aquisição de competência em processos, por isso os processos devem ser definidos, padronizados e melhorados incrementalmente (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, 2000). Os MDP propiciam previsibilidade, estabilidade e maior garantia sobre o produto de software (Boehm and Turner, 2004b).

2.3 Métodos Ágeis

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 em princípios que valorizam o suporte ao surgimento de novos requisitos, enfatizam a importância da comunicação (Segal, 2005) e realçam a competência individual como sendo um fator crítico para o sucesso de um projeto. (Cockburn and Highsmith, 2001). Esses princípios foram propostos em um manifesto (Alliance, 2001) definido pelo grupo *Agile Software Development Alliance* (Alliance, 2007).

Os Métodos Ágeis (MA) visam à rapidez na obtenção de resultados e ao au-

mento da adequabilidade a mudanças de requisitos (Boehm and Turner, 2004a), como forma de reduzir o custo de mudanças durante o projeto. Assumem que os requisitos vão mudar, e valorizam o suporte ao surgimento de novos requisitos. Priorizam o conhecimento tácito dos indivíduos (Highsmith and Cockburn, 2001) e enfatizam a importância da comunicação (Segal, 2005).

Nesses métodos, a comunicação utilizada entre os membros da equipe é realizada, principalmente, de maneira informal ao invés da comunicação formal e documentada (Boehm, 2002); pessoas podem compartilhar idéias mais rapidamente através de conversas informais do que através da leitura e escrita de documentos. 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: possibilidade de alterações e mudanças com facilidade (*embrace change*), ciclos de desenvolvimento curtos, entregas de versões frequentes, 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 and Turner, 2004a).

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 constante 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).

2.4 Métodos Híbridos

Podem ocorrer situações onde a disciplina e a documentação inerentes aos MDP tragam sobrecarga ao desenvolvimento. Um sistema simples, por exemplo, desen-

volvido 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 a longo prazo.

O nível de detalhamento do processo deve se adaptar às características de cada problema (Boehm and Turner, 2004a). 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.

Neste trabalho, método híbrido é aquele que combina características de MDP com características de MA, objetivando maior adequabilidade do processo ao tipo de desenvolvimento de software praticado (Boehm and Turner, 2004a). A vantagem dessa combinação reside no fato de que as abordagens resultantes podem se adequar mais ao contexto de um conjunto maior de projetos de desenvolvimento de software do que os MA e MDP isoladamente.

A obtenção de um método híbrido requer uma metodologia. Em (Boehm and Turner, 2004a), um método é proposto para balancear MA e MDP a partir da análise de determinadas características inerentes ao contexto de desenvolvimento de software. Este método será detalhado na seção 3

2.5 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

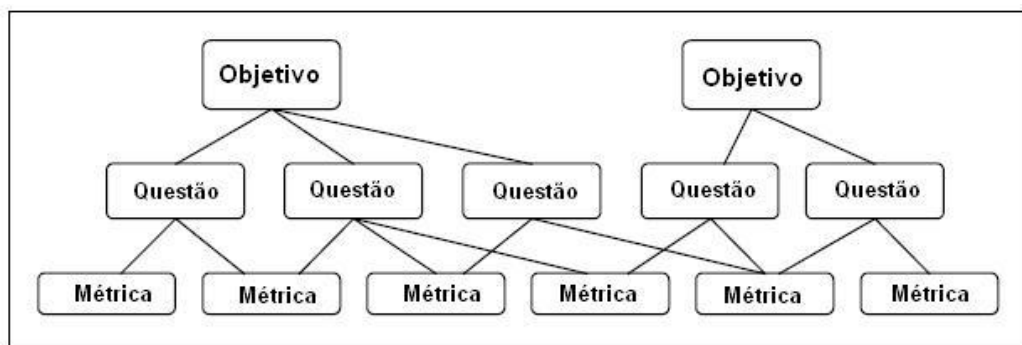
com modelos de processos de software, produtos e perspectivas de qualidade, baseadas nas necessidades de um projeto ou organização (Basili, 1992).

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 2.1, 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



Fonte: Adaptado de (Basili et al., 1994)

Figura 2.1: Estrutura Hierárquica da Abordagem GQM

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).

2.6 Trabalhos Relacionados

Nesta seção, são listados trabalhos presentes na literatura relacionados ao escopo deste trabalho ou que representam algum tipo de complemento para o desenvolvimento deste.

Em (Manzoni and Price, 2007), é proposta um framework - PRiMA-F para elaborar processos de desenvolvimento de software baseados em padrões organizacionais e critérios de risco. O objetivo do framework é a gerência de riscos em projetos de software, por meio da definição de processos de desenvolvimento adaptados a partir de um processo de software padrão da organização, de acordo com os riscos priorizados para o tipo de projeto.

Em (de Almeida Falbo et al., 2004) é apresentada uma abordagem de gerenciamento de conhecimento para apoiar a adaptação de processo; adotada em uma organização CMM nível 3. É utilizada uma ferramenta - ProKnowHow cujos objetivos são: apoiar a elaboração do processo padrão, coletar e disseminar o conhecimento adquirido durante a elaboração deste e apoiar sua atualização baseado no feedback de

projetos passados. O conceito de processo híbrido não é abordado. A adaptação do processo é realizada a partir do processo padrão com a adição de atividades que sejam pertinentes; e com base nas informações armazenadas sobre projetos passados; e não com base em um perfil abrangendo características sobre o desenvolvimento como um todo e sobre a equipe.

Em (Villela et al., 2004) é apresentada uma abordagem adotada por um meta-ambiente de desenvolvimento de software, para a definição de processos para organizações e projetos específicos dentro do contexto dos Ambientes de Desenvolvimento de Software Orientados a Organização. Esta abordagem está baseada na definição de processos em três níveis: processos padrão, especializado e instanciado; onde um processo de desenvolvimento se caracteriza pela descrição de uma seqüência de atividades, suas ferramentas de apoio, produtos de software gerados e recursos consumidos. O conceito de processo híbrido não é abordado. A adaptação do processo é realizada a partir de um processo padrão.

Cockburn (2000) propõe um framework para escolha da metodologia 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 em (Boehm and Turner, 2004a,b, 2003), onde é proposta uma abordagem baseada em risco para balancear Métodos Ágeis e Métodos Dirigidos por Planejamento visando a obtenção de métodos híbridos que se adequem melhor às necessidades, objetivos, restrições e prioridade de um projeto ou organização.

São identificados cinco fatores críticos (tamanho, criticalidade, dinamismo, competência pessoal e cultura da equipe) que podem ser utilizados para descrever uma organização ou projeto em função de suas características ágeis e disciplinadas. Este trabalho não define a forma como esses cinco fatores devem ser estimados. São caracterizados contextos nos quais cada um dos dois tipos de método é mais indicado e são examinados aspectos dos Métodos Ágeis e Métodos Dirigidos por Planejamento em exemplos e estudos de caso.

Capítulo 3

Metodologia

Este capítulo apresenta a metodologia utilizada para a construção do método proposto, cuja finalidade é obter requisitos para estimar características inerentes à equipe e ao desenvolvimento de software e que podem ser a base para a customização de processo de desenvolvimento de software.

As características consideradas neste trabalho são aquelas denominadas por Boehm and Turner (2004a) como sendo fatores críticos inerentes ao desenvolvimento e que são utilizados na adequação de Métodos Ágeis (MA) ou Métodos Dirigidos por Planejamento (MDP) a projetos de software. Essa adequação pode ser realizada pelo método proposto em (Boehm and Turner, 2004a) objetivando um processo híbrido que se adeque melhor a um contexto específico de desenvolvimento.

A realização desse trabalho foi composta por várias atividades como é mostrado na Figura 3.1:

1. *Atividade 1 (A1)*: aplicação da abordagem GQM na determinação dos requisitos necessários para se medir os cinco fatores críticos (Boehm and Turner, 2004a) que impactam o desenvolvimento de software.
2. *Atividade 2 (A2)*: elaboração de um questionário para coletar os dados relativos aos requisitos definidos.
3. *Atividade 3 (A3)*: validação do questionário através da aplicação do mesmo a uma equipe de desenvolvimento com a finalidade de se descobrir erros e ajustes

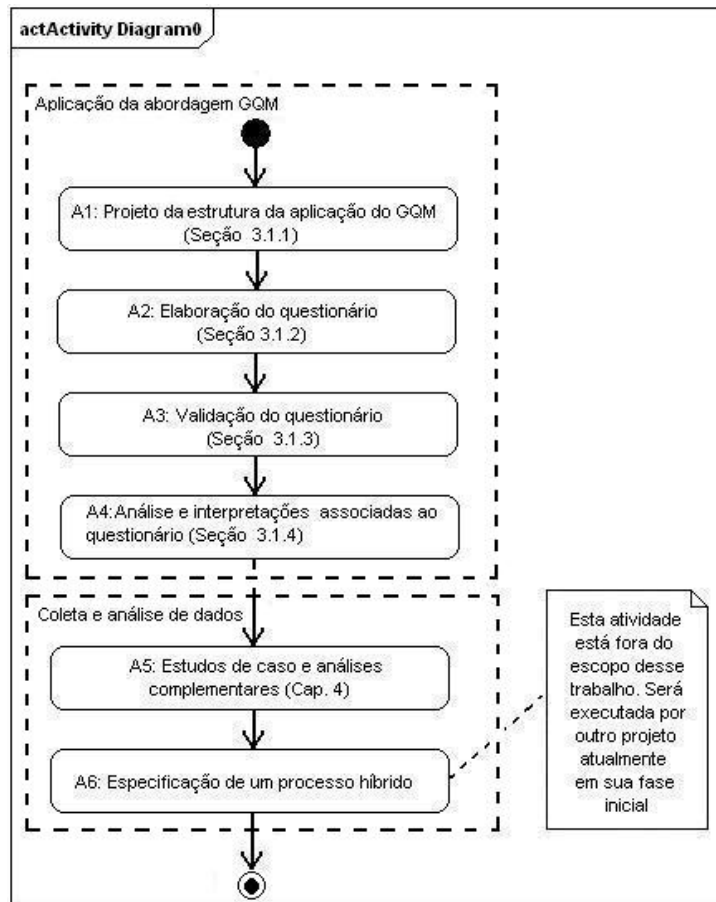


Figura 3.1: Fluxograma das etapas de desenvolvimento desse trabalho.

a serem feitos.

4. *Atividade 4 (A4)*: descrição das interpretações e análises associadas aos dados a serem coletados através do questionário.
5. *Atividade 5 (A5)*: realização de estudos de caso, aplicando-se o método proposto nesse trabalho a três equipes diferentes objetivando o delineamento do perfil de desenvolvimento de cada uma. Descrição de análises complementares sobre os fatores críticos.
6. *Atividade 6 (A6)*: especificação de um processo híbrido a partir da definição do perfil de cada equipe analisada com base na metodologia para elaboração de processo proposta por Boehm and Turner (2004a)

Nas subseções seguintes cada uma das atividades é detalhada.

Metodologia para customização de processo

Segundo Boehm and Turner (2004a), os processos de software podem ser adaptados, combinando-se características dos MA e MDP, dando origem aos processos híbridos. A vantagem dessa combinação reside no fato de que as abordagens resultantes podem se adequar mais ao contexto de um conjunto maior de projetos de desenvolvimento de software do que os MA e MDP isoladamente.

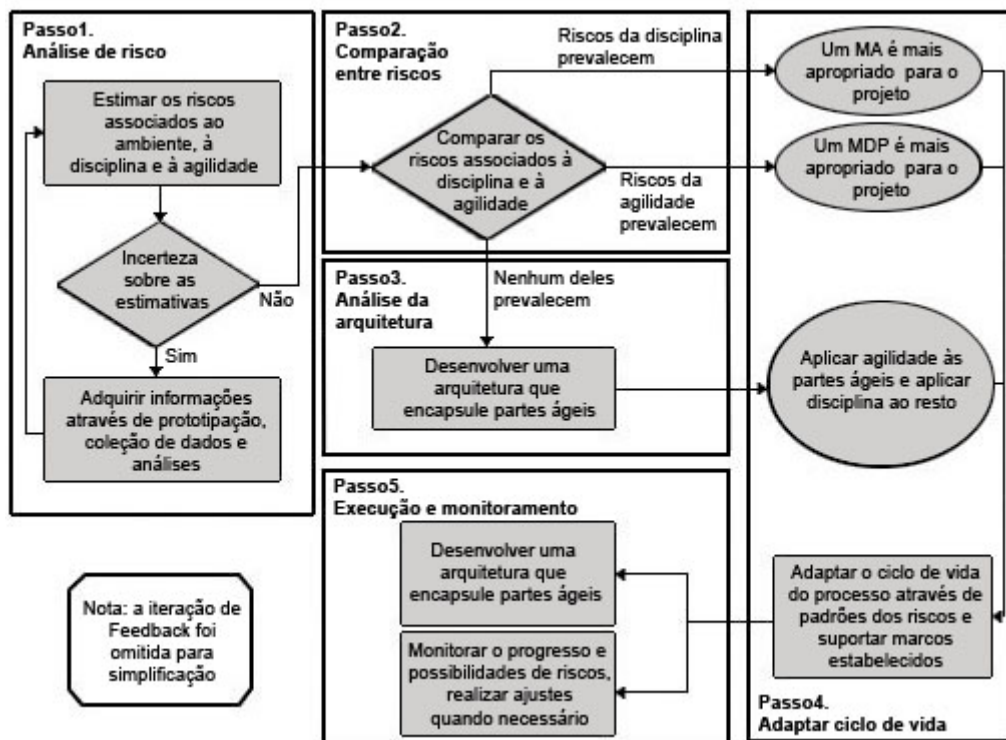
Em Boehm and Turner (2004a), um método é proposto para balancear MA e MDP. Este método é composto de cinco passos:

1. *Passo 1* - avaliar os riscos associados ao projeto com relação ao ambiente, à agilidade e à disciplina (os riscos são mostrados na Tabela 3.1). Se houver dúvidas sobre algum aspecto da avaliação é indicado despendar recursos através de prototipação, coleção de dados e análise para esclarecer esse aspecto.
2. *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.
3. *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.
4. *Passo 4* - desenvolver uma estratégia para todo o projeto para identificar os riscos. Identificar e integrar a estratégia para solução de cada risco.
5. *Passo 5* - Monitorar o progresso e os riscos apresentados pelo processo selecionado e realizar os ajustes necessários

Risco do ambiente: riscos que resultam do ambiente geral do projeto
- Incertezas relacionadas à tecnologia
- Muitos tipos de pessoas envolvidas para serem coordenados
- Complexidade do sistema
Riscos da agilidade: riscos que são específicos do uso de Métodos Ágeis
- Escalabilidade e criticalidade
- Uso de projeto simples
- Substituição de profissionais ou de clientes
- Profissionais sem habilidades para trabalhar com Métodos Ágeis
Riscos da disciplina: riscos que são específicos do uso de Métodos Dirigidos por Planejamento
- Mudanças rápidas
- Necessidade de resultados rápidos
- Surgimento de requisitos

Fonte: Extraído de Boehm and Turner (2004a)

Tabela 3.1: Riscos associados ao projeto.



Fonte: Adaptado de Boehm and Turner (2004a)

Figura 3.2: Resumo do método proposto por Boehm and Turner (2004a).

A Figura 3 mostra o resumo do método proposto por (Boehm and Turner, 2004a).

São definidos 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 (Boehm and Turner, 2004a):

- *Tamanho*: refere-se ao tamanho da equipe envolvida e ao esforço necessário no desenvolvimento.
- *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.
- *Dinamismo*: refere-se à possibilidade de mudanças nos requisitos em função de mudanças no ambiente do problema implicando em alterações no software.
- *Competência Pessoal*: refere-se às habilidades dos profissionais envolvidos no desenvolvimento. Um profissional pode ser associado a cinco níveis de habilidade diferentes os quais podem ser visualizados na Tabela 3.2.
- *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.

É importante ressaltar a origem da classificação dos níveis de habilidade dos profissionais mostrada na Tabela 3.2. A partir dos três níveis de entendimento: *Shu-Ha-Ri* (provenientes da arte marcial Aikido), onde:

- *Shu - sabedoria tradicional*: aprendizado sobre fundamentos, técnicas, heurísticas e provérbios.

- *Ha - quebrando com tradição*: descoberta de exceções à sabedoria tradicional, refletindo em sua verdade, descoberta de caminhos, técnicas e provérbios.
- *Ri - transcendência*: não há técnicas ou provérbios, todos os movimentos são naturais.

Cockburn (2002) identificou três níveis de entendimento 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.

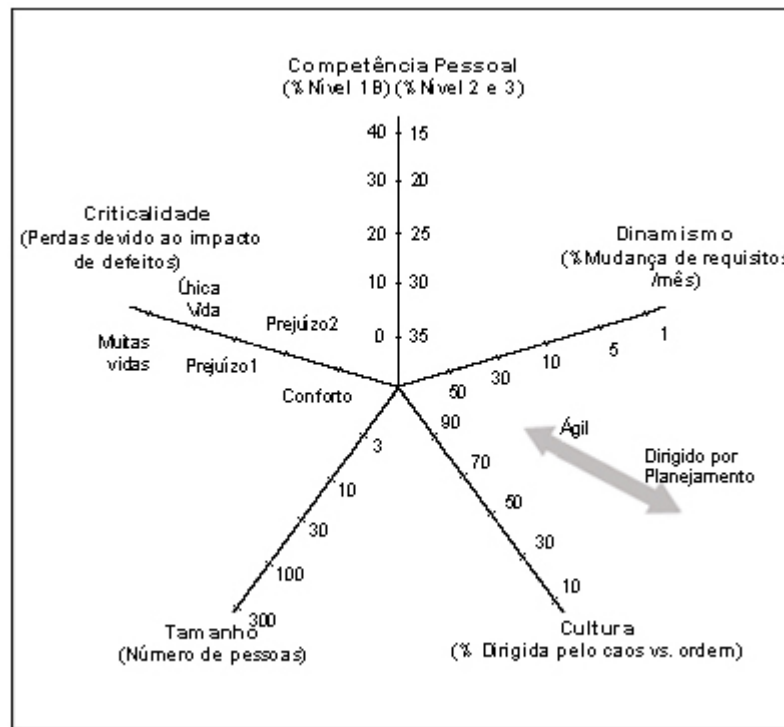
Em (Boehm and Turner, 2004a), o nível *1* foi dividido em *-1*, *1B* e *1A* para identificar diferenças entre MA e MDP e lidar com métodos ad-hoc. Assim tem-se os cinco níveis de habilidade mostrados na Tabela 3.2 e pelos quais os profissionais tem sua competência pessoal definida.

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.
1A	Com treinamento, é capaz de seguir os passos de um método discreto (dimensionar histórias para ajustar incrementos, definir padrões, realizar refatorações complexas). Com experiência, pode alcançar o nível 2.
1B	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 1A.
-1	Possui habilidades técnicas, mas não é capaz ou não está disposto a colaborar ou seguir métodos compartilhados.

Fonte: Extraído de Boehm and Turner (2004a)

Tabela 3.2: Classificação dos níveis de habilidade dos indivíduos.

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 3.3, 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



Fonte: Adaptado de Boehm and Turner (2004a)

PREJUÍZO 1 - prejuízo maior, capaz de provocar impacto negativo no projeto.

PREJUÍZO 2 - prejuízo menor.

Figura 3.3: Gráfico polar com a representação dos fatores críticos.

híbrido.

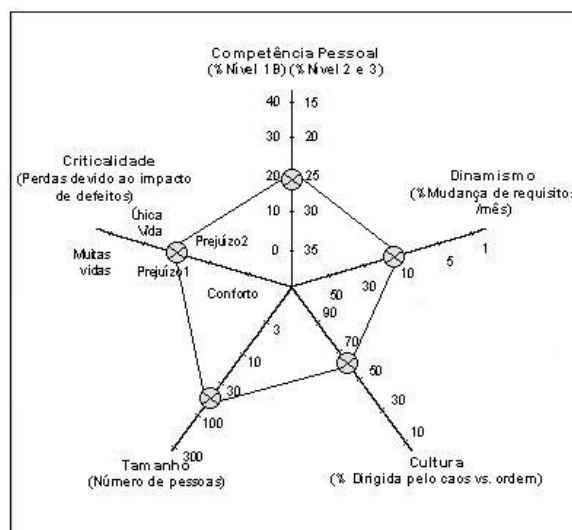
Essa área resultante no gráfico pode delimitar uma porção mais interna do gráfico ou uma porção que se aproxime da área total do gráfico. No primeiro caso, tem-se uma área que revela um perfil onde os riscos são amenizados com adoção de MA. No segundo caso, tem-se uma área que revela um perfil onde os 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 pra o desenvolvimento.

Exemplo da aplicação do método (Fonte: Adaptado de Boehm and Turner (2004a)):

Em (Boehm and Turner, 2004a), são descritos alguns exemplos da aplicação da metodologia detalhada acima. Na figura 3.4 é mostrado a área de análise para a determinação da proporção entre MA e MDP, referente a um desses exemplos - SupplyChain.com. Trata-se de uma empresa especializada no desenvolvimento de sistemas de gerência de estoque para grandes companhias que trabalham com redes complexas de fornecimento e distribuição.

Pela área do gráfico, observa-se que os riscos associados à aplicação de agilidade são escalabilidade, criticalidade e o uso de projeto simples. Os riscos associados à aplicação da disciplina são mudanças rápidas, necessidade de resultados rápidos e surgimento de requisitos.

Tanto uma equipe ágil experiente, como uma equipe adepta a processos como RUP ou TSP e com alguns traços de agilidade é indicada ao desenvolvimento. De acordo com o ambiente e a cultura da SupplyChain.com, uma abordagem ágil é selecionada. Se o contexto de desenvolvimento apresentasse maior estabilidade e uma rotatividade maior de profissionais na equipe, uma abordagem planejada seria mais apropriada.



Fonte: Adaptado de Boehm and Turner (2004a)

Figura 3.4: Gráfico polar referente à SupplyChain.com.

O método proposto neste trabalho visa a obter requisitos que permitam a definição de um perfil do desenvolvimento de software praticado por uma equipe e que sirva de base para a obtenção de um processo híbrido utilizando-se o método proposto por Boehm and Turner (2004a).

3.1 Aplicação da abordagem GQM

3.1.1 Projeto da estrutura da aplicação do GQM

Como foi descrito na seção 2.5, 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 cinco fatores críticos, mostrados no gráfico da Figura 3.3, 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 cinco 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 3.1.2 a atividade de elaboração do questionário é detalhada. A seguir, é mostrada a aplicação da abordagem GQM na quantificação de cada um dos fatores críticos.

Tamanho

Na Tabela 3.3 é mostrada a aplicação da abordagem GQM na quantificação do fator crítico *tamanho*. O objetivo é descobrir quais são as métricas necessárias para estimar

o tamanho da equipe envolvida no desenvolvimento de software.

TAMANHO	
Objetivo:	<i>Estimar.</i>
Característica:	<i>Tamanho.</i>
Objeto:	<i>Equipe de desenvolvimento.</i>
Ponto de Vista:	<i>Gerente de processo ou projeto.</i>
Questão:	Qual é o tamanho da equipe de desenvolvimento?
Métrica(s):	- Tamanho médio da equipe de desenvolvimento por projeto. - Número médio de projetos aos quais cada indivíduo se dedica ao mesmo tempo.

Tabela 3.3: Aplicação de GQM na quantificação do fator crítico *tamanho*

Dinamismo

Na Tabela 3.4 é mostrada a aplicação da abordagem GQM na quantificação do fator crítico *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.

DINAMISMO	
Objetivo:	<i>Estimar.</i>
Característica:	<i>Periodicidade das alterações sofridas pelos requisitos de software em função de alterações sofridas pelo contexto do problema.</i>
Objeto:	<i>Requisitos do software.</i>
Ponto de Vista:	<i>Gerente de processo ou projeto.</i>
Questão:	Qual é a maior causa para mudança nos requisitos do sistema?
Métrica(s):	- % 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.

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?
Métrica(s):	- % de mudança de requisitos por mês (do total estabelecido até o momento).

Tabela 3.4: Aplicação de GQM na quantificação do fator crítico *dinamismo*.

Cultura

Na Tabela 3.5 é mostrada a aplicação da abordagem GQM na quantificação do fator crítico *cultura* como foi proposto em (Basili et al., 1994). 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.

CULTURA	
Objetivo:	<i>Estimar.</i>
Característica:	<i>Hábito de trabalhar de forma organizada.</i>
Objeto:	<i>Equipe de desenvolvimento.</i>
Ponto de Vista:	<i>Gerente de processo ou projeto.</i>
Questão:	O ciclo de desenvolvimento de software inclui quais das seguintes fases: análise, especificação, projeto, codificação e teste?
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.

Tabela 3.5: Aplicação de GQM na quantificação do fator crítico *cultura*.

Criticalidade

Na Tabela 3.6 é mostrada a aplicação da abordagem GQM na quantificação do fator crítico *criticalidade* 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.

CRITICALIDADE	
Objetivo:	<i>Estimar.</i>
Característica:	<i>Gravidade das perdas ocasionadas por falhas no software a ser desenvolvido para resolver o problema.</i>
Objeto:	<i>Problema a ser resolvido pelo desenvolvimento de software.</i>
Ponto de Vista:	<i>Gerente de processo ou projeto.</i>
Questão:	Qual é a criticalidade dos problemas resolvidos pelo desenvolvimento de software?
Métrica(s):	- % 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.

- % de problemas onde falhas no sistema causam transtorno,
mas não acarretam prejuízos substanciais.

Tabela 3.6: Aplicação de GQM na quantificação do fator crítico *criticalidade*.

Competência Pessoal

Na Tabela 3.7 é mostrada a aplicação da abordagem GQM na quantificação do fator crítico *competência pessoal*. 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 estabelecidos na Tabela 3.2.

COMPETÊNCIA PESSOAL	
Objetivo:	<i>Estimar</i>
Característica:	<i>Competência pessoal dos integrantes da equipe</i>
Objeto:	<i>Equipe de desenvolvimento</i>
Ponto de Vista:	<i>Gerente de processo ou projeto.</i>
Questão:	Qual é a porcentagem de profissionais pertencentes à equipe de desenvolvimento que possuem outra área de atuação?
Métrica(s):	- % de membros da equipe de desenvolvimento cuja a área de atuação principal não é a área da computação.
Questão:	Qual é a divisão do número de membros da equipe entre as diversas atividades de desenvolvimento do software?
Métrica(s):	- % 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):	<ul style="list-style-type: none"> - Tempo médio de experiência dos indivíduos cuja a 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 a área de atuação principal é a área de computação e que desempenham atividades gerenciais. - Tempo médio de experiência dos indivíduos cuja a 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 a 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 a 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.

Tabela 3.7: Aplicação de GQM na quantificação do fator crítico *competência pessoal*.

3.1.2 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:

- *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 a extrair informações relativas aos indivíduos pertencentes à equipe. Deve ser respondido por cada um dos membros da equipe

independente da sua função.

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

- 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 adequem 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).

3.1.3 Validação do questionário

Além das considerações citadas anteriormente, um teste de adequação permite que pequenos erros, perguntas mal formuladas, opções de respostas ambíguas e opções insuficientes de respostas sejam detectadas. Este teste consiste em realizar uma aplicação dos questionários a uma equipe de desenvolvimento de software, diferente daquelas analisadas pelo foco do trabalho.

Foi realizado um teste de adequação com uma equipe de desenvolvimento de software pertencente à Central de Processamento de Dados (CPD) da Universidade

Federal de Viçosa. A CPD é responsável pelas atividades de manutenção dos computadores do campus, gerência da rede da UFV, manutenção de diversos websites além de possuir um setor voltado para o desenvolvimento de sistemas utilizados por professores, alunos e funcionários da instituição. Trata-se de sistemas para controle acadêmico, financeiro, administrativo, de recursos humanos, de controle de estoque dentre outros.

O desenvolvimento de software realizado pela CPD é dividido entre dois grupos. A primeiro grupo é formado por profissionais efetivados pela UFV. O segundo grupo é formado por profissionais terceirizados cujo vínculo com a CPD tem um tempo determinado. Dentro desses grupos os profissionais se dividem em equipes menores que se dedicam a projetos diferentes.

O teste de adequação foi realizado com uma dessas equipes menores, responsável por desenvolver um dos sistemas acadêmicos da UFV - o *PVANet*. Trata-se de uma ferramenta usada pelos professores e alunos onde são disponibilizados materiais, notícias e calendários das disciplinas. A equipe é formada por:

- um profissional formado na área de Computação com pós-graduação e mestrado também na área de Computação, com treze anos de experiência em atividades de desenvolvimento de software e gerência de projetos. Atua como gerente de projetos das equipes pertencentes à CPD.
- dois indivíduos da área da Computação, um graduando e o outro graduado que exercem atividades de análise, projeto, codificação e teste dos sistemas.

Durante a aplicação do questionário, os respondentes foram incentivados a relatar qualquer item do questionário que não estivesse claro e a fazer sugestões que julgassem pertinentes.

O gerente de projetos foi selecionado para responder Questionário Gerencial por estar na equipe há mais tempo, desempenhar atividades gerenciais e acompanhar o cotidiano do desenvolvimento da equipe.

O Questionário Individual foi respondido por todos os membros da equipe, inclusive pelo gerente de projetos. Os principais pontos levantados pelos respondentes

na aplicação dos dois questionários foram:

- a necessidade de mais opções de respostas em determinadas perguntas a fim de cobrir todos os casos possíveis de acordo com o mundo real.
- o esclarecimento de siglas presentes nos enunciados das perguntas, tais como RUP (*Rational Unified Process*), PSP (*Personal Software Process*), TSP (*Team Software Process*) e CMM (*Capability Maturity Model*), que mesmo estando amplamente difundidas na literatura, poderiam não ser conhecidas pelo respondente.

Após a realização do teste de adequação, o Questionário Gerencial e o Questionário Individual foram ajustados conforme as observações realizadas pelos respondentes.

As perguntas presentes nas versões finais dos questionários podem ser visualizadas na seção 3.1.4.

3.1.4 Questionário

Questionário Gerencial

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

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

- a) até 3 pessoas
- b) entre 3 e 10 pessoas
- c) entre 10 e 30 pessoas
- d) entre 30 e 100 pessoas
- e) entre 100 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 - 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 100% das vezes
- c) 50% a 80% das vezes
- d) 50% das vezes
- e) 30% a 50% das vezes
- f) 0% a 30% 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 100% das vezes
- c) 50% a 80% das vezes
- d) 50% das vezes

e) 30% a 50% das vezes

f) 0% a 30% das vezes

g) nunca

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

a) 100% das vezes

b) 80% a 100% das vezes

c) 50% a 80% das vezes

d) 50% das vezes

e) 30% a 50% das vezes

f) 0% a 30% das vezes

g) nunca

4 - Com que freqüência 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

5 - 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) 20% a 40%

c) 40% a 60%

d) 60% a 80%

e) 80% a 100%

6 - Com que freqüência 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 freqüência
- h) nunca

7 - 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

8 - 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

9 - 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

10 - 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

11 - 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

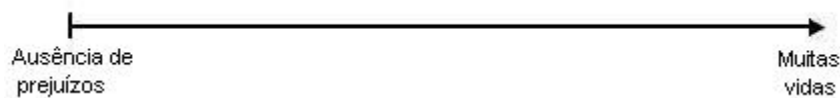
c) especificação de requisitos

d) teste

e) projeto

12 - 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).



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/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/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/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 atuo/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 atuo/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 atuo/atuei nessa atividade

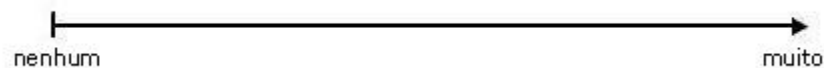
3 - Com relação ao CMM (*Capability Maturity Model*), você (marque apenas uma opção):

- a) desconhece
- b) já ouviu falar
- c) conhece
- d) já trabalhou em um ambiente que utilizasse CMM

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?



3.1.5 Análise e Interpretações associadas ao questionário

As informações referentes a cada um dos fatores críticos 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 crítico, agrupá-las e mapeá-las para que estejam de acordo com as escalas do gráfico da Figura 3.3.

A atividade de agrupar e mapear 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.

Nas seções seguintes, o agrupamento e o mapeamento das informações para cada fator crítico são detalhados.

Tamanho

O fator crítico *tamanho* é quantificado através da *Pergunta 1* do questionário gerencial mostrado na seção 3.1.4. Especificamente nesse caso, as opções de resposta para essa pergunta são intervalos de valores que já estão de acordo com a escala do eixo *tamanho* do gráfico.

A partir da resposta obtida, é possível categorizar a equipe de acordo com seu tamanho. Este trabalho estabelece a seguinte classificação:

- **Equipe Pequena:** equipe que contenha até 10 pessoas;
- **Equipe Média:** equipe que contenha de 10 a 20 pessoas;
- **Equipe Grande:** equipe que contenha mais de 20 pessoas;

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).

Dinamismo

A extração de informações para quantificar o fator crítico *dinamismo* é feita através das *perguntas 3, 4 e 5* do questionário gerencial mostrado na seção 3.1.4. As respostas a essas perguntas não estão de acordo com a escala do gráfico da Figura 3.3. É

necessário associar e mapear as respostas a estas três perguntas para gerar um valor que represente o % 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:

$$\%deMudancaDeRequisitos = \frac{RespostaPergunta3A}{100} * RespostaPergunta5 * Freqüencia$$

Onde *Freqüência* é:

- 0: se a Resposta à Pergunta4 é *nunca*
- 30: se a Resposta à Pergunta4 é *diariamente*
- 4,29: se a Resposta à Pergunta4 é *semanalmente*
- 1,00: se a Resposta à Pergunta4 é *mensalmente*
- 0,50: se a Resposta à Pergunta4 é *bimestralmente*
- 0,33: se a Resposta à Pergunta4 é *trimestralmente*
- 0,17: se a Resposta à Pergunta4 é *semestralmente*
- 0,08: se a Resposta à Pergunta4 é *anualmente*

A equação acima produz um valor para o % de mudanças de requisitos em função de mudanças no contexto do problema. Esse valor é obtido a partir das respostas às *Perguntas 3A* e *5* do questionário gerencial mostrado na seção 3.1.4, e do valor *Freqüência*. O valor *Freqüência* é obtido em função da resposta à *Pergunta 4* 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 4* for *diariamente*, então o valor para *Freqüência* deve ser: 30. Se a resposta for, por exemplo, *bimestralmente*, então o valor para *Freqüência* deve ser: 0,50. O mesmo raciocínio aplica-se às outras respostas.

Como as opções de resposta da *Pergunta 5* 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 o % de mudanças de requisitos/ mês, em função de mudanças no contexto do problema.

Com a definição desse intervalo, pode-se fazer uma aproximação da situação da equipe para o eixo *dinamismo* do gráfico da Figura 3.3. Além disso, pode-se cate-

gorizar o contexto dos problemas resolvidos pela equipe de acordo com o dinamismo apresentado. Este trabalho estabelece a seguinte classificação:

- **Muito dinâmico:** contexto de desenvolvimento que apresenta um % de mudanças / mês igual ou superior a 10%;
- **Pouco dinâmico:** contexto de desenvolvimento que apresenta um % de mudanças / mês inferior a 10%;

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).

Cultura

Para quantificar o fator crítico *cultura*, 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.

As *perguntas 9, 10 e 11* do questionário gerencial 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 A. 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. Isto significa que, mesmo que seus membros possuam alguma predileção por trabalhar

em 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 3.3.

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. Para que este valor fique de acordo com a escala do eixo *cultura* do gráfico da Figura 3.3 é necessário observar apenas que a escala do gráfico está estruturada em função do % dirigida pelo caos vs. ordem, logo se o percentual obtido anteriormente for, por exemplo 25%, então no gráfico deverá ser marcada o ponto referente a 75% (referente ao percentual orientado pelo caos).

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

- **Orientada pela ordem:** equipe onde o percentual obtido é igual ou superior a 50%.
- **Orientada pelo caos:** equipe onde o percentual obtido é inferior a 50%.

Criticalidade

O fator crítico *criticalidade* é quantificado através da *Pergunta 12* do questionário gerencial mostrado na seção 3.1.4. 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.

Uma escala caracteriza as variáveis cujo levantamento pretende-se fazer, e em

particular, 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).

O enunciado da *Pergunta 12* 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 muitas 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* do gráfico.

A escala oferecida tem 10 centímetros. Para realizar o mapeamento, ela é dividida em intervalos de 2 centímetros, onde cada um dos intervalos corresponde a cada uma das situações enumeradas no eixo *criticalidade* do gráfico da Figura 3.3. De acordo com o ponto marcado, é identificado o intervalo e a situação correspondente. O mapeamento entre os intervalos e as possíveis situações de perdas é mostrado na Figura 3.1.5.

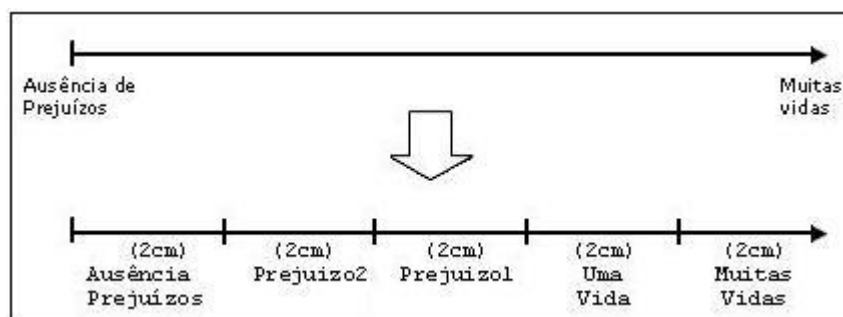


Figura 3.5: Mapeamento dos intervalos da escala para as possíveis situações de perdas

A partir do resultado do mapeamento, este trabalho estabelece a seguinte classificação para a criticalidade dos problemas:

- **Ausência de prejuízo:** refere-se aos casos onde a presença de falhas no sistema desenvolvido não implicam em prejuízo de qualquer tipo. Trata-se de uma

situação de conforto. Esta classificação é feita a partir de respostas em que o ponto foi marcado dentro do intervalo correspondente a *ausência de prejuízo*.

- **Risco de prejuízo financeiro:** refere-se aos casos onde a presença de falhas no sistema desenvolvido implicam em prejuízos financeiros. Esta classificação é feita a partir de respostas em que o ponto foi marcado dentro do intervalo correspondente a *prejuízo2* ou *prejuízo1*.
- **Risco de vida:** refere-se aos casos onde a presença de falhas no sistema desenvolvido implicam em risco à vida humana. Esta classificação é feita a partir de respostas em que o ponto foi marcado dentro do intervalo correspondente a *uma vida* ou *muitas vidas*.

Competência Pessoal

Para quantificar o fator crítico *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 são utilizadas para classificar cada membro da equipe de acordo com os cinco níveis de habilidade mostrados na Tabela 3.2.

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 3.2. A atribuição de pesos às respostas é mostrada no Apêndice B.

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. Isso contribui para que o profissional da área de Computação possua, naturalmente, um nível mais alto de habilidade de acordo com a Tabela 3.2.

- **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 *Pergunta 2* têm o propósito de analisar este fator. Esses seis 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 3.2.

- **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* 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 3.2.

- **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 3.2 como é mostrado a seguir:

Nível de Habilidade	Pontuação
Profissional 3	≥ 31
Profissional 2	25 A 30
Profissional 1A	20 A 24
Profissional 1B	13 A 19
Profissional -1	≤ 12

Tabela 3.8: *Pontuação* necessária para cada nível de habilidade.

- **Classificação da equipe de acordo com a competência pessoal de seus membros** De acordo com a metodologia proposta por Boehm and Turner (2004a) e detalhada na seção 3, o fator crítico *competência pessoal* é estimado em função da proporção de profissionais com nível de habilidade 1B, 2 e 3 de acordo com a Tabela 3.2.

Desta forma, após realizar a classificação do nível de habilidade de cada membro da equipe, é necessário calcular qual é o percentual de profissionais da equipe com níveis de habilidade 1B, 2 e 3; gerando a proporção de profissionais na equipe de acordo com seus respectivos níveis de habilidade. O resultado obtido já pode ser expresso de acordo com a escala do eixo *Competência Pessoal* do gráfico da Figura 3.3

Desta forma, após realizar a classificação do nível de habilidade de cada membro da equipe, é necessário calcular qual é o percentual de profissionais da equipe com níveis de habilidade 1B, 2 e 3, obtendo assim a proporção de profissionais de acordo com seus respectivos níveis de habilidade. O resultado obtido fica de acordo com a escala do eixo *competência pessoal* do gráfico da Figura 3.3

A partir da proporção obtida, este trabalho estabelece a seguinte classificação para a equipe:

- **Equipe com perfil mais adequado à agilidade:** equipe que contenha pelo menos 35% de seus membros pertencentes ao nível de habilidade 2 ou 3.
- **Equipe com perfil menos adequado à agilidade:** outros casos.

Essa classificação baseia-se principalmente na demanda dos MA por um alto percentual de profissionais com nível mais alto de habilidade. Os defensores dos MA dizem ser capazes de obter sucesso em uma equipe ágil se houver a proporção de cinco profissionais de nível 1A para cada profissional de nível 2 (Boehm and Turner, 2003). Isso implica em um percentual aproximado de pelo menos 20% de profissionais com nível de habilidade 2 na equipe. Sendo que ainda podem haver profissionais de nível 3 na equipe e que o valor mais interno do eixo *competência pessoal* no gráfico da Figura 3.3 é 0% de profissionais com nível 1A e 35% de profissionais com nível 2 ou 3, então 35 foi o valor selecionado para classificar as equipes de acordo com o percentual de profissionais com nível 2 ou 3 .

Capítulo 4

Resultados e Discussões

4.1 Estudos de caso

Foram realizados alguns estudos de casos para este trabalho. A metodologia proposta foi aplicada a três equipes diferentes obtendo-se o perfil do contexto de desenvolvimento inerente a cada equipe.

As equipes analisadas pertencem ao meio acadêmico. 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 a 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 C.

4.1.1 Formato de apresentação dos estudos de casos

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

- **Nome:** nome da equipe ou do grupo/empresa que a mantém.
- **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.
- **Sobre a aplicação do questionário:** informações sobre a aplicação do questionário.
- **Tabela de classificação:** classificação da equipe em relação aos cinco fatores críticos. Realizada a partir dos dados coletados e das interpretações associadas ao questionário descritas na seção 3.1.5.

EQUIPE X	
Tamanho da equipe	-
Dinamismo do problema	-
Cultura da equipe	-
Criticalidade dos problemas	-
Competência pessoal da equipe	-

Tabela 4.1: Classificação da equipe analisada.

- **Gráfico Polar:** representação gráfica do perfil da equipe, obtida através dos dados coletados e das interpretações descritas na seção 3.1.5.
- **Observações:** informações complementares que enriqueçam o estudo de caso (quando existirem).
- **Considerações sobre o processo:** sugestões a serem consideradas na customização do processo para a equipe analisada.

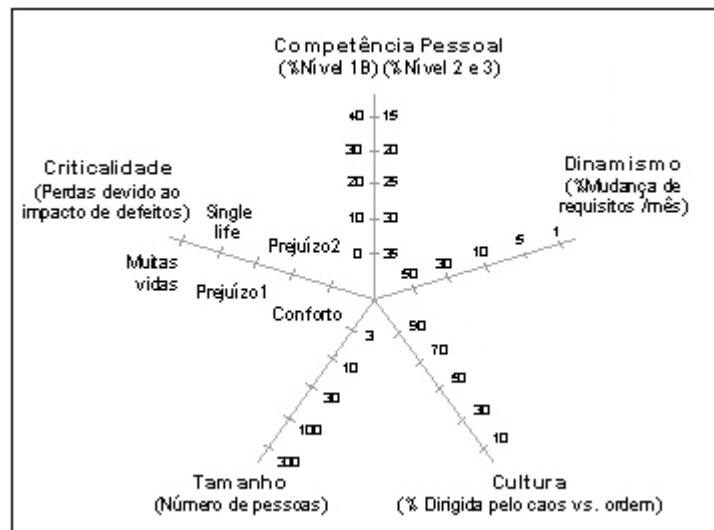


Figura 4.1: Gráfico Polar

4.1.2 Equipe1

- **Nome:** Bioma
- **Descrição:** equipe pertencente ao Projeto Bioma, cujo propósito é o desenvolvimento de um software de mesmo nome (*Biodiversity data Manager*). Trata-se de um sistema para gerenciar e tratar todas as informações obtidas por indivíduos e equipes em projetos biológicos de pesquisa com cupins. Toda informação armazenada pelo sistema é ligada a um espécime. Além de espécime, o sistema suporta o conceito de coleta e localidade (de forma recursiva ou não).
- **Composição da equipe:** a equipe é formada por aproximadamente cinco pessoas sendo que algumas delas são substituídas ocasionalmente. Os integrantes da equipe são:
 - um professor pertencente ao Departamento de Biologia Animal da Universidade Federal de Viçosa. Esse profissional tem noções sobre conceitos de bancos de dados e programação adquiridos através do desenvolvimento de pequenas aplicações. Possui vasto conhecimento sobre o conteúdo coberto pelo software. É o fundador do projeto e atua como gerente geral da equipe participando, eventualmente, de decisões ligadas ao projeto do software.

- um Doutor na área de Entomologia. Possui um grande conhecimento sobre programação e bancos de dados. Além de também possuir vasto conhecimento sobre o conteúdo coberto pelo software. Presente na equipe desde sua criação, atua como gerente de projetos .
 - dois graduandos do curso de Ciência da Computação que desenvolvem as atividades propriamente ditas de projeto, codificação e teste do sistema. Mantém um vínculo com a equipe na forma de bolsistas de projeto de pesquisa, sendo que a permanência na equipe acompanha o tempo de duração da bolsa (geralmente um ano, podendo ocorrer a renovação). Ocasionalmente, ocorre a substituição desses membros por outros indivíduos com mesmo perfil em função de término da bolsa ou finalização do curso de graduação por parte do graduando.
 - eventualmente, um aluno de intercâmbio é incluído na equipe. Esses indivíduos permanecem na equipe por um tempo reduzido e, em função disso, sua colaboração para o desenvolvimento do sistema é bem mais restrita.
- **Sobre a 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 indivíduo dentro da equipe e identificar qual membro estava mais apto a responder o questionário gerencial.

O questionário gerencial foi respondido pelo Doutor na área de Entomologia. Ele foi o selecionado por pertencer à equipe desde sua criação, atuar como gerente de projetos e acompanhar o cotidiano do desenvolvimento praticado.

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.

- **Observações:**

- O eixo competência pessoal mostra um ponto fora do gráfico. Isso é justi-

EQUIPE1	
Tamanho da equipe	Pequena
Dinamismo do problema	Pouco dinâmico
Cultura da equipe	Orientada pela ordem
Criticalidade dos problemas	Risco de Prejuízo Financeiro
Competência pessoal da equipe	Equipe com perfil menos adequado à agilidade

Tabela 4.2: Classificação da Equipe 1 segundo os cinco fatores críticos.

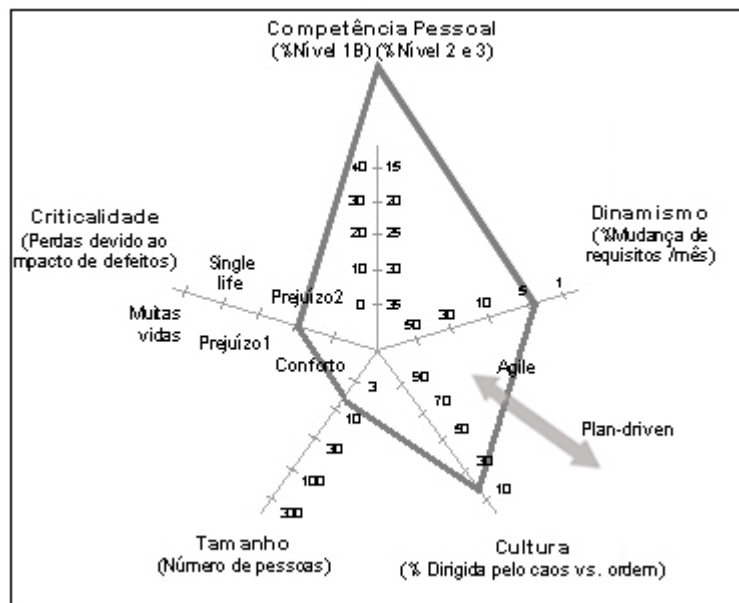


Figura 4.2: Gráfico polar referente ao perfil de desenvolvimento da Equipe 1

ficado pela ausência de profissionais do nível 2 e 3 nessa equipe, sendo que a maioria de seus integrantes enquadram-se no nível 1B, de acordo com os níveis de experiência exibidos na Tabela 3.2.

- **Considerações sobre o processo:** através dos resultados obtidos e da área do gráfico da Figura 4.2, sugere-se que o processo a ser customizado para essa equipe seja predominantemente disciplinado, com um certo grau de agilidade.

Os fatores críticos *criticalidade* e *tamanho* apresentam valores mais internos no gráfico o que pode implicar em uma equipe com perfil mais voltado para processos ágeis. No entanto, a análise da *competência pessoal* da equipe revela a ausência de profissionais dos níveis 2 e 3 e a presença de profissionais -1. Isso

inviabiliza o uso de abordagens ágeis puras, as quais demandam profissionais com um nível maior de habilidade.

Mas o processo deve ser combinado com um certo grau de agilidade para se adaptar melhor ao *dinamismo* apresentado, aproximadamente 5% de mudanças de requisitos por mês.

4.1.3 Equipe2

- **Nome:** GPRH - Hidrologia
- **Descrição:** equipe pertencente ao Grupo de Pesquisa em Recursos Hídricos (GPRH) do departamento de Engenharia Agrícola da Universidade Federal de Viçosa. Trata-se de um grupo voltado para o desenvolvimento de tecnologias e sistemas que visem ao adequado planejamento e manejo dos recursos hídricos. Essa equipe é responsável pelo desenvolvimento de sistemas na área de Hidrologia, e em geral, tem como cliente agências do governo que solicitam a implementação de software para resolver problemas bem definidos e para os quais já existem soluções bem estabelecidas na literatura especializada. Os sistemas desenvolvidos são usados no planejamento e gerenciamento de recursos hídricos e falhas no software podem implicar não só em prejuízos financeiros como em graves prejuízos ambientais.
- **Composição da equipe:** a equipe é formada por quatro pessoas sendo que algumas delas são substituídas ocasionalmente. Os integrantes da equipe são:
 - um profissional formado em Ciência da Computação. Ele desempenha o papel de gerente de projeto e também atividades de análise, projeto e codificação. Ele já pertence à equipe há alguns anos e nesse período adquiriu um bom nível de experiência sobre a área de conhecimento coberta pelo software.
 - três graduandos do curso de Ciência da Computação que desenvolvem as atividades de codificação e teste do sistema; seguindo as orientações do

gerente de projeto. Mantém um vínculo com a equipe na forma de bolsistas de projeto de pesquisa, sendo que a permanência na equipe acompanha o tempo de duração da bolsa (geralmente um ano, podendo ocorrer renovação). Ocasionalmente, ocorre a substituição desses membros por outros indivíduos com mesmo perfil em função de término da bolsa ou finalização do curso de graduação por parte do graduando.

- **Sobre a aplicação do questionário:** assim como no estudo de caso relativo à equipe 1, foi realizado um contato prévio com a equipe 2 para esclarecer o papel de cada indivíduo dentro da equipe e qual deles seria o membro selecionado para responder o questionário gerencial.

O questionário gerencial foi respondido pelo gerente de projeto e o questionário individual foi respondido por todos os membros, inclusive pelo gerente de projeto que também desempenha atividades ligadas diretamente ao desenvolvimento.

EQUIPE2	
Tamanho da equipe	Pequena
Dinamismo do problema	Pouco dinâmico
Cultura da equipe	Orientada pelo caos
Criticalidade dos problemas	Risco de Prejuízo Financeiro
Competência pessoal da equipe	Equipe com perfil menos adequado à agilidade

Tabela 4.3: Classificação da Equipe 2 segundo os cinco fatores críticos.

- **Observações:** nenhuma
- **Considerações para a customização do processo:** através dos resultados obtidos e da área do gráfico da Figura 4.3, sugere-se que o processo a ser customizado para essa equipe seja predominantemente ágil.

Os fatores críticos *criticalidade*, *tamanho*, *competência pessoal* e *cultura* apresentam valores mais internos ao gráfico formando uma área de análise que reforça o caráter ágil que o processo deve ter para ser mais aderente ao contexto de desenvolvimento da equipe.

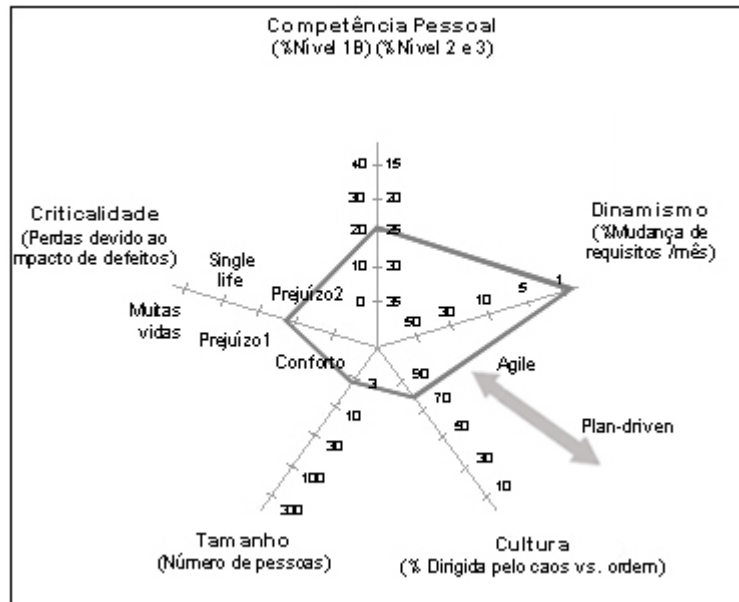


Figura 4.3: Gráfico polar referente ao perfil de desenvolvimento da Equipe 2

Apesar de o *dinamismo* ser baixo (próximo de 0% de mudanças de requisitos por mês), ele não inviabiliza a escolha de um processo mais ágil.

4.1.4 Equipe3

- **Nome:** GPRH - Engenharia de Conservação do Solo
- **Descrição:** equipe também pertencente ao Grupo de Pesquisa em Recursos Hídricos (GPRH) do departamento de Engenharia Agrícola da Universidade Federal de Viçosa. Essa equipe é responsável pelo desenvolvimento de sistemas na área de Engenharia de Conservação do Solo, e em geral, implementa software para dar apoio a trabalhos de pesquisa em nível de mestrado e doutorado. As soluções a serem implementadas não são bem definidas e muitas vezes são justamente o foco das pesquisas realizadas nos trabalhos de mestrado e doutorado. Assim, os principais clientes dos sistemas desenvolvidos são os realizadores desses trabalhos de pesquisa e cuja área de conhecimento não é a Computação. Em função da natureza dos sistemas desenvolvidos por essa equipe, o contexto de desenvolvimento acaba revelando diferenças significativas em relação ao contexto

de desenvolvimento da Equipe 2 (GPRH - Hidrologia) e por isso as duas equipes foram tratadas como equipes independentes.

- **Composição da equipe:** a equipe é formada por quatro pessoas sendo que algumas delas são substituídas ocasionalmente. Os integrantes da equipe são:

- um profissional formado em outra área de atuação. Apesar de não ser originalmente da área da Computação, ele é o membro que está há mais tempo na equipe e exerce a função de gerente de projeto, desempenhando também atividades ligadas à análise, projeto e codificação dos sistemas.
- três graduandos do curso de Ciência da Computação que desenvolvem as atividades de codificação e teste do sistema; seguindo as orientações do gerente de projeto. Mantém um vínculo com a equipe na forma de bolsas de projeto de pesquisa, sendo que a permanência na equipe acompanha o tempo de duração da bolsa (geralmente um ano, podendo ocorrer renovação). Ocasionalmente, ocorre a substituição desses membros por outros indivíduos com mesmo perfil em função de término da bolsa ou finalização do curso de graduação por parte do graduando.

- **Sobre a aplicação do questionário:** assim como no estudo de caso relativo à equipe 1, foi realizado um contato prévio com a equipe 3 para esclarecer o papel de cada indivíduo dentro da equipe e qual deles seria o membro selecionado para responder o questionário gerencial.

O questionário gerencial foi respondido pelo gerente de projeto e o questionário individual foi respondido por todos os membros, inclusive pelo gerente de projeto que também desempenha atividades ligadas diretamente ao desenvolvimento.

- **Observações:**

- O eixo competência pessoal mostra um ponto fora do gráfico. Isso é justificado pela ausência de profissionais do nível 2 e 3 nessa equipe, de acordo com os níveis de experiência exibidos na Tabela 3.2.

EQUIPE3	
Tamanho da equipe	Pequena
Dinamismo do problema	Pouco dinâmico
Cultura da equipe	Orientada pelo caos
Criticalidade dos problemas	Risco de Prejuízo Financeiro
Competência pessoal da equipe	Equipe com perfil menos adequado à agilidade

Tabela 4.4: Classificação da Equipe 3 segundo os cinco fatores críticos.

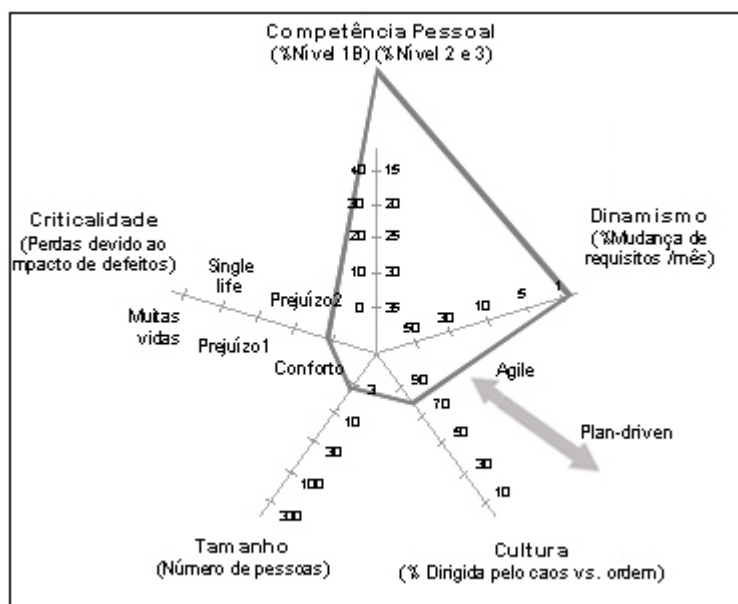


Figura 4.4: Gráfico polar referente ao perfil de desenvolvimento da Equipe 3

- **Considerações sobre o processo:** através dos resultados obtidos e da área do gráfico da Figura 4.4, sugere-se que o processo a ser customizado para essa equipe seja predominantemente disciplinado.

Os fatores críticos *criticalidade*, *cultura* e *tamanho* apresentam valores mais internos no gráfico o que pode implicar em uma equipe com perfil mais voltado para processos ágeis. No entanto, a análise da *competência pessoal* da equipe revela a ausência de profissionais dos níveis 2 e 3 e a presença de profissionais -1. Isso inviabiliza o uso de abordagens ágeis puras, as quais demandam profissionais com um nível maior de habilidade.

O dinamismo apresentado é baixo, próximo de 0% de mudanças de requisitos

por mês, e se adapta bem a sugestão de um processo disciplinado.

4.1.5 Resultados comparativos entre as equipes

Os resultados da aplicação do método proposto nesse trabalho às três equipes analisadas apresentam muitas similaridades entre si. Isso se deve, principalmente ao fato de elas pertencerem a um nicho específico de desenvolvimento de software, que produz sistemas voltados para o ambiente acadêmico. Trata-se de sistemas ligados a objetivos acadêmicos ou de pesquisa, não envolvendo múltiplos papéis de usuários externos. Desenvolvido por equipes pequenas onde parte dos desenvolvedores não tem como área de atuação principal a Ciência da Computação, mas têm formação especializada em suas respectivas áreas.

Na Tabela 4.5 são sumarizados os resultados referentes às equipes.

	EQUIPE1	EQUIPE2	EQUIPE3
Tamanho da equipe	Pequena	Pequena	Pequena
Dinamismo do problema	Pouco dinâmico	Pouco dinâmico	Pouco dinâmico
Cultura da equipe	Orientada pela ordem	Orientada pelo caos	Orientada pelo caos
Criticalidade dos problemas	Risco de Prejuízo	Risco de Prejuízo	Risco de Prejuízo
Criticalidade	Financeiro	Financeiro	Financeiro
Competência pessoal da equipe	Equipe com perfil menos adequado à agilidade	Equipe com perfil menos adequado à agilidade	Equipe com perfil menos adequado à agilidade

Tabela 4.5: Classificação de todas as equipes analisadas.

4.2 Análise complementar dos fatores

Durante o desenvolvimento da metodologia e realização dos estudos de casos, foram observados alguns pontos importantes e que merecem um pouco mais de atenção para serem devidamente discutidos.

4.2.1 Escala do eixo competência pessoal no gráfico polar

Cada ponto na escala do eixo *competência pessoal* representa uma proporção, entre habilidades de profissionais pertencentes ao nível *1B* e aos níveis *2* e *3* de um equipe. Essas proporções não totalizam 100% referente ao total de membros da equipe, indicando que o percentual referente outros membros pode ser composto de profissionais de nível *-1* e *1B*. Essas proporções são restritivas para refletir a realidade de todas as equipes de desenvolvimento.

É importante lembrar que os níveis *2* e *3* são os mais altos para um profissional, de acordo com a Tabela 3.2. A presença de profissionais com nível de habilidade *2* e *3*; e até *1B* não pode ser considerada uma realidade para a maioria das equipes de desenvolvimento de software, principalmente quando se trata de equipes pequenas, que são responsáveis por grande parte do desenvolvimento de software. Segundo o MCT05 (2005), as micro e pequenas empresas são responsáveis por acolher aproximadamente 70% do profissionais ligados ao setor de software no país. Observa-se uma realidade similar também em outros países. Segundo o Departamento do Trabalho dos Estados Unidos, 80% dos profissionais de TI estão em pequenas empresas (Harris et al., 2007).

Assim, esta escala torna-se restritiva para ser utilizada na análise de todos os possíveis tipos de equipe de desenvolvimento. Esse trabalho propõe uma adaptação na escala do eixo *competência pessoal* para que a mesma seja expressa em termos da proporção entre profissionais dos níveis *-1*, *1B*; e dos níveis *1A*, *2*, *3*; como é mostrado na Figura 4.5.

Cada ponto na nova escala representa uma proporção de profissionais pertencentes a dois grupos: um dos grupos é constituído por profissionais com nível de habilidade *-1* ou *1B*; o outro grupo é constituído por profissionais com nível de habilidade *1A*, *2* ou *3*.

As razões para a escolha dessa distribuição dos níveis de habilidade pelos grupos são comentadas a seguir:

- Os Métodos Ágeis demandam a presença de um número mais alto de profissionais com nível de habilidade alto. Os defensores dos Métodos Ágeis dizem ser capazes

4.2.2 Interferência entre os fatores críticos

Pode-se observar que alguns dos cinco fatores críticos interferem entre si em determinadas situações. Abaixo são discutidas algumas interferências observadas.

Criticalidade e Cultura:

Observa-se que o fator crítico *criticalidade* pode influenciar o fator crítico *cultura* em contextos de desenvolvimento onde a criticalidade do problema é alta. O aumento da criticalidade (isto é, o aumento da gravidade das perdas ocasionadas por falhas no sistema desenvolvido) implica em uma exigência maior de qualidade e confiabilidade por parte dos sistemas desenvolvidos. Isso demanda mais organização e estruturação durante o desenvolvimento do software, o que impulsiona a cultura da equipe a ser mais orientada pela ordem do que pelo caos.

Em contextos de desenvolvimento onde a criticalidade do problema é baixa, nada pode ser afirmado sobre a interferência entre o fator *criticalidade* e o fator *cultura*. Equipes com cultura tanto orientada pela ordem como pelo caos têm condições de desenvolver software para problemas com criticalidade baixa.

A figura 4.6 mostra que se a criticalidade do problema aumenta, a cultura tende para um ponto mais externo no gráfico. Se a criticalidade diminui, nada pode ser afirmado.

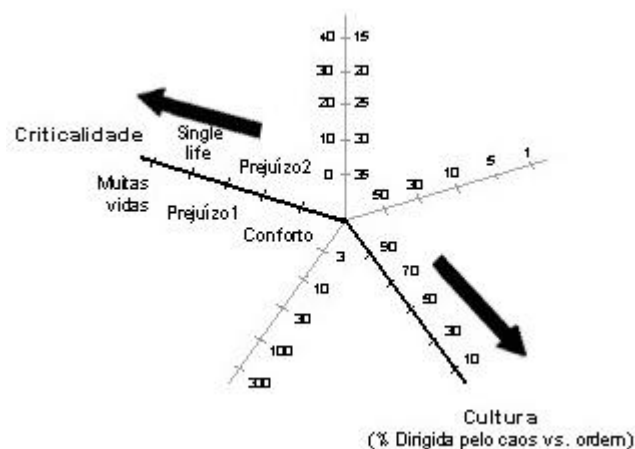


Figura 4.6: Representação da interferência entre o fator *Criticalidade* e o fator *Cultura*

Tamanho e Cultura:

Observa-se que o fator crítico *tamanho* pode influenciar o fator crítico *cultura* quando a equipe é grande. O aumento do tamanho da equipe implica em uma maior necessidade de organização e estruturação do desenvolvimento para viabilizar a gerência do trabalho de seus membros e evitar erros de comunicação. Todos os membros da equipe devem estar ambientados com técnicas e métodos que formalizem e documentem o desenvolvimento dos projetos e a comunicação entre a equipe, o que impulsiona a cultura da equipe a ser mais orientada pela ordem do que pelo caos.

Quando a equipe é pequena, nada pode ser afirmado sobre a interferência entre o fator *tamanho* e o fator *cultura*. Equipes pequenas podem possuir uma cultura tanto orientada pela ordem como pelo caos.

A figura 4.7 mostra que se o tamanho da equipe aumenta, a cultura tende para um ponto mais externo no gráfico. Se o tamanho da equipe diminui, nada pode ser afirmado.

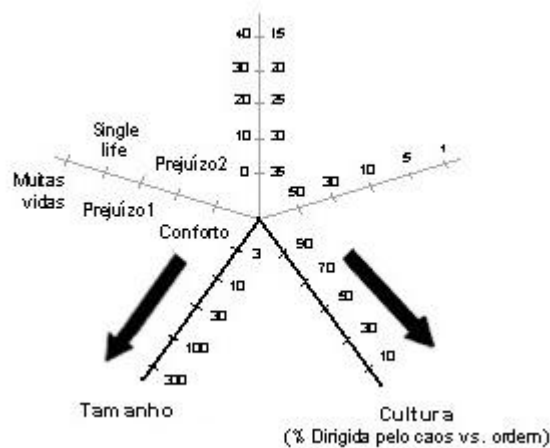


Figura 4.7: Representação da interferência entre os fatores críticos *Tamanho* e *Cultura*

Dinamismo e Cultura:

Observa-se que o fator crítico *dinamismo* pode influenciar o fator crítico *cultura* em contextos de desenvolvimento que apresentem um alto dinamismo. Nesse caso, uma cultura orientada pela ordem pode não ser a mais apropriada, pois incluem atividades

destinadas à organização do desenvolvimento e que implicam em trabalho adicional quando ocorrem mudanças nos requisitos. Já uma cultura orientada pelo caos impõe menos sobrecarga quando essas mudanças ocorrem, permitindo que a equipe esteja mais preparada para lidar com mudanças constantes nos requisitos.

Em contextos onde o dinamismo é baixo, nada pode ser afirmado sobre a interferência entre o fator *dinamismo* e o fator *cultura*. Equipes com cultura tanto orientada pela ordem como pelo caos têm condições de desenvolver software que apresentem uma taxa mais baixa de mudança nos requisitos.

A figura 4.8 mostra que se o dinamismo aumenta, a cultura tende para um ponto mais interno no gráfico. Se o dinamismo diminui, nada pode ser afirmado.

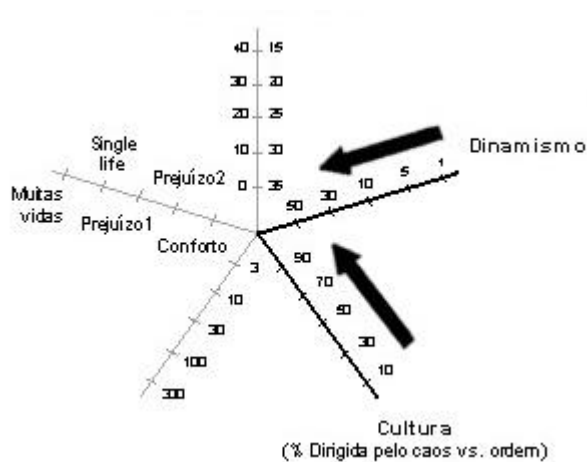


Figura 4.8: Representação da interferência entre os fatores críticos *Dinamismo* e *Cultura*

Competência Pessoal e Cultura

Observa-se que o fator crítico *competência pessoal* pode influenciar o fator crítico *cultura* em contextos de desenvolvimento onde a equipe é constituída por um percentual alto de profissionais com nível de habilidade 2 ou 3. Esta situação viabiliza uma cultura orientada pelo caos, visto que estes profissionais possuem mais experiência para lidar com situações adversas (com e sem precedentes) e possuem conhecimento tácito significativo podendo tomar decisões sobre o desenvolvimento de projetos de software sem a necessidade de ter como suporte uma vasta documentação.

A figura 4.9 mostra que se a equipe é constituída por um percentual alto de profissionais com nível de habilidade 2 ou 3, a cultura tende para um ponto mais interno no gráfico. Se a equipe é constituída por um percentual alto de profissionais com nível de habilidade 1B, nada pode ser afirmado.

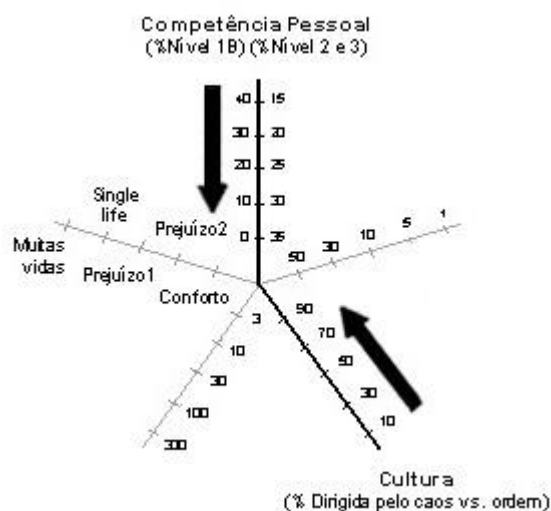


Figura 4.9: Representação da interferência entre o fator *Competência Pessoal* e o fator *Cultura*.

Observa-se que o fator crítico *cultura* pode influenciar o fator crítico *competência pessoal* em contextos de desenvolvimento onde a cultura da equipe é orientada pela ordem. Uma cultura orientada pela ordem pode impor a necessidade de um percentual mais alto de profissionais com nível de habilidade mais baixo e que serão responsáveis por seguir os procedimentos e políticas estabelecidas para o desenvolvimento.

A figura 4.10 mostra que se a cultura da equipe é orientada pela ordem, percentual de profissionais com nível de habilidade mais baixo na equipe aumenta. Se a cultura da equipe é orientada pelo caos, nada pode ser afirmado.

4.2.3 Fatores determinantes

Nas seções anteriores foram discutidas algumas interferências observadas entre os fatores críticos e que podem influenciar na elaboração de um processo.

Independente das interferências entre os fatores críticos, observa-se também que



Figura 4.10: Representação da interferência entre o fator *Cultura* e o fator *Competência Pessoal*.

alguns desses fatores, em situações específicas, podem ser determinantes na escolha ou elaboração de um processo, independente da configuração dos outros fatores. São eles:

Criticalidade

O fator crítico *criticalidade* pode ser determinante no caso em que falhas no sistema implicam em risco a vidas humanas, isto é, alta criticalidade. Nesse caso, independente da configuração dos outros fatores críticos, faz-se necessário a aplicação de MDP que apresentem documentação detalhadas e procedimentos rígidos, a fim de garantir a corretude e a confiabilidade por parte dos sistemas desenvolvidos.

Tamanho

O fator crítico *tamanho* pode ser determinante quando trata-se de equipes muito grandes. Equipes com mais de cinquenta profissionais, por exemplo, devem utilizar MDP, pois a aplicação de MA tornaria inviável o gerenciamento do trabalho de todos os profissionais.

Competência Pessoal

O fator crítico *Competência Pessoal* pode ser determinante quando a equipe analisada é composta, em sua maioria, por profissionais com nível de habilidade -1 e 1B. Isso inviabiliza a aplicação de MA visto que esses métodos demandam um maior quantidade de profissionais com nível de habilidade mais alto (Boehm and Turner, 2003).

Capítulo 5

Conclusões e perspectivas

Esse trabalho demonstrou que é possível fundamentar a escolha da técnica ou método a ser usado durante a aplicação de um processo, utilizando-se uma análise sistemática do contexto de desenvolvimento.

A principal contribuição foi a proposta de uma metodologia para determinar os requisitos necessários à quantificação de parâmetros relativos a equipes de desenvolvimento de software e que permitam a customização de processos de desenvolvimento de software aderentes às suas necessidades. O processo obtido dessa forma certamente permitirá que a equipe explore melhor seu potencial de trabalho, melhorando o ambiente organizacional e diminuindo o estresse na equipe.

Através da aplicação do método proposto aos estudos de casos, foi obtido um perfil do desenvolvimento praticado para cada equipe analisada.

Algumas observações importantes foram derivadas dos estudos de caso e originaram outras contribuições. Foi proposta uma alteração na escala do eixo *competência pessoal* do gráfico 3.3 visando uma maior adaptação da escala a todos os tipos de equipes existentes. Além disso, foram identificadas interferências entre alguns dos cinco fatores críticos em situações específicas. Observou-se também que alguns fatores apresentam um comportamento determinante sobre a seleção do processo em situações específicas. Nesses casos, independente da configuração dos outros fatores críticos, um processo é selecionado ou inviabilizado por um desses fatores determinantes.

5.1 Trabalhos Futuros

Como trabalho futuro, sugere-se a elaboração de processos híbridos que se adequem ao perfil de desenvolvimento das equipes analisadas nos estudos de casos desse trabalho. Além disso, sugere-se a análise qualitativa e quantitativa dos benefícios trazidos pela aplicação do processo híbrido elaborado. Para isso, podem ser estabelecidas métricas que permitam a realização desse tipo de análise.

Outra sugestão é construir um framework para a elaboração de processos híbridos. Para a construção desse framework, pode-se selecionar um método ágil e um método dirigido por planejamento que estejam presentes na literatura e que sirvam como base para a elaboração do processo híbrido, através da combinação de suas partes. A elaboração do processo pode ter como ponto de partida as características contidas no perfil do desenvolvimento gerado através da metodologia proposta por este trabalho.

Apêndice A

Atribuições de pesos

A seguir é mostrada a atribuição de pesos às respostas referentes às *questões 9, 10 e 11* 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.

Pergunta	Resposta	Peso
9 - 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 tarefas desempenhadas por cada membro	1
10 - 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
11 - 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

Tabela A.1: Pesos associados às respostas às *questões 9, 10 e 11* do questionário gerencial

Apêndice B

Atribuições de pesos

A seguir 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 um score para cada membro da equipe de desenvolvimento.

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	0
	6 a 12 meses	0
	1 a 3 anos	0
	3 a 5 anos	0
	mais de 5 anos	0
	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	0
	6 a 12 meses	0
	1 a 3 anos	0
	3 a 5 anos	0
	mais de 5 anos	0
	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, você (marque apenas uma opção):	desconhece	0
	já ouviu falar	2
	conhece	4
	já trabalhou em um ambiente que utilizasse CMM	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

Tabela B.1: Pesos associados às respostas às *questões 1, 2A, 2B, 2C, 2D, 2E, 2F, 3 e 4* do questionário individual

Apêndice C

Respostas dos questionários

Nas tabelas a seguir são mostradas as respostas obtidas através da aplicação dos questionários gerencial e individual às três equipes descritas na seção 4.1.

EQUIPE1 - Bioma													
Perguntas - Questionário Gerencial													
1	2	3a	3b	3c	4	5	6	7	8	9	10	11	12
b	c	d	e	f	f	b	f	g	d	a+b+c+d	a+b+d	a+b+c+d	5,4cm

Perguntas - Questionário Individual										
Respondente	1	2a	2b	2c	2d	2e	2f	3	4	5
1	b	c	a	c	f	f	c	a	a	a
2	b	c	c	d	d	d	d	b	b	a
3	b	a	a	a	f	f	a	b	a	b
4	a	c	c	c	c	c	c	c	c	a
5	a	c	b	b	d	a	f	d	b	a
6	a	c	a	c	d	c	f	d	c	a

Tabela C.1: Respostas fornecidas pela equipe1

EQUIPE2 - GPRH - Hidrologia													
Perguntas - Questionário Gerencial													
1	2	3a	3b	3c	4	5	6	7	8	9	10	11	12
a	a	g	a	g	d	a	f	h	d	a	—	a+b d	4,4cm

Perguntas - Questionário Individual										
Respondente	1	2a	2b	2c	2d	2e	2f	3	4	5
1	a	e	f	e	e	e	c	c	c	b
2	a	f	f	f	c	c	f	b	a	a
3	a	f	f	f	a	a	f	a	a	a
4	a	c	f	c	c	c	f	c	c	a

Tabela C.2: Respostas fornecidas pela equipe2

EQUIPE3 - GPRH - Engenharia de Conservação do Solo													
Perguntas - Questionário Gerencial													
1	2	3a	3b	3c	4	5	6	7	8	9	10	11	12
a	a	g	a	g	d	a	f	h	d	a	—	a+b d	2,2cm

Perguntas - Questionário Individual										
Respondente	1	2a	2b	2c	2d	2e	2f	3	4	5
1	b	a	f	a	c	c	a	a	a	a
2	a	a	f	c	c	c	f	b	c	b
3	a	f	f	f	b	b	f	c	c	b
4	a	f	f	a	b	f	f	a	a	a

Tabela C.3: Respostas fornecidas pela equipe3

Referências Bibliográficas

- Agerfalk, P. J. and Fitzgerald, B. (2006). Flexible and distributed software processes: old petunias in new bowls? *Commun. ACM*, 49(10):26–34.
- Alliance, A. (2001). Manifesto for agile software development. <http://agilemanifesto.org/>. Acessado em 04/01/2007.
- Alliance, A. (2007). Agile alliance. <http://www.agilealliance.org>. Acessado em 04/01/2007.
- Ambler, S. W. (2002). *Agile modeling effective practices for eXtreme programming and the unified process*. Wiley, New York.
- Basili, V. R. (1992). Software modeling and measurement: The Goal Question Metric paradigm. Technical report, UMIACS-TR-92-96. CS-TR-2956.
- Basili, V. R., Caldiera, G., and Rombach, D. (1994). Goal Question Metric Approach. *Encyclopedia of Software Engineering*, pages 528–532.
- Beck, K. (2000). *Extreme Programming Explained: Embrace Change*. Reading, MA: Addison Wesley Longman, Inc.
- 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*, pages 316–325, New York, NY, USA. ACM Press.
- Boehm, B. (2002). Get ready for agile methods. *IEEE Computer*, 35(1):64–69.

- Boehm, B. and Turner, R. (2003). Using risk to balance agile and plan-driven methods. *IEEE Computer*, 36(6):57–66.
- Boehm, B. and Turner, R. (2004a). *Balancing Agility and Discipline: A Guide for the Perplexed*. Addison-Wesley, Boston.
- Boehm, B. and Turner, R. (2004b). Balancing agility and discipline: Evaluating and integrating agile and plan-driven methods. In *ICSE '04: Proceedings of the 26th International Conference on Software Engineering*, pages 718–719, Washington, DC, USA. IEEE Computer Society.
- Cockburn, A. (2000). Selecting a project's methodology. *IEEE Software*, 17(4):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):131–133.
- Constantine, L. L. (2001). Methodological agility. *Software Development*, pages 67–69.
- de Almeida Falbo, R., Borges, L. S. M., and Valente, F. F. R. (2004). Using knowledge management to improve software process performance in a cmm level 3 organization. In *Fourth International Conference on Quality Software, 2004. QSIC 2004. Proceedings.*, pages 162 – 169. IEEE Computer Society.
- de Andrade Martins, G. (2005). *Estatística geral e aplicada*. Atlas.
- Dowson, M., Nejme, B., and Riddle, W. (1991). Fundamental software process concepts. In *Proceedings of First European Workshop Software Process Modeling*, pages 15–37.
- Feiler, P. and Humphrey, W. (1993). A structured conceptual and terminological framework for the software process engineering. In *Software process development and enactment: Concepts and definitions*, pages 28–40.
- Firlej, M. and Hellens, D. (1991). *Knowledge Elicitation*. Prentice Hall.

- Germain, E. and Robillard, P. N. (2005). Engineering-based processes and agile methodologies for software development: a comparative case study. *J. Syst. Softw.*, 75(1-2):17–27.
- Gruhn, V. (2002). Process-centered software engineering environments a brief history and future challenges. *Annals of Software Engineering*, 14:363–382.
- Harris, M., Aebischer, K., and Klaus, T. (2007). The whitewater process: software product development in small it businesses. *Commun. ACM*, 50(5):89–93.
- Highsmith, J. and Cockburn, A. (2001). Agile software development: the business of innovation. *Computer*, 34(9):120–127.
- Humphrey, W. S. (1989). The software engineering process: definition and scope. In *Proceedings of the 4th international software process workshop on Representing and enacting the software process*, pages 82–83, New York, NY, USA. ACM Press.
- 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.
- ISO/IEC (1998). ISO/IEC 9126-1.2, Information Technology - Software product quality - part 1: Quality model.
- Kruchten, P. (2000). *The Rational Unified Process: An Introduction*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.
- Lonchamp, J. (1993). A structured conceptual and terminological framework for the software process engineering. In *Proceedings of Second International Conference on the Software Proces*, pages 41–53.
- Manzoni, L. V. and Price, R. T. (2007). A framework for tailoring software process. In *SEKE'2007: International Conference on Software Engineering and Knowledge Engineering*.

- MCT05 (2005). Pesquisa sobre qualidade e produtividade no setor de software – 2005. Disponível em: http://www.mct.gov.br/upd_blob/0002/2674.pdf. Acessado em 05/07/2007.
- Methods, A. D. (2007). Scrum. <http://www.controlchaos.com/>. Acessado em 04/01/2007.
- Osterweil, L. (1987). Software processes are software too. In *Proceedings of the 9th international conference on Software Engineering*, Monterey, California, United States.
- Paula Filho, W. P. (2003). *Engenharia de Software: Fundamentos, Métodos e Padrões*. LTC, Rio de Janeiro, segunda edition.
- Pressman, R. S. (2005). *Software Engineering: A Practitioner's Approach*, volume 6. McGraw-Hill.
- Rocha, A. R. C., Maldonado, J. C., and Weber, K. C. (2001). *Qualidade de Software: Teoria e Prática*. Prentice Hall, São Paulo, SP, BRASIL.
- Rundle, P. J. and Dewar, R. G. (2006). Using return on investment to compare agile and plan-driven practices in undergraduate group projects. In *ICSE '06: Proceeding of the 28th international conference on Software engineering*, pages 649–654, New York, NY, USA. ACM Press.
- SBC (2006). Grandes desafios da pesquisa em computação no brasil - 2006 - 2016. SBC.
- Segal, J. (2005). When software engineers met research scientists: a case study. *Empirical Software Engineering*, 10(4):517–536.
- 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 IIISIC'06: 5ta. Jornada Iberoamericana de Ingeniería del Software e Ingeniería del Conocimiento*, Puebla, Puebla, Mexico.

- Solingen, R. V. and Berghout, E. (1999). *The Goal/Question/Metric Method A Practical Guide for Quality Improvement of Software Development*. McGraw-Hill Publishing Company, England.
- Solingen, V., Basili, V. R., Caldiera, G., and Rombach, D. (2002). Goal Question Metric (GQM) approach. *Encyclopedia of Software Engineering*.
- Villela, K., Santos, G., Montoni, M., Berger, P., Figueiredo, S., Mafra, S., Rocha, A. R., and Travassos, G. H. (2004). Iii simposio brasileiro de qualidade de software. In *Anais do SBQS*, pages 22–37. Sociedade Brasileira de Computação.