

ANÁLISE DE REQUISITOS DA REVISÃO PELOS PARES NA APRENDIZAGEM

Universidade Federal de Santa Catarina
Programa de Pós-Graduação em
Engenharia de Produção

ANÁLISE DE REQUISITOS DA REVISÃO PELOS PARES NA APRENDIZAGEM

Álvaro Sampaio Corrêa Neto

Dissertação apresentada ao
Programa de Pós-Graduação em Engenharia de Produção da
Universidade Federal de Santa Catarina
como requisito parcial para obtenção
do título de Mestre em
Engenharia de Produção

Florianópolis
2004

Álvaro Sampaio Corrêa Neto

ANÁLISE DE REQUISITOS DA REVISÃO PELOS PARES NA APRENDIZAGEM

Esta Dissertação foi julgada adequada para obtenção do título de Mestre em Engenharia de Produção e aprovada em sua forma final pelo Programa de Pós-Graduação em Engenharia de Produção da Universidade Federal de Santa Catarina.

Florianópolis, 20 de dezembro de 2004.

Prof. Edson Pacheco Paladini, Dr.

Coordenador do Curso

BANCA EXAMINADORA

Prof. Vinícius Medina Kern, Dr.

Orientador

Prof^a. Joyce Munarski Pernigotti, Dra.

Prof^a. Luciana Martins Saraiva, Dra.

Prof. Roberto Carlos dos Santos Pacheco, Dr.

Agradecimentos

Agradeço primeiramente ao professor Vinícius Medina Kern pela competência, paciência e dedicação sempre exemplar na orientação deste trabalho.

A todos os professores e colaboradores que fazem a excelência do Programa de Pós-graduação em Engenharia de Produção.

Gostaria de agradecer a minha família e namorada que souberam me incentivar nas horas de dificuldade, trazendo sempre palavras de esperança e motivação.

E, por fim, a todas as pessoas que participaram, direta ou indiretamente, na conclusão deste trabalho.

“Eu agora diria a nós, como educadores e educadoras: ai daqueles e daquelas, entre nós, que pararem com sua capacidade de sonhar, de inventar a sua coragem de denunciar e de anunciar. Ai daqueles e daquelas que, em lugar de visitar de vez em quando o amanhã, o futuro, pelo profundo engajamento com o hoje, com o aqui e com o agora, ai daqueles que, em lugar desta viagem constante ao amanhã, se atrelarem a um passado de exploração e rotina. “

Paulo Freire

Sumário

LISTA DE FIGURAS	IX
LISTA DE QUADROS	X
LISTA DE REDUÇÕES	XI
RESUMO	XII
ABSTRACT	XIII
1. INTRODUÇÃO	1
1.1. CONTEXTUALIZAÇÃO.....	1
1.2. DEFINIÇÃO DO PROBLEMA	2
1.3. OBJETIVOS	3
1.4. JUSTIFICATIVA	3
1.5. METODOLOGIA	5
1.6. ESTRUTURA DO TRABALHO.....	7
2. REVISÃO PELOS PARES NA APRENDIZAGEM	8
2.1. A REVISÃO PELOS PARES NA CIÊNCIA.....	8
2.2. ADAPTAÇÃO DA REVISÃO PELOS PARES PARA APLICAÇÃO NA APRENDIZAGEM.....	9
3. ENGENHARIA DE REQUISITOS	16
3.1. ENGENHARIA DE SOFTWARE	16
3.2. DINÂMICA DOS REQUISITOS	19
3.3. REQUISITOS E OS MODELOS DE CICLO DE VIDA.....	20
3.3.1. <i>Modelo em cascata</i>	21
3.3.2. <i>Modelo espiral</i>	22
3.3.3. <i>Modelo iterativo e incremental</i>	23
3.4. MELHORES PRÁTICAS DA ENGENHARIA DE REQUISITOS	25
3.4.1. <i>Analisar o problema</i>	26
3.4.2. <i>Compreender as necessidades dos usuários</i>	28
3.4.3. <i>Definindo o sistema</i>	30
3.4.4. <i>Gerenciamento do escopo</i>	32
3.4.5. <i>Refinamento da definição do sistema</i>	33

3.4.6.	<i>Verificação e validação dos requisitos</i>	34
3.5.	SUMÁRIO DO CAPÍTULO	37
4.	MATERIAIS E MÉTODOS: CUSTOMIZAÇÃO DO PROCESSO DE ANÁLISE DE REQUISITOS	38
4.1.	INTRODUÇÃO	38
4.2.	FASES E ARTEFATOS UTILIZADOS	38
4.3.	FORMATO DOS USE-CASES	40
4.4.	FLUXO DE TRABALHO	41
4.4.1.	<i>Analisar o problema</i>	42
4.4.2.	<i>Compreender as necessidades dos stakeholders</i>	44
4.4.3.	<i>Definir o sistema</i>	45
4.4.4.	<i>Gerenciar o escopo do sistema</i>	45
4.4.5.	<i>Refinar a definição do sistema</i>	46
4.4.6.	<i>Gerenciar as mudanças nos requisitos</i>	46
4.5.	FERRAMENTAS UTILIZADAS	47
5.	REQUISITOS DE SOFTWARE DA REVISÃO PELOS PARES NA APRENDIZAGEM .	48
5.1.	ESCOPO DO SISTEMA A SER DESENVOLVIDO	49
5.2.	ORIGEM DOS REQUISITOS IDENTIFICADOS	49
5.3.	ESTUDO EXPLORATÓRIO	51
5.4.	ACORDO NA DEFINIÇÃO DO PROBLEMA.....	53
5.5.	DEFINIÇÃO DO LIMITE DA SOLUÇÃO.....	54
5.6.	RESTRICÇÕES IMPOSTAS À SOLUÇÃO	55
5.7.	IDENTIFICAÇÃO DOS USUÁRIOS E <i>STAKEHOLDERS</i>	55
5.8.	ATORES	56
5.9.	PRIORIZAÇÃO DOS REQUISITOS	57
5.10.	CARACTERÍSTICAS DE ALTO NÍVEL	59
5.11.	DIAGRAMA DE USE-CASE NO FORMATO RESUMIDO	60
5.12.	DETALHAMENTO DOS USE-CASES	61
5.13.	RASTREAMENTO DOS REQUISITOS.....	64
5.14.	REQUISITOS NÃO FUNCIONAIS.....	66

5.15.	GLOSSÁRIO	67
5.16.	ARTEFATOS COMPLEMENTARES UTILIZADOS	68
5.16.1.	<i>Modelo entidade-relacionamento</i>	68
5.16.2.	<i>Protótipo</i>	71
6.	DISCUSSÃO	73
7.	CONCLUSÃO.....	75
7.1.	SUMÁRIO DA DISSERTAÇÃO	75
7.2.	TRABALHOS FUTUROS.....	76
	REFERÊNCIAS BIBLIOGRÁFICAS	79
	APÊNDICE.....	83
	APÊNDICE A - DOCUMENTO VISÃO.....	83
	APÊNDICE B – USE-CASES	86
	APÊNDICE C - GLOSSÁRIO.....	91

Lista de Figuras

Figura 1: Diagrama de blocos com a abordagem utilizada	6
Figura 2: Exemplo de relatório de revisão de artigo	11
Figura 3: Responsabilidades e fluxo de documentos na revisão pelos pares aplicada à aprendizagem.....	13
Figura 4: Maiores problemas no desenvolvimento de software por categoria.....	19
Figura 5: Custo na reparação de um erro dependendo de quando for descoberto ...	20
Figura 6: Diagrama simplificado do modelo em cascata	21
Figura 7: Modelo espiral.....	22
Figura 8: Distribuição das atividade nas fases do modelo iterativo	24
Figura 9: Pirâmide representando a quantidade de informação.....	30
Figura 10 : Estrutura-exemplo de documento Visão	31
Figura 11: Detalhamento dos use-cases no decorrer das iterações	34
Figura 12: Relação de rastreamento	36
Figura 13: Atividades e artefatos componentes da disciplina de requisitos	42
Figura 14: Fontes utilizadas na elicitação dos requisitos	50
Figura 15 : Diagrama de contexto do sistema	54
Figura 16: Atores do sistema.....	56
Figura 17: Use-case identificados na fase de elaboração.....	61
Figura 18: Diagrama de use-case na primeira iteração da fase de elaboração	62
Figura 19: Diagrama de use-case na segunda iteração da fase de elaboração	63
Figura 20: Estrutura de rastreamento.....	65
Figura 21: Rastreamento dos requisitos.....	66
Figura 22: Modelo ER	69
Figura 23: Modelo ER	70
Figura 24: Protótipo descartável para validação de conceitos	72

Lista de Quadros

Quadro 1: Papéis e atribuições dos participantes	10
Quadro 2: A Indústria de software no Brasil, China e Índia.....	17
Quadro 3: Principais desafios dos projetos de software.....	18
Quadro 4: Número de iterações por fase	39
Quadro 5: Conceituação dos artefatos utilizados no processo de desenvolvimento.	39
Quadro 6: Artefatos criados (C), ou refinados (R) durante as iterações.....	40
Quadro 7: Gabarito do use-case detalhado	41
Quadro 8: Formato padronizado para declaração do problema.....	43
Quadro 9: Formato padronizado para registro de <i>stakeholders</i>	43
Quadro 10: Atributos utilizados no gerenciamento do escopo	45
Quadro 11: Sumário das principais características de softwares de gerência de conferências no que tange a abordagem na aprendizagem	51
Quadro 12: Definição do problema no formato estruturado	53
Quadro 13: Restrições impostas à solução	55
Quadro 14: <i>Stakeholders</i> e usuários do futuro sistema.....	56
Quadro 15: Atores e principal interesse	57
Quadro 16: Algumas combinações de valores dos atributos e a estratégia adotada	58
Quadro 17: Características de alto nível levantadas	59
Quadro 18: Exemplo de use-case no formato detalhado	64
Quadro 19: Principais requisitos não funcionais.....	67
Quadro 20: Amostra de termos registrados no glossário	68

Lista de Reduções

ACM	<i>Association for Computing Machinery</i>
ANSI	<i>American National Standards Institute</i>
CMM	<i>Capability Maturity Model</i>
EAD	Educação a distância
ER	Entidade-relacionamento
ESPITI	<i>European Software Process Improvement Training Initiative</i>
SEI	<i>Software Engineering Institute</i>
SQL	<i>Structured Query Language</i>
UML	<i>Unified Modeling Language</i>
UP	<i>Unified Process</i>
XML	<i>Extensible Markup Language</i>

Resumo

Currículos de graduação na área tecnológica costumam omitir ou tratar superficialmente o desenvolvimento de aspectos não técnicos, porém críticos para a formação profissional, tais como expressão escrita, ética, pensamento crítico e responsabilidade profissional. Esta dissertação identifica e documenta as principais características necessárias a um software que viabilize a utilização da revisão pelos pares na aprendizagem, uma abordagem que evidencia o desenvolvimento de habilidades não técnicas. Os requisitos foram elicitados e documentados utilizando-se diagramas de use-case UML e por meio da customização e aplicação de um processo de desenvolvimento de software iterativo e incremental. Foram aplicadas as melhores práticas da engenharia de requisitos visando evitar o re-trabalho introduzido quando não há um processo eficiente de análise de requisitos. A principal contribuição deste trabalho está no levantamento, documentação e apresentação dos requisitos de software para apoiar a revisão pelos pares na aprendizagem, facilitando futuras implementações. A implementação dos requisitos especificados nesta pesquisa reduzirá a carga burocrática envolvida na aplicação da abordagem, promovendo o desenvolvimento de habilidades não técnicas nos cursos de graduação.

Palavras-chave: revisão pelos pares, aprendizagem, educação de adultos, análise de requisitos, educação a distância, novas metodologias educacionais.

Abstract

Undergraduate curricula of technology majors omit or treat superficially the development of non-technical skills such as written expression, ethics, critical thinking, and professional responsibility, although they are critical for professional education. This dissertation elicits and documents the main software requirements of an environment that supports peer review in learning, an approach that promotes the development of non-technical skills. The requirements were elicited and documented using UML use case diagrams, through the customization and application of an iterative and incremental software development process. The requirements engineering best practices were applied to avoid rework. The main contribution is the identification, documentation, and presentation of the software requirements of an environment for peer review in learning, thus facilitating future implementations. Those implementations will reduce the bureaucratic workload involved in the application of peer review in learning, promoting the development of non-technical skills in the undergraduate curricula of technology majors.

Key words: peer review, learning, lifelong learning, requirements analysis, distance education, new education methodologies.

1. INTRODUÇÃO

1.1. Contextualização

Os profissionais recém formados encontram um mercado de trabalho cada vez mais exigente e seletivo. Além de uma boa formação acadêmica, diversas qualidades pessoais são indispensáveis aos egressos da educação superior. As habilidades requeridas vão além da competência técnica (MORAN, 2001) (CASTELLS e SANTOMÉ, 1999) (MORIN, 1995). Há necessidade do desenvolvimento de habilidades de leitura crítica, prática na expressão escrita e falada e senso de responsabilidade profissional.

Códigos de ética e conduta profissional apontam a habilidade de oferecer e receber crítica como fundamental para profissionais de Computação. O Código de Ética da *Association for Computing Machinery* (ACM), por exemplo, requer dos membros a busca por crítica ao próprio trabalho, e o oferecimento de crítica documentada ao trabalho de terceiros. Segundo Mulder (2000), currículos de graduação em Computação devem incentivar práticas que desenvolvam valores éticos e habilidades na expressão oral e escrita. Em outras áreas, como por exemplo as Ciências da Saúde e Biológicas, a abordagem educacional e discussão sobre a revisão pelos pares são mais comuns do que na área tecnológica (HORROBIN 1974 e 1996). Adversamente a necessidade exploração de tais habilidades os atuais currículos dos cursos de graduação brasileiros não enfocam tais questões. Mas, como auxiliar o aperfeiçoamento destas características em um ambiente de aprendizagem?

A aplicação do processo de revisão pelos pares no meio acadêmico, reconhecido como tendência na formação em Computação (HUGHES, 2003), é uma abordagem pedagógica que auxilia o desenvolvimento das habilidades não-específicas da profissão, tais como a expressão escrita, o trabalho colaborativo, a crítica objetiva e impessoal e a responsabilidade profissional (KERN, 2003). A revisão pelos pares é especialmente promissora como desestabilizadora do papel passivo do aluno em aulas expositivas, de forma que este assuma responsabilidade por sua aprendizagem, conforme relatos de relativo sucesso da aplicação (KERN,

SARAIVA E PACHECO, 2003) (MOREIRA, 2003) e iniciativas correlatas (BUDNY et al., 2002). Neste contexto a abordagem busca o aprimoramento do processo de aprendizagem através da participação ativa dos envolvidos, seguindo o conceito de aprendizagem defendido por Piaget (1974), no qual aprender envolve a assimilação do objeto (informação) a esquemas mentais existentes, construindo-se novas estruturas por meio da modificação deste conhecimento.

Embora exista uma grande demanda por desenvolvimento de software de qualidade, muitas empresas ainda trabalham de forma artesanal, sem adoção ou até mesmo desconhecendo os processos de engenharia de software, trabalhando basicamente com a capacidade criadora da equipe de desenvolvimento. Oesteriech (2002) afirma que a engenharia de sistemas possui o nível mais alto de amadorismo dentre todas as ramificações da engenharia.

Os requisitos compõem a base para as atividades de desenvolvimento e gerenciamento do projeto de desenvolvimento de software (WIEGERS, 2003). Os erros nestas atividades chegam a consumir de 25% a 40% do orçamento total dos projetos de desenvolvimento de software (LEFFINGWELL, 2003). Grande parte dos erros no gerenciamento dos requisitos são relacionados a falta de compreensão das reais necessidades dos usuários ou a omissão da natureza dinâmica dos requisitos (YOUNG, 2004).

O gerenciamento de requisitos exige da equipe de desenvolvimento uma série habilidades e competências visando registrar as necessidades dos usuários de maneira exata e mais flexível possível. Utilizando-se um processo sistemático de descobrimento, organização, documentação e gerenciamento dos requisitos reduzem-se os custos do desenvolvimento, além de atender as reais necessidades dos usuários dentro de um cronograma e orçamento definidos.

1.2. Definição do problema

O desenvolvimento de habilidades não técnicas nos cursos de graduação de base tecnológica pode ser exercitada por meio da abordagem de revisão pelos pares na aprendizagem. Uma ferramenta computacional que suporte a abordagem tem como principal objetivo permitir aos envolvidos a simulação de um fórum científico sem sobrecarregar o organizador do evento com trabalho administrativo.

1.3. Objetivos

Esta dissertação propõe-se a investigar, identificar e documentar quais os requisitos de software para um ambiente que suporte a aplicação da abordagem de revisão pelos pares na aprendizagem presencial ou à distância, minimizando o trabalho burocrático e potencializando a aplicabilidade da abordagem. A investigação deve considerar as tarefas de cadastro, trâmite de documentos, alocação de revisores e apoio à avaliação do trabalho discente. Visando alcançar esse objetivo, serão contemplados os seguintes objetivos específicos:

- estudar as melhores práticas da engenharia de requisitos no contexto de um processo de desenvolvimento de software iterativo e incremental.
- apresentar as peculiaridades da revisão pelos pares na aprendizagem representando as etapas, processos, funções e participantes;
- avaliar ferramentas para gestão de conferências que suportem o processo de revisão pelos pares, identificando suas principais características;
- levantar junto a docentes que têm experiência na organização de congressos ou são interessados na revisão pelos pares na aprendizagem os seus principais requisitos de software;
- desenvolvimento de um protótipo para auxílio na validação requisitos.

1.4. Justificativa

A necessidade de diversas habilidades não técnicas aos egressos dos cursos de graduação, principalmente os das áreas tecnológicas, requer a utilização de abordagens que busquem o desenvolvimento destas características, formando profissionais mais adequados às exigências ao mercado de trabalho. Kern (2002) conjectura que a revisão pelos pares na educação possui valor na mudança de atitude do aprendiz desenvolvendo habilidades de um pesquisador, tornando-o co-responsável na construção coletiva do conhecimento de forma ativa e crítica. Saraiva (2002) apresenta um estudo no qual utiliza a revisão pelos pares na educação como subsidio à educação ambiental demonstrando a efetividade da abordagem na mudança da postura passiva e tradicional dos alunos.

Ao abordar o desenvolvimento de habilidades não-técnicas na aprendizagem de graduação, os resultados desta pesquisa afetam o processo de aprendizagem, que é parte da atividade ou indústria da Educação. Neste sentido, a pesquisa fica caracterizada como de Engenharia de Produção. Mais do que isso, ao apontar direções de pesquisas que envolvem engenharia do conhecimento, como a marcação semântico-retórica de texto, esta dissertação assinala o momento histórico da transição da engenharia de produção para a engenharia de conhecimento.

O excesso de carga burocrática envolvida na aplicação da abordagem demanda a utilização de um grupo de apoio, dificultando a utilização em sala de aula, na qual o professor é responsável por orientar e organizar todo o processo. Neste contexto torna-se indispensável reduzir o excesso de trabalho administrativo atribuído ao professor, para que as questões de cunho pedagógico possam ser mais bem investigadas. Segundo Kern (2003) a disponibilidade de uma interface computacional facilitará a aplicação do processo, reduzindo consideravelmente o trabalho administrativo envolvido.

De acordo com Fowler (2004) o descobrimento e compreensão das reais necessidades dos usuários representa uma das maiores dificuldades dos projetos de software. De 25% a 40% do orçamento total dos projetos de desenvolvimento de software são desperdiçados devido a erros na especificação e gerenciamento dos requisitos (LEFFINGWELL, 2003). A utilização de técnicas de engenharia de requisitos torna-se indispensável na captura, registro e gerenciamento das mudanças nas necessidades dos usuários (ROBERTSON, 1999). Utilizando-se um processo sistemático de descobrimento, organização, documentação e gerenciamento dos requisitos reduzem-se os custos do desenvolvimento, além de atender as reais necessidades dos usuários. Conforme Wieggers (2003) o investimento na compreensão e documentação dos usuários possui alto retorno para o sucesso do projeto de desenvolvimento de software.

Kruchten (2003) define a Linguagem de Modelagem Unificada (UML) como: “linguagem gráfica para visualizar, especificar, construir e documentar os artefatos de um software”. O Processo Unificado (UP) de desenvolvimento é um processo iterativo e incremental que engloba todo ciclo de vida do desenvolvimento de software e utiliza como base de notação a linguagem UML. No UP os requisitos são

usualmente documentados por meio de use-case representando as iterações entre atores e o sistema.

Embora existam algumas ferramentas de gerenciamento de conferência que atendem parcialmente as necessidades da revisão pelos pares na aprendizagem muitas peculiaridades da abordagem aplicada na aprendizagem não são enfocadas. Torna-se indispensável coletar, analisar e registrar os principais requisitos para um software que suporte a aplicação da revisão pelos pares na aprendizagem, de forma que atenda as necessidades específicas de gerenciamento e otimização do fluxo de informações (CORRÊA e KERN, 2003).

Este trabalho pretende utilizar no contexto do processo de desenvolvimento UP as técnicas e melhores práticas da engenharia de requisitos no levantamento e captura das características necessárias a um software que suporte a aplicação da revisão pelos pares na aprendizagem, registrando as principais funcionalidades e características que viabilizarão a aplicação da abordagem com redução significativa da burocracia.

1.5. Metodologia

Este projeto de pesquisa possui natureza qualitativa e exploratória visando levantar e documentar os requisitos de software necessários a uma plataforma de suporte a revisão pelos pares na aprendizagem por meio de revisão bibliográfica, entrevistas com especialistas no domínio da revisão pelos pares e educadores. Visando alcançar os objetivos propostos, sugere-se a abordagem proposta pela Figura 1:

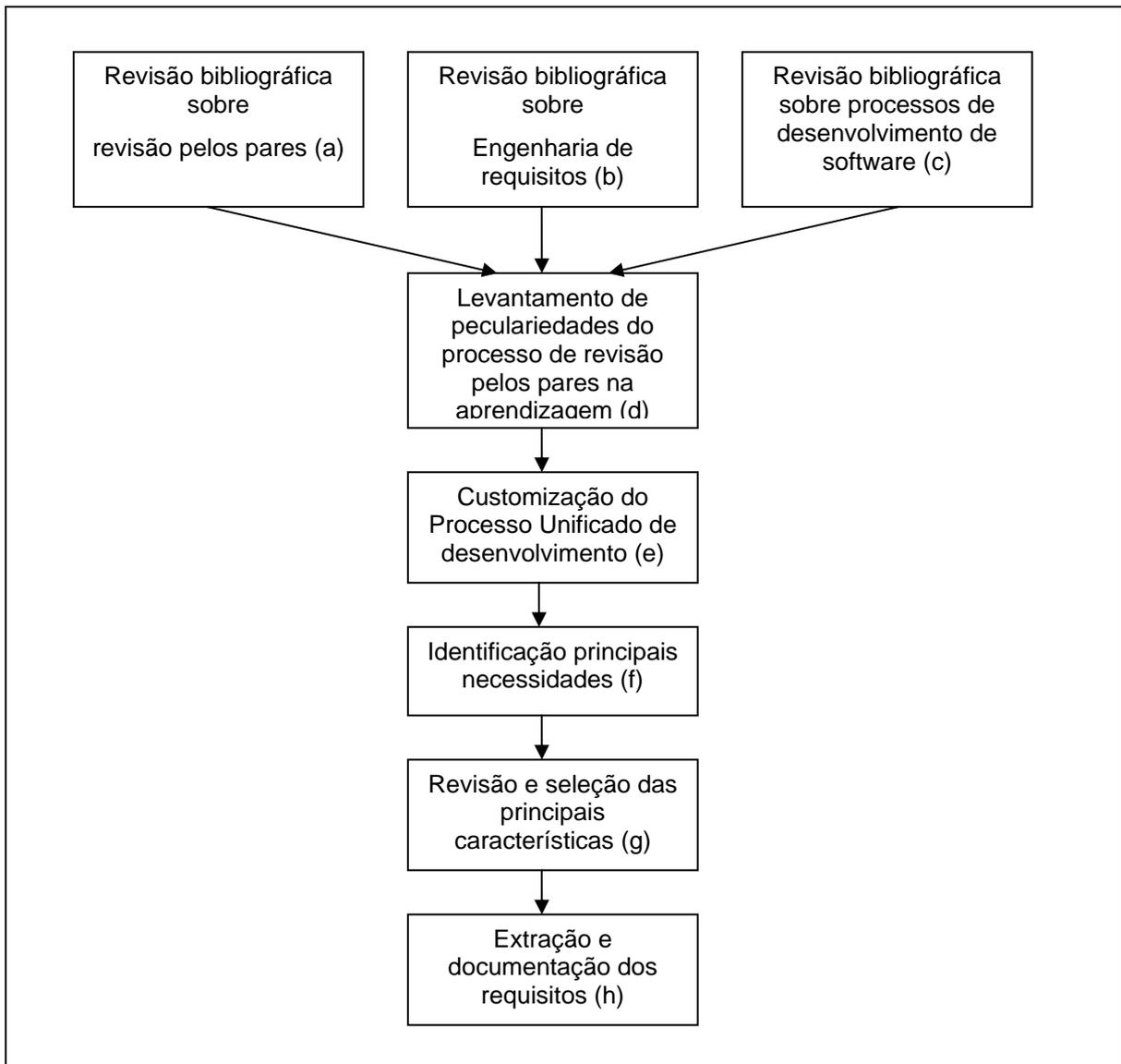


Figura 1: Diagrama de blocos com a abordagem utilizada

a) revisão bibliográfica sobre a utilização da revisão pelos pares na ciência e em sala de aula;

b) revisão bibliográfica sobre os principais métodos, modelos e melhores práticas utilizadas na captura e documentação dos requisitos;

c) revisão bibliográfica sobre os processos de desenvolvimento de software iterativos e incrementais;

d) levantamento de peculiaridades do processo de revisão pelos pares na educação por meio da observação de aplicações práticas;

e) customização do Processo Unificado (UP) de desenvolvimento no que tange a análise de requisitos para adequar as necessidades específicas do projeto em questão;

f) investigação e registro dos principais requisitos requeridos por meio de entrevistas com especialistas em revisão pelos pares;

g) revisão e identificação das principais características das ferramentas de revisão pelos pares mais utilizadas no meio científico;

h) extração e documentação dos principais requisitos identificados e necessários a um sistema computacional que suporte a revisão pelos pares na educação, utilizando as melhores práticas na engenharia de requisitos e os principais artefatos definidos no UP;

1.6. Estrutura do trabalho

Esta dissertação está estruturada em sete capítulos. O capítulo dois apresenta a revisão da literatura, expondo os conceitos da revisão pelos pares quando aplicada à aprendizagem. No capítulo três são apresentados os conceitos de engenharia de software e engenharia de requisitos necessários a um melhor entendimento deste trabalho.

O quarto capítulo descreve em detalhes a customização do processo de engenharia de requisitos para atender as necessidades específicas da aplicação do processo no capítulo cinco. O capítulo cinco apresenta o levantamento de requisitos utilizando o processo customizado na captura dos requisitos para um software que suporte a aplicação da revisão pelos pares na aprendizagem.

Finalizando este trabalho, o capítulo 6 discute o impacto dos resultados e as conclusões e recomendações são apresentadas no capítulo sete. Complementam o texto as referências bibliográficas e o apêndice.

2. REVISÃO PELOS PARES NA APRENDIZAGEM

Neste capítulo são apresentadas definições e técnicas que fundamentam o processo de revisão pelos pares. Apresenta-se um breve histórico sobre o início e a evolução do processo de revisão pelos pares até os dias atuais, bem como sua adaptação para aplicação na aprendizagem. Também são relatadas as iniciativas existentes e a postura do pesquisador atuando como revisor.

2.1. A Revisão Pelos Pares na Ciência

Antigamente os cientistas trocavam cartas entre si para comunicar os resultados de suas pesquisas. Pessanha (1998) salienta que na idade moderna surgiram as primeiras revistas científicas, e a troca de correspondência foi substituída por publicações em revistas especializadas. Ainda segundo Pessanha (1998), a Royal Society of London, em 1753, iniciou a utilização do Conselho da Sociedade como avaliador do material a ser publicado em sua revista, marcando o início oficial da aplicação da revisão pelos pares.

Atualmente o processo de revisão dos artigos científicos submetidos a publicação utiliza pares dos autores como forma de avaliação e validação do conhecimento científico. Mediante o olhar crítico e competente de pesquisadores pares os fóruns científicos subsidiam a decisão da aceitação um artigo para publicação (SMITH, 1990). Segundo Smith (1990) esta decisão deve estar fundamentada se a pesquisa apresenta resultados novos e interessantes, ou uma síntese inovadora e esclarecedora de resultados preexistentes, ou ainda um estudo ou tutorial útil sobre determinado assunto.

Desde seu início oficial até os dias atuais a revisão pelos pares vem sofrendo alterações face a uma série de críticas motivadas principalmente por distorções no uso (HORROBIN, 2001). Rowland (2002) argumenta que a abordagem não detecta a falsificação dos resultados ou plágio de trabalhos, além de sub-reconhecer o trabalho do revisor. Já Rooyen (2001) alega que o processo pode bloquear a originalidade citando o caso da rejeição do primeiro artigo que tratava da vacinação contra a varíola. Pessanha (1998) argumenta que diversos mecanismos e procedimentos foram introduzidos para evitar a má conduta ética dos participantes,

dentre os quais cita: utilizar um número maior de árbitros, empregar questionários de avaliação com critérios que oriente o revisor, permitir ao autor recorrer de uma avaliação que julgue inadequada e fazer uso do processo de forma confidencial.

2.2. Adaptação da revisão pelos pares para aplicação na aprendizagem

Esta seção apresenta a revisão pelos pares na aprendizagem conforme proposta de Kern et al. (2002), onde faz se uso da simulação de um congresso científico em sala de aula envolvendo alunos e professor. Nesta abordagem os artigos submetidos pelos alunos são avaliados por alunos pares que sugerem melhorias. A utilização desta metodologia na aprendizagem busca o aprimoramento dos artigos submetidos por meio da participação crítica dos alunos pares, ao contrário da sistemática aplicada na divulgação da ciência, onde os pares anuem os artigos a serem publicados.

A utilização da sistemática requer do professor dedicação nas tarefas de organização e administração da conferência, bem como desempenhar o papel de presidente do comitê científico. Também deve motivar os participantes, garantindo que o processo transcorra no prazo, com a participação ativa de todos e atingindo os objetivos definidos em cada fase.

Os alunos participam da simulação do congresso exercendo duas funções distintas durante o transcorrer do processo. Em um primeiro momento atuam na função de autores, trabalhando de maneira parecida como um cientista que pesquisa e escreve um artigo segundo as diretrizes do congresso. Após a elaboração de seu trabalho deve submetê-lo a apreciação do comitê científico, que também é formado por alunos. Neste segundo momento os alunos se comportam como revisores, que devem avaliar e sugerir melhorias aos trabalhos submetidos.

Existem variações na forma da aplicação da abordagem. Por exemplo, no processo utilizado por Budny et al. (2002) os revisores são ex-alunos e professores voluntários, enquanto na abordagem utilizada por Kern et. al. (2002) os revisores são alunos pertencentes a mesma sala aula. O processo de análise e desenvolvimento de uma ferramenta que suporte a abordagem deve considerar

estas diferenças, suportando a utilização mesmo com variações na forma de aplicação da revisão pelos pares.

A utilização da abordagem na aprendizagem consiste na simulação de um congresso com a participação de alunos e professores. Cada participante desempenha determinados papéis, e ao final da sistemática são produzidos diversos artigos, resultantes de uma colaboração entre editor, revisores e autores. o Quadro 1 resume os principais papéis e responsabilidade assumida por cada envolvido na sistemática.

Quadro 1: Papéis e atribuições dos participantes

Papel	Desempenhado por	Principal atribuição
Organizador e editor	Professor	Orientar e coordenar atividades, editar caderno técnico de anais
Autor	Aluno	Pesquisar e escrever artigo
Revisor	Aluno	Avaliar artigo

O editor (papel representado pelo professor) inicia o processo publicando o enunciado, convocando os alunos a participarem da simulação do congresso e explicando o funcionamento, papéis e responsabilidades envolvidas. O editor também fica encarregado de orientar e coordenar os trabalhos, publicar a convocatória de artigos, definir um formulário com as avaliações a serem utilizadas na revisão dos artigos. A Figura 2 apresenta um formulário exemplo de um relatório de revisão de artigo.

Nome do revisor: <#preencha...#>

RELATÓRIO DE REVISÃO DE ARTIGO

Número do artigo: <#preencha...#>

Título do artigo: <#preencha com as primeiras palavras...#>

IMPORTANTE - INSTRUÇÕES PARA O REVISOR:

Preencha os campos acima conforme solicitado. Preencha o relatório de revisão abaixo conforme o enunciado do trabalho de nota e sua avaliação. Envie o relatório para o editor (professor) via e-mail, *em formato texto* (.txt) ou no corpo do e-mail, *não* em binário - .doc, .rtf, etc.

SEÇÃO I - VISÃO GERAL:

<# Resuma o artigo em um parágrafo ou sentença. #>

SEÇÃO II - AVALIAÇÃO GERAL:

Atribua graus de 0 (inadequado, discordo totalmente) a 5 (excelente, concordo plenamente) para os aspectos a seguir (3 é o índice mínimo para um trabalho satisfatório):

Autenticidade - o trabalho é dos autores.

Mérito - o artigo relata um tema relevante.

Organização e legibilidade do texto - o texto está organizado de forma que se aprende a partir da leitura e não depois da leitura (não há suspense, pois o Resumo resume, a Introdução introduz...).

Citações/Refs. - fica claro o que é concepção dos autores e o que é conhecimento oriundo da literatura.

Formatação: o artigo respeita a formatação, tanto em aparência quanto em uso dos estilos de parágrafo do template da simulação de conferência.

SEÇÃO III - ASPECTOS ESPECÍFICOS:

Atribua graus de 0 (inadequado, discordo totalmente) a 5 (excelente, concordo plenamente) para os aspectos a seguir (3 é o índice mínimo para um trabalho satisfatório):

O título é adequado -- é sucinto, claro e pode ser considerado o menor resumo do trabalho.

O Resumo *resume* o artigo -- apresenta em poucas palavras a motivação, método, resultados e conclusões.

A Introdução *introduz* o artigo -- aprofunda a motivação e permite ao leitor entender a forma como o conteúdo será apresentado, isto é, permite formar um modelo mental do artigo.

A Conclusão sumaria o artigo e está baseada no conteúdo apresentado (isto é, não tira conclusões "da manga").

A lista de referências mostra *todas* as obras citadas no texto e nenhuma outra.

A linguagem é científica (precisa, rigorosa) e não de senso comum (com juízo de valor, adjetivos e generalizações indevidas ou sem suporte em evidências)

A escolha das seções e a ordem de apresentação faz sentido, não sobram trechos desconectados, não falta conteúdo importante.

O encadeamento de idéias é bom -- é possível compreender o artigo a partir da leitura (sem precisar usar a imaginação); o leitor é guiado através de mudanças de assunto e não precisa tentar adivinhar por que há mudanças de assunto.

Os parágrafos foram bem construídos -- não são excessivamente longos e têm *duas* ou mais sentenças que expressam uma idéia -- ou seja, não há *choppy writing* (redação fragmentada ou esquartejada).

Há constância no estilo -- o autor não alterna aleatoriamente tempos verbais, pessoa do discurso ou formatos.

O uso da língua portuguesa é satisfatório. As sentenças são bem construídas sintática e semanticamente.

SEÇÃO IV - COMENTÁRIOS DETALHADOS:

<# Ofereça aos autores opiniões específicas sobre o que pode ou deve ser mudado para que o artigo seja melhor. Dê opiniões que expliquem os pontos avaliados acima, especialmente se a avaliação é baixa.

É importante separar mudanças necessárias de detalhes. É importante comentar o que o revisor sentiu falta no artigo ou gostaria de saber mais, o que está errado ou incompreensível. É importante referenciar a observação que se faz (indicar onde está o trecho de interesse). #>

Figura 2: Exemplo de relatório de revisão de artigo

Além de gerir todo o trâmite de documentos, o editor deve definir as datas limites para que as etapas do processo ocorram de acordo com o cronograma da disciplina. Outra atribuição muito importante e trabalhosa é a definição dos responsáveis para a revisão dos artigos. Nesta etapa o professor deve obter a melhor relação entre revisor e artigo a ser revisado, não permitindo que um aluno revise artigo do próprio grupo e, evitando a avaliação recíproca.

Neste processo de simulação de revisão pelos pares no meio acadêmico os alunos desempenham dois papéis: autores e revisores. Como autores possuem a responsabilidade de negociação com o professor do tema do artigo a ser pesquisado, de forma que o tema seja desafiante e passível de abordagem dentro do tempo disponível e evitando trabalhos sobre um mesmo assunto e enfoque. Outra atribuição desempenhada por um autor é redação do artigo, que deverá ser entregue até a data limite de submissão, seguindo as diretrizes definidas pelo editor.

Exercendo o papel de revisores devem avaliar artigos dos colegas de maneira crítica e responsável. Com base na avaliação, devem preencher um formulário apresentando suas recomendações aos autores. Estas recomendações deverão ser claras e justificadas. A imparcialidade do processo deve ser buscada. Para isto, podem-se omitir nomes de autores e revisores durante o processo de revisão. Os revisores podem avaliar a apresentação dos artigos como forma de complementação do processo.

A Figura 3 apresenta tarefas, responsabilidades e fluxo de documentos envolvidos na aplicação da sistemática em sala de aula. As tarefas são listadas a seguir.

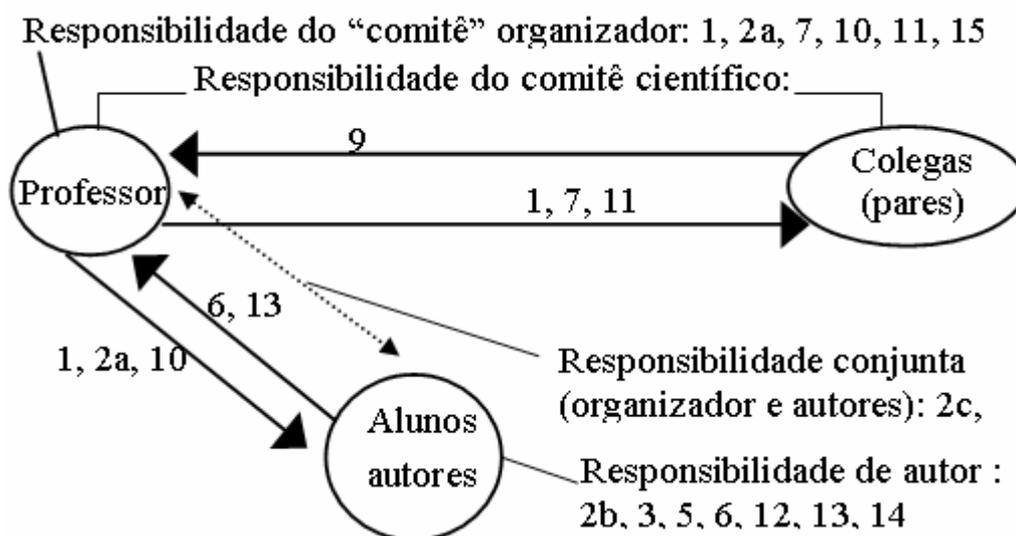


Figura 3: Responsabilidades e fluxo de documentos na revisão pelos pares aplicada à aprendizagem

1. Proposta: Atividade de responsabilidade do professor onde são definidas as regras e diretrizes a serem utilizadas na simulação do congresso em sala de aula. Esta atividade pode ser observada na Figura 3 pelo índice 1, que denota a responsabilidade do professor (no papel de comitê organizador) pela preparação da proposta, e o envio desta proposta aos alunos (no papel de autores).

2. Preparação: Pode ser dividida em três sub atividades: (a) Sugestão de tema: O professor sugere de temas de pesquisa, indica fontes bibliográficas e solicita a formação de grupos de trabalho que devem selecionar tópicos de seu interesse possíveis de serem abordados no artigo. (b) Seleção de tema: Os alunos formam grupos, consultam a literatura e elegem temas candidatos. (c) Aprovação de tema: O professor auxilia os alunos na formação dos grupos e na definição dos temas de preferência, orientando quanto a relevância, propósito, escopo e abordagem do trabalho, de forma que este seja, ao mesmo tempo, interessante, factível e desafiador. Na Figura 3, o professor (comitê organizador) assume a responsabilidade por orientar os alunos (autores) quanto aos temas válidos, inclusive enviando sugestões de temas (item 2a); os alunos (autores) consultam a literatura em busca de temas que os interessem (item 2b); e contam com a orientação do professor na escolha do tema apropriado (item 2c).

3. Esboço: Cada grupo de alunos desenvolve pesquisa e revisão da literatura sobre o tópico escolhido e prepara um esboço do artigo em formato livre. A tarefa de esboço e apreciação do esboço é opcional, sendo que o professor pode utilizá-las caso perceba um baixo envolvimento da turma no processo.

4. Apreciação do esboço: Os alunos de cada grupo discutem o esboço entre si, com o fim de construir uma primeira crítica ao trabalho, e também com o professor, com o fim de receber orientação e acompanhamento.

5. Redação: Atividade onde os grupos completam a redação dos seus artigos conforme as normas do enunciado, buscando orientação do professor quando necessária.

6. Submissão de originais: Cada grupo submete seu artigo original, enviando-o ao professor, que confirma a recepção.

7. Alocação e distribuição dos originais: O professor aloca revisores para os artigos distribuídos, para cada aluno, um ou mais artigos de seus pares, sem a identificação dos autores. Cada artigo é distribuído em número proporcional inteiro ao número de seus autores, o que permite atribuir a cada aluno exatamente a mesma quantidade de artigos para revisar. Não se permite que um aluno revise artigo do próprio grupo e, se possível (se a turma não for muito pequena), evita-se a avaliação recíproca.

8. Revisão pelos pares: Atividade onde os alunos (no papel de revisor) avaliam os artigos, produzindo os relatórios de revisão, baseado no formulário de revisão do processo.

9. Submissão de revisões: Cada aluno envia seu(s) relatório(s) de revisão para o professor, recebendo confirmação da recepção.

10. Distribuição das revisões: O professor edita os relatórios de revisão, elimina comentários dirigidos ao professor e qualquer informação que possa comprometer o anonimato do revisor, e envia a cada grupo de autores os relatórios sobre seu artigo, juntamente com a revisão do professor.

11. Feedback sobre revisões: O professor envia, também, para cada revisor, cópia dos demais relatórios sobre o mesmo artigo, também preservando o anonimato dos revisores.

12. Apropriação do feedback: Com base nas revisões recebidas e em seu próprio julgamento, cada grupo de autores executa as alterações pertinentes no seu artigo.

13. Submissão final: Cada grupo de autores submete ao professor uma versão final e seu artigo, recebendo confirmação.

14. Apresentação: Cada grupo de autores apresenta seu artigo em sessões (aulas) dedicadas especificamente a este fim. O professor estimula o feedback objetivo e impessoal sobre artigo e apresentação.

15. Edição do caderno técnico: O professor conclui sua tarefa de editor compondo um caderno técnico que contém as versões finais dos artigos dos alunos.

A aplicação do processo de revisão pelos pares, conforme a abordagem sugerida por Kern et. al. (2002) envolve extraordinário trabalho administrativo, principalmente as atividades relativas ao trâmite de documentos: recebimento da submissão de originais (6), alocação e distribuição dos originais (7), recebimento da submissão de revisões (9), distribuição das revisões (10), distribuição do feedback sobre revisões (11) e o recebimento da versão final e edição do caderno técnico (13).

Visando facilitar a aplicação do processo em sala de aula, torna-se indispensável a utilização de um software capaz de facilitar a execução das etapas envolvidas no processo, minimizando o trabalho administrativo. Investir tempo na compreensão das necessidades dos usuários é um fator no sucesso de um projeto de desenvolvimento de software (WIEGERS, 2003). Neste contexto, faz-se necessário estudar os conceitos relativos a engenharia de requisitos, que serão apresentados no próximo capítulo.

3. ENGENHARIA DE REQUISITOS

Neste capítulo são apresentados os principais motivos que levam os projetos de desenvolvimento de software ao insucesso, apresentando a engenharia de software como forma para atingir a qualidade e repetitividade nos processos produtivos de software. Mais especificamente, são apresentados os conceitos de engenharia de requisitos, sua influência nos processos de desenvolvimento e as melhores práticas indispensáveis na análise e levantamento dos requisitos.

3.1. Engenharia de software

A disseminação dos computadores pessoais, a popularização da Internet, as agendas eletrônicas, e outros fatores aumentaram grandemente a demanda por sistemas de software. Essa elevação na demanda pressionou os prazos de desenvolvimento e houve uma maior cobrança no cumprimento de prazos. Isto vem causando a profissionalização da indústria de desenvolvimento (PADUAN 2003). Apesar desta demanda iminente, muitas empresas ainda trabalham de forma artesanal, sem métodos, ferramentas, técnicas ou mesmo controle de qualidade. Os profissionais destas empresas são como artistas *high tech*, trabalhando basicamente com criatividade.

A idéia de automatização no processo de desenvolvimento, gerando verdadeiras fábricas de software, faz com que o processo ganhe um ritmo industrial, como uma linha de produção com todos os atributos necessários para aumento de produtividade e garantia de qualidade.

A indústria de software mundial movimenta bilhões de dólares. O Brasil ocupa o sétimo lugar, rivalizando com Índia e China (PADUAN, 2003). Mas ao contrário destes dois países o mercado brasileiro volta-se basicamente para o mercado interno. Um fator que chama a atenção com relação China e Índia é o número de certificações de qualidade, como o *Capability Maturity Model* (CMM) do Software Engineering Institute (SEI). Enquanto que a Índia, um grande pólo exportador de tecnologia, possui setenta e uma empresas em estágios avançados da certificação CMM o Brasil possui apenas treze empresas. O Quadro 2 apresenta dados relativos a indústria de software no Brasil comparado a seus concorrentes mais próximos

(China e Índia). Neste quadro percebe-se que o Brasil exporta menos que seus concorrentes diretos. Segundo Paduan (2003) existe relação deste fato com a falta de profissionalização da indústria de software brasileira.

Quadro 2: A Indústria de software no Brasil, China e Índia

		(US\$ bilhões)	Brasil	China	Índia
Ano 2000	Mercado doméstico (produtos e serviços)		7,2	7,0	1,8
		Produtos	3,2	3,0	-
		Serviços	4,0	4,0	-
	Exportações (produtos e serviços)		0,1	0,4	4,0
	Total software (mercado doméstico + exportações)		7,3	7,4	5,8
	Pessoas em software		158.353	186.000	350.000
	Empresas em software		10.713	10.000	>2.800
	Empresas em desenvolvimento de software		2.398	5.700	-
	Graduados anualmente na área de TI		23.109	50.000	73.000
Ano 2001	Mercado doméstico (produtos e serviços)		7,7	7,9	2,0
		Produtos	3,6	3,6	-
		Serviços	4,1	4,3	0,4
	Exportações (produtos e serviços)		-	-	6,2
	Total software (mercado doméstico + exportações)		7,7	7,9	8,2

Fonte: Adaptado de Paduan (2003)

Conforme Kotonya (1998) a engenharia de software é responsável pelo estudo das disciplinas envolvidas na especificação, desenvolvimento, gerenciamento e evolução de sistemas de softwares. Sommerville (1995) define o processo de software como um conjunto de atividades e resultados associados que produzem um produto de software. A engenharia de software tem por objetivo estudar as teorias, atividades, métodos, técnicas e ferramentas envolvidas no desenvolvimento de sistemas de software, visando tornar os processos de desenvolvimento mais produtivos e com maior qualidade.

Apesar de uma evolução extraordinária nas últimas três décadas, nota-se que a engenharia de software ainda está longe da maturidade necessária (OESTEREICH, 2002). Outro fato importante a se citar é a grande morosidade na aplicação de novas práticas por parte do mercado, fazendo com que haja uma grande diferença entre pesquisa-mercado, conforme Armour (2001). O grande desafio deste início de século está na transferência dos avanços nas pesquisas de engenharia de software para o dia-a-dia das empresas de desenvolvimento.

Apesar das novas e efetivas técnicas de engenharia de software, muitos projetos de desenvolvimento de sistemas de software tendem a falhar, especialmente devido as falhas na definição dos requisitos (KOTONYA, 1998). Conforme Leffingwell (2003) um estudo conduzido pelo The Standish Group identificou em 1994 os principais desafios dos projetos de desenvolvimento de software, conforme apresentado Quadro 1.

Quadro 3: Principais desafios dos projetos de software

Ordem	Fatores de desafio dos projetos	% de respostas
1	Falta de contribuição do usuário	13%
2	Especificação dos requisitos incompleta	12%
3	Mudança nas especificações dos requisitos	12%

Fonte: Adaptado de Leffingwell (2003)

Por este estudo percebe-se que trinta e sete por cento dos projetos de desenvolvimento indicam que os maiores desafios dos projetos de software estão relacionados a coleta e gerenciamento dos requisitos. A Figura 4 apresenta um estudo realizado pelo *European Software Process Improvement Training Initiative* (ESPITI) realizado em 1995 visando identificar a importância dos vários problemas na indústria de software, onde se percebe que os principais problemas são relacionados a engenharia de requisitos.

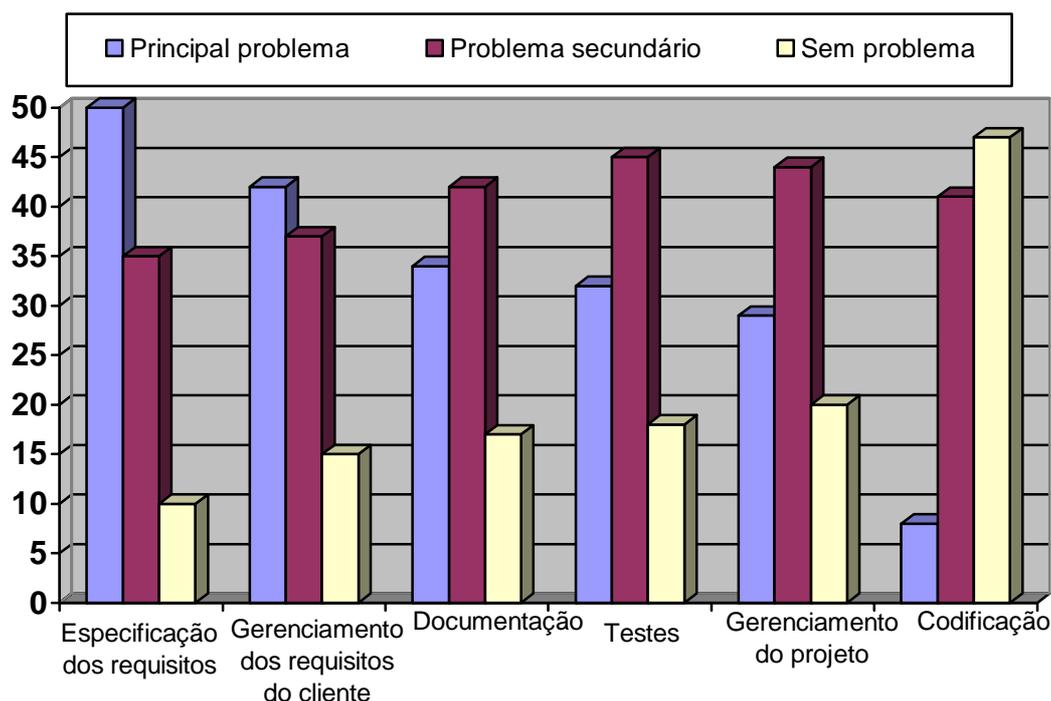


Figura 4: Maiores problemas no desenvolvimento de software por categoria

Fonte: Adaptado de Leffingwell (2003)

3.2. Dinâmica dos requisitos

Os *stakeholders* representam uma pessoa ou grupo de pessoas que serão influenciadas com o resultado do desenvolvimento de um software (Kruchten, 2003). Os requisitos são dinâmicos por natureza, ajustando-se com o passar tempo para acomodar as necessidades dos *stakeholders* (YOUNG, 2004). A equipe de desenvolvimento deve estar preparada para gerenciar mudanças, ao invés de coibi-las (LEFFINGWELL, 2003). Pressman (1997) afirma que a modificação nos requisitos de software existem por diversas razões, algumas são fatores internos e perfeitamente controláveis, outras são fatores externos que estão fora do controle de usuários e da equipe de desenvolvimento.

Existem vários fatores externos que geram mudança no desenvolvimento de software, dentre os quais Leffingwell (2003) cita: mudanças na economia, leis governamentais, modificação da necessidade do mercado consumidor, entre outros. Os fatores internos estão relacionados a engenharia dos requisitos e pode-se

destacar a falta do envolvimento dos usuários, ou a não compreensão das reais necessidades como os principais motivos que impulsionam as mudanças.

A mudança é uma constante no processo de desenvolvimento, e deve ser gerenciada da forma mais eficiente visando realizá-las nas fases iniciais do projeto. Nestas fases o cliente pode rever as exigências e recomendar modificações sem causar grande impacto sobre os custos. Porém, quando são exigidas mudanças durante as fases mais avançadas do projeto de software o impacto sobre os custos eleva-se rapidamente. A Figura 5 mostra o custo relativo na reparação de um erro dependendo de qual estágio do desenvolvimento é descoberto, ou seja, a correção de um erro com o sistema em operação custa mais de cem vezes do que se fosse descoberto durante a análise de requisitos.

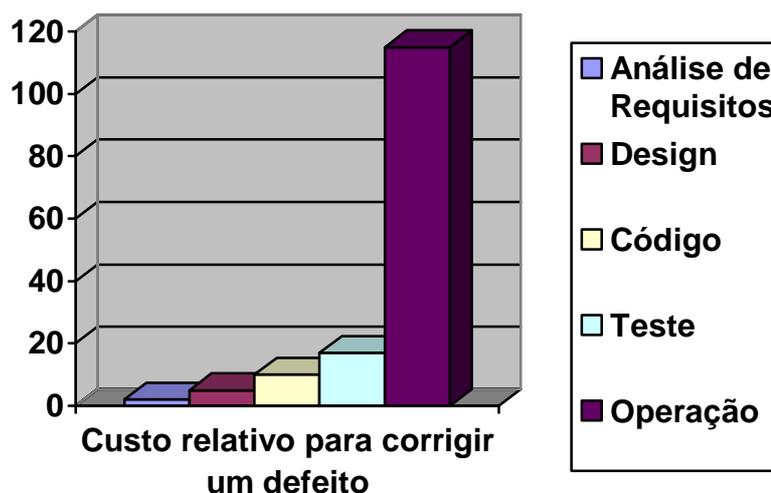


Figura 5: Custo na reparação de um erro dependendo de quando for descoberto

Fonte: Adaptado de Wieggers (2003)

3.3. Requisitos e os modelos de ciclo de vida

Os ciclos de vida regem as atividades envolvidas na construção de software, são responsáveis por definir: atividades, artefatos, seqüências, responsáveis, entre outros. Para que as atividades de engenharia de requisitos aconteçam é imperativo

um processo de desenvolvimento de software razoavelmente definido (WIEGERS, 2003).

Não existe um único ciclo de vida aceito universalmente ou correto, assim como não existe uma única linguagem de desenvolvimento padrão. A organização onde o processo será aplicado; o tipo de produto a ser construído; o alinhamento com os objetivos estratégicos da organização; dentre outros fatores; faz com cada tipo de projeto possua um processo de desenvolvimento mais adequado. A evolução e adaptação contínua de um processo de software a realidade da empresa é condição básica para alcançar uma maior maturidade do processo de desenvolvimento.

3.3.1. Modelo em cascata

Também conhecido como modelo clássico, é um modelo linear, onde cada fase transcorre completamente antes do início da próxima. As saídas de uma atividade são utilizadas como entrada na próxima atividade e o software só entregue ao final de todas as atividades. Existem algumas variações deste modelo, mas conforme Kotonya (1998) as principais fases de modelo são: definição dos requisitos, definição do sistema de software, implementação, teste de unidade, teste de integração e manutenção, apresentados na Figura 6

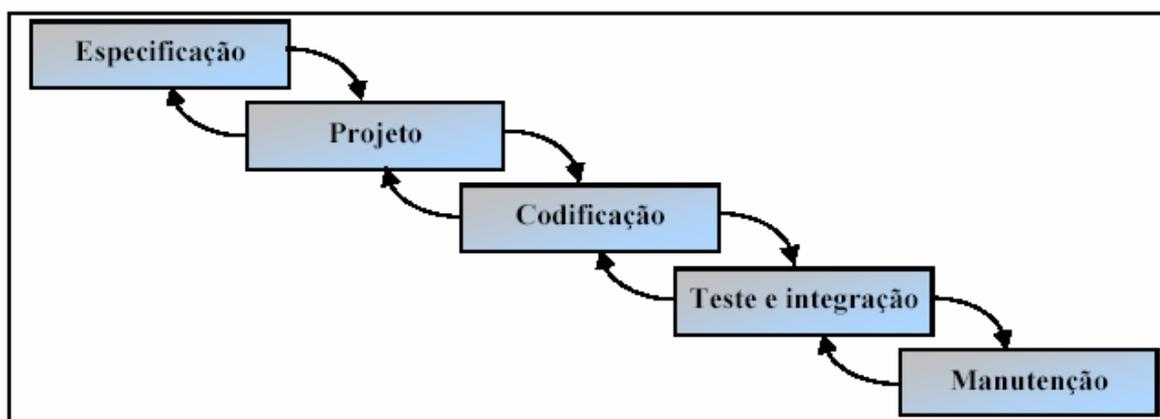


Figura 6: Diagrama simplificado do modelo em cascata

Young (2004) cita que o modelo em cascata tem como principal vantagem a necessidade do levantamento dos requisitos como o primeiro passo no desenvolvimento de software, contudo para se passar para a fase seguinte o modelo enfatiza uma elaboração quase que total dos requisitos. Isto representa uma abordagem fixa e rígida, onde os requisitos são “congelados” por toda a vida do projeto, muitas vezes causando a entrega de um produto que não atende as reais necessidades dos usuários.

3.3.2. Modelo espiral

Conforme Pressman (1997) o modelo espiral é um modelo “evolucionário” baseado em uma seqüência de fases que culminam em versões incrementais do software. Possui a representação de uma espiral dividida em quatro quadrantes (conforme apresentado na Figura 7) representando as quatro atividades básicas: planejamento, análise de riscos, engenharia, avaliação do cliente (YOUNG, 2004).

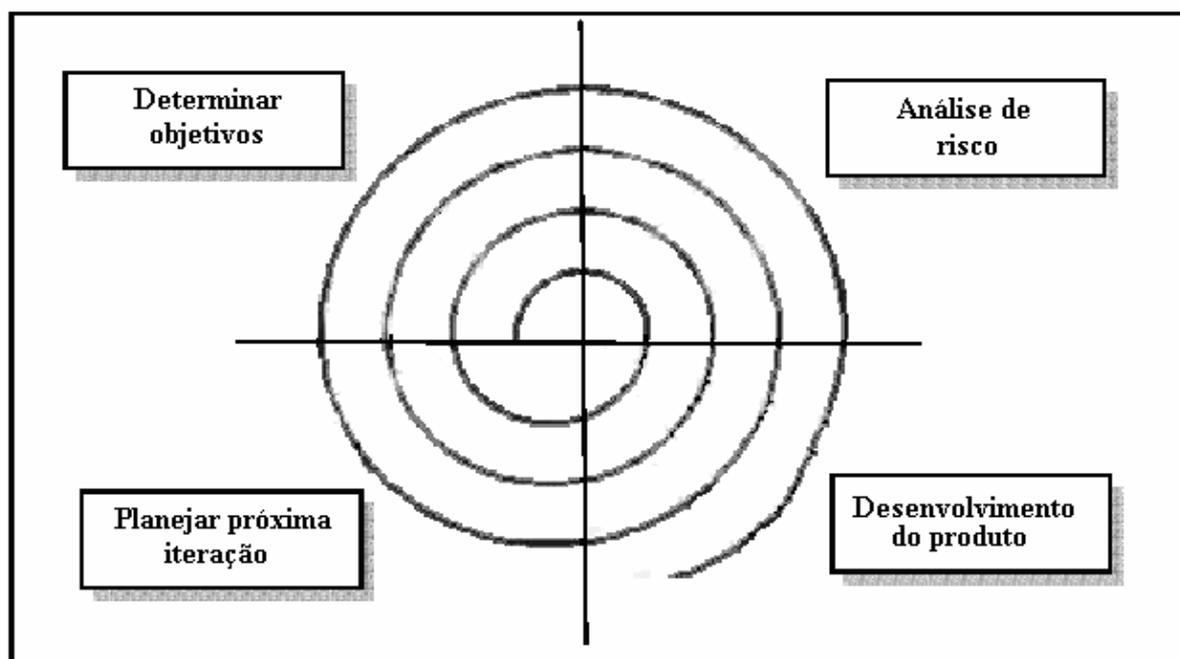


Figura 7: Modelo espiral

Fonte: Adaptado de Young (2004)

O modelo espiral inicia com o planejamento dos requisitos e o conceito de validação, seguido por um ou mais protótipos (YOUNG, 2004). Os protótipos no

começo do processo tornam possível realimentação dos *stakeholders*, melhorando o entendimento dos requisitos e reduzindo os riscos. Em contrapartida o desenvolvimento de vários protótipos pode ser um grande consumidor de tempo.

3.3.3. Modelo iterativo e incremental

O modelo iterativo e incremental toma como premissa básica a impossibilidade de conhecer todos os requisitos no início do projeto. Para tanto, prega a necessidade de um desenvolvimento que melhore pouco a pouco, ou seja, refinando o sistema a cada iteração (KROLL 2003). As atividades envolvidas no modelo iterativo e incremental podem ser divididas em quatro fases: concepção, elaboração, construção e transição (KRUCHTEN, 2003). A Figura 8 apresenta as fases e iterações definidas no UP bem como o esforço necessário em cada disciplina. A quantidade de esforço em cada disciplina varia dependendo conforme a fase do projeto de desenvolvimento, por exemplo, na fase de concepção a equipe de desenvolvimento empreende um maior esforço nas disciplinas de modelagem de negócios e análise de requisitos.

Kroll (2003) informa que cada fase é composta por uma ou mais iterações, representando a evolução natural do projeto durante o decorrer do tempo. Durante cada iteração a equipe envolve-se tempo apropriado em cada disciplina, não sendo necessário a execução completa de todas as disciplinas para término de uma fase, como pode ser visto na Figura 8.

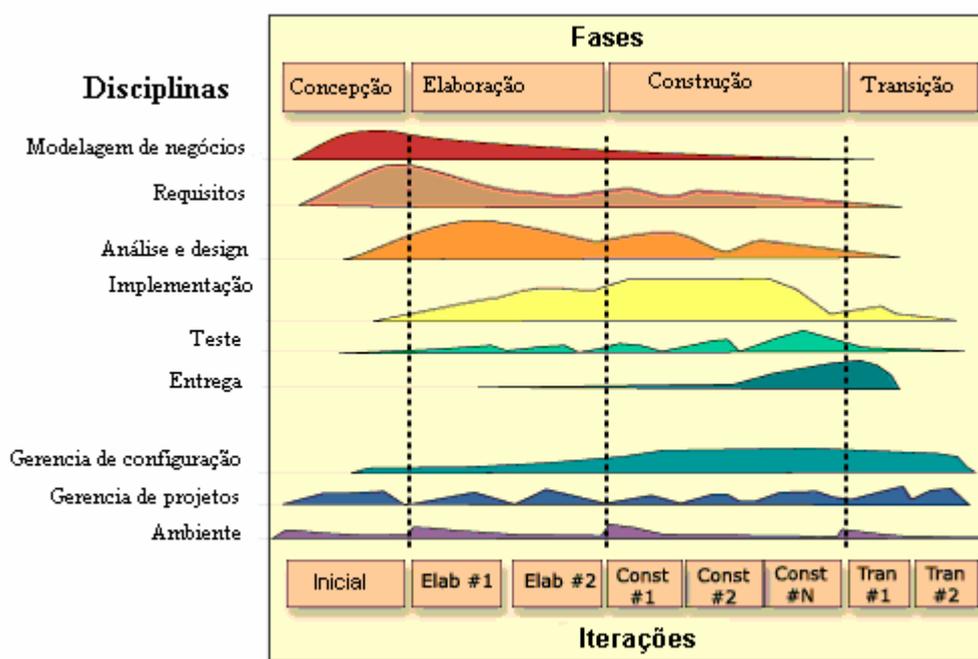


Figura 8: Distribuição das atividade nas fases do modelo iterativo

Fonte: Adaptado de Kruchten (2003)

Fases do modelo iterativo e incremental com o principal objetivo a ser alcançado (KROLL 2003):

- **Concepção:** focada na clara definição e entendimento do problema, bem como seu escopo. As primeiras estimativas de tempo são realizadas nesta fase, bem como a análise de negócios e o entendimento inicial das necessidades dos usuários. O principal artefato gerado nesta fase é um documento de alto nível contendo uma visão geral do sistema a ser desenvolvido, normalmente chamado de "Documento Visão".
- **Elaboração:** acontece o detalhamento dos requisitos para o sistema, e uma definição inicial da arquitetura é estabelecido. O objetivo principal desta fase é preparar os artefatos necessários para que a codificação ocorra de maneira organizada.
- **Construção:** fase focada na implementação, mas possui atividades relacionadas com análise e projeto.

- Transição: fase onde são realizados e acontece a implantação e estabilização da aplicação. Os treinamentos aos usuários da aplicação serão realizados nesta fase.

Segundo Leffingwell (2003) o modelo de desenvolvimento iterativo e incremental tem duas vantagens significativas para o gerenciamento eficiente dos requisitos, dentre as quais pode-se citar:

1. Redução dos riscos nas fases iniciais do projeto: por sua natureza os sistemas de software são intangíveis, abstratos e, na teoria, infinitamente modificáveis. A geração de uma versão executável a cada iteração materializa uma idéia, permitindo ao usuário validar se as características desenvolvidas realmente correspondem a sua necessidade. Com o passar das iterações o número de modificações é reduzido gradativamente.

2. Melhor gerenciamento do escopo: supondo-se que o um projeto tenha sido mal estimado e esteja atrasado na primeira iteração. Com existirem múltiplas versões, uma cada iteração, o gerente do projeto pode tomar as medidas necessárias para administrar este imprevisto.

3.4. Melhores práticas da engenharia de requisitos

Sommerville (1995) define que requisitos são especificações de serviços ou restrições que o sistema deve prover. Ou, conforme o Robertson e Robertson (1999), requisitos são características e propriedades que um produto deve apresentar para a resolução de um problema específico do mundo real. Os requisitos são as definições que delineiam as funções e características que um produto deverá possuir visando atingir um objetivo determinado, depois de construído, sem referir-se ao modo utilizado na resolução.

Diversas atividades e habilidades estão envolvidas no processo de obtenção, documentação e validação dos requisitos. O conjunto de técnicas empregadas para levantar, detalhar, documentar e validar os requisitos de um produto forma a engenharia de requisitos. Sommerville (1995) define a engenharia de requisitos como um processo sistemático para elicitacão, entendimento, análise e documentação dos requisitos. Ainda segundo Sommerville (1995), o fato de se

chamar a isto um processo de “engenharia” é estranho, mas o termo é utilizado para indicar que isto é um processo sistemático e prático.

A disciplina de engenharia de requisitos é muito ampla. Pode-se abordar diferentes aspectos: da compreensão do fator humano na captura dos requisitos até modelos matemáticos para a representação formal das necessidades dos *stakeholders*. Esta dissertação fundamenta-se na visão apresentada por Leffingwell (2003) onde os autores apresentam uma série de práticas que devem ser exploradas pela equipe de desenvolvimento visando analisar os requisitos de forma eficiente. Outros autores constataam as melhores práticas definidas por Leffingwell (2003) como imprescindíveis no processo de análise de requisitos, embora utilizem outra forma de apresentação do conteúdo, dentre eles podemos citar: Wiegers e Kulak (2003).

Leffingwell (2003) define seis práticas que devem ser aplicadas a qualquer processo de análise de requisitos listadas abaixo e detalhadas nas seções seguintes.

- Analisar o problema
- Compreender as necessidades dos usuários
- Definindo o sistema
- Gerenciamento do escopo
- Refinamento da definição do sistema
- Verificação e validação dos requisitos

3.4.1. Analisar o problema

Muitos projetos de software não conseguem satisfazer as reais necessidades dos usuários, muito embora a tecnologia e as ferramentas de suporte a análise de sistemas tenham evoluído muito nos últimos tempos (WIEGERS, 2003). Uma das principais causas nesta diferença de semântica entre a real necessidade dos usuários e o produto entregue está no entendimento inadequado do problema a ser resolvido. Conforme Young (2004) é necessário que a equipe de desenvolvimento estude o problema a ser resolvido através de múltiplos pontos de vista, tentando identificar as reais causas do problema.

Kotonya (1998) define a análise do problema como: “A atividade de análise de problema envolve o aprendizado sobre o problema a ser desenvolvido (normalmente *brainstorming* e ou questionamento), entendimento das necessidades dos usuários potenciais, e tentativas de descobrir quem o usuário realmente é, e entendimento de todas as restrições da solução.” O objetivo desta habilidade é prover linhas gerais para análise de problema, é conforme Kotonya (1998) existem quatro dimensões a serem exploradas:

- Compreensão sobre do domínio da aplicação: adquirir conhecimento geral sobre o assunto a ser desenvolvido.
- Entendimento do problema: especializar o conhecimento do domínio da aplicação com detalhes particulares há realidade específica.
- Entendimento do negócio: explorar a integração do futuro sistema com o restante do negócio.
- Entendimento das necessidades e restrições dos *stakeholders* do sistema: definição de um modelo de negócios que represente o trabalho realizado pelos *stakeholders*.

Leffingwell (2003) apresenta cinco atividades a serem utilizadas objetivando a compreensão exata do problema a ser revolido pela equipe de desenvolvimento e cliente. São elas:

- Obter um acordo sobre a definição do problema: documentar o problema por meio de uma sentença textual e ganhar o consentimento dos envolvidos. Pode ser útil escrever esta sentença de forma padronizada identificando o problema, quem esta sendo afetado por este problema, quais os efeitos que este problema gera e listar alguns poucos benefícios chaves a serem atingidos com a nova solução.
- Entender as raízes dos problemas: Deve-se realizar uma investigação detalhada com *stakeholders* que possuam conhecimento no domínio da aplicação para identificar as principais causas dos problemas a serem resolvidos visando identificar os “problemas por trás dos problemas”.

- Identificar *stakeholders* e usuários: diferentes grupos de usuários estão envolvidos no desenvolvimento do sistema, cada qual com necessidades e objetivos diferentes. A definição dos grupos de e suas necessidades particulares é um ponto chave para o sucesso de uma solução. Existem diversos *stakeholders* que não são usuários do sistema, mas são afetados pelo desenvolvimento de uma nova solução, portanto devem ser identificados e documentados como *stakeholders*.
- Definir as fronteiras para a solução proposta: identifica até onde a solução proposta pretende abranger e sua interação com os elementos restantes da solução. Algo ou alguma coisa fora das fronteiras do sistema que interaja com o sistema é considerado um ator.
- Identificar restrições na solução: orçamento, linguagem de desenvolvimento, banco de dados e outras restrições devem ser definidos no início do projeto. Estas restrições irão limitar o grau de liberdade que poderá ser adotado na solução a ser desenvolvida.

3.4.2. Compreender as necessidades dos usuários

O estudo realizado em 1994 pelo *The Standish Group* demonstra que no mundo real a falta de participação do usuário é um dos principais problemas nos projetos de software. Visando maximizar a participação do usuário no processo de desenvolvimento devem-se utilizar as mais diversas técnicas para que o analista consiga eliciar as principais necessidades dos usuários.

Leffingwell (2003) utiliza três metáforas para explicar a dificuldade na compreensão das necessidades dos usuários:

- A síndrome do “Sim, mas”: a imaterialidade do processo de desenvolvimento de software faz com que o usuário apresente uma reação de surpresa com o produto entregue e logo em seguida inicia diversas sugestões de melhorias e modificações. Esta reação é inerente a natureza humana, que deve ser minimizada utilizando-se técnicas de prototipação e o desenvolvimento iterativo incremental visando obter estas reações o mais cedo possível.

- A síndrome das “Ruínas não descobertas”: por mais minucioso que seja o procura por requisitos sempre restarão requisitos a serem descobertos. Wiegers (2003) afirma que encontrar todos os requisitos é uma tarefa impossível, mas em determinado momento a equipe de desenvolvimento deve estar convicta de que um conjunto determinado de requisitos é suficiente para guiar o processo de desenvolvimento..
- A síndrome do “Usuário e o desenvolvedor”: tipicamente usuários e a equipe de desenvolvimento possuem muitas diferenças: objetivos, background, motivação, linguajar. Este grande distância entre estes “dois mundos” dificulta a comunicação, causando erros na transmissão, recepção e interpretação das necessidades dos usuários.

Visando facilitar à compreensão das necessidades dos usuários e minimizar o aparecimento das síndromes descritas a equipe de desenvolvimento deve utilizar as técnicas que melhor auxiliem na compreensão dos requisitos de usuários e *stakeholders* (LEFFINGWELL, 2003). Segundo Leffingwell (2003) as principais técnicas que auxiliam na captura efetiva das necessidades dos usuários são:

- Entrevista e questionário: consiste em aplicar questionários ou entrevistar os principais *stakeholders*.
- Workshop de requisitos: consiste de realizar workshops com os *stakeholders* chaves visando capturar os principais requisitos. Segundo Leffingwell (2003) talvez seja uma das técnicas mais eficientes no elicitación de requisitos.
- *Brainstorming* e redução de idéias: técnica a ser utilizada quando é necessário encontrar novas idéias ou soluções criativas para problemas específicos.
- Quadro de histórias: utiliza personagens e histórias para explicar o comportamento do sistema. Objetiva obter a reação do usuário com os conceitos propostos para a aplicação.
- Use-cases: são muito parecidos com quadro de histórias e apresentam o comportamento do sistema e seus atores, capturando um

subconjunto muito importante dos requisitos: a interação entre usuário e o sistema e como o sistema responde a estes estímulos.

- Prototipação: o protótipo garante a interação do usuário com o sistema em seu ambiente, antecipado a síndrome do “Sim, mas”.

3.4.3. Definindo o sistema

Após analisar o problema e compreender as necessidades dos usuários deve-se encontrar e documentar as características que o sistema deverá possuir para atender as necessidades dos usuários (WIEGERS, 2003). Esta transição representa a passagem do espaço do problema (necessidades dos usuários) para o espaço da solução (características do sistema), onde a equipe de desenvolvimento deve estar focada em definir as soluções para atender as necessidades dos usuários (KULAK, 2003).

A quantidade de informação a ser gerenciada nesta etapa tende a aumentar, pois uma simples necessidade pode demandar diversas características e comportamento a serem cumpridos pelo sistema. Segundo Leffingwell (2003), a cada nível que se desce em direção a parte inferior da pirâmide (Figura 9), o nível de detalhamento aumenta. Na definição das características a equipe de desenvolvimento deve permanecer em um alto nível de abstração e deixar que os requisitos de software sejam bem detalhados em use-cases ou texto corrido.

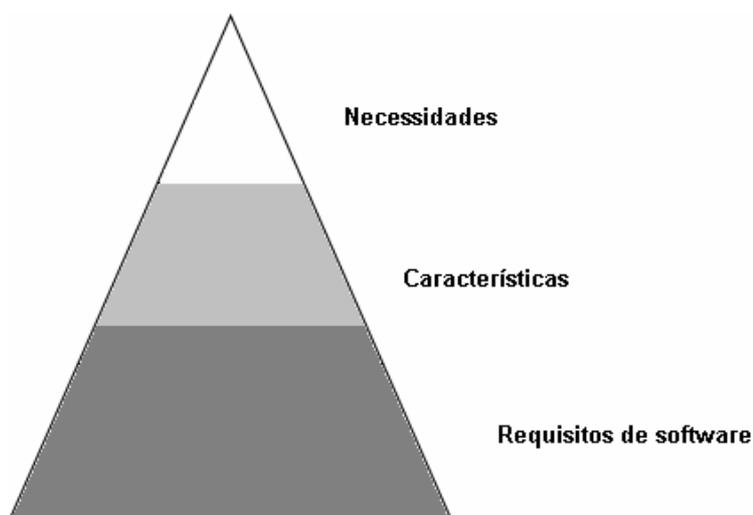


Figura 9: Pirâmide representando a quantidade de informação

Segundo Wieggers (2003), o documento visão é um dos mais importantes artefatos em um projeto de software, seu principal objetivo é descrever usuários e suas necessidades, características em alto nível de abstração, além de outros dados necessários ao projeto. O documento visão deve ser desenvolvido nas fases iniciais do projeto, representando uma visão geral das informações mais importantes do produto a ser desenvolvido. Como o documento visão deve ser desenvolvido em texto plano, serve como base para discussão e acordos com todos os envolvidos no projeto.

Diferentes tipos de projetos requerem diferentes técnicas para organizar os requisitos, mas o documento visão é um artefato que pode ser utilizado em quase todo projeto de software e deve ser personalizado a para refletir necessidades específicas. A Figura 10 apresenta um exemplo de estrutura de documento visão.

Índice

1. Introdução	1
1.1 Propósito	1
1.2 Visão geral	1
2. Posicionamento	1
2.1 Oportunidade de Negócio	1
2.2 Afirmação do Problema	1
2.3 Posicionamento do Produto	2
3. Stakeholder e Descrição dos Usuários	2
3.1 Ambiente do Usuário	2
4. Visão Geral do Produto	2
4.1 Resumo das Capacidades	3
5. Características de Alto Nível do Produto	3

Figura 10 : Estrutura-exemplo de documento Visão

Fonte: Adaptado de Wieggers (2003)

Deve haver uma pessoa responsável pelo documento visão que o mantenha atualizado durante todo o ciclo de desenvolvimento do software, desta forma o documento sempre representará as características que o futuro sistema irá possuir.

3.4.4. Gerenciamento do escopo

Leffingwell (2003) alerta que muitos projetos de software têm seu escopo inicial com duas vezes mais características possíveis de se implementar no tempo estimado inicialmente. A equipe de desenvolvimento deve estar capacitada a administrar três variáveis envolvidas no gerenciamento do escopo:

- Conjunto de funcionalidades a ser entregue para o usuário;
- Recursos a serem utilizados;
- Tempo necessário para implementação;

O primeiro passo na definição do escopo é a criação de uma lista contendo as características que se pretende entregar em uma versão específica. Leffingweel (2003) sugere que esta lista deve conter de 25 a 99 características, não importando o tamanho do projeto. Já Larman (2003) recomenda que o documento visão não possua mais do que 50 características. Escopos que superem o limite superior tendem a possuir características muito detalhadas, dificultando a comunicação efetiva com os clientes e a equipe de desenvolvimento. Com menos características, o nível de detalhe pode ser muito baixo para prover suficiente compreensão da aplicação e o nível associado de esforço para implementação.

Com base na lista de características, os usuários e *stakeholders* devem realizar a priorização das características. Esta priorização deve ser realizada sem muita influência da parte técnica, podendo-se utilizar valores como: indispensável, importante, útil.

Como próximo passo os desenvolvedores devem estimar o esforço aproximado para implementar cada característica, definindo sua ordem de grandeza. Segundo Leffingweel (2003) a equipe de desenvolvimento reluta em fornecer esta primeira estimativa sem antes realizar uma definição completa dos requisitos. Contudo é importante realizar esta atividade para se ter uma estimativa inicial do projeto antes mesmo que se gaste mais tempo na definição dos requisitos, as vezes definindo até requisitos que nem mesmo serão implementados.

Cada característica possui um risco associado que deve ser definido pela equipe de desenvolvimento. As características de risco alto possuem uma maior probabilidade de causar problemas na sua implementação e trazer prejuízos ao

projeto. Características que possuam prioridade indispensável e risco alto devem possuir estratégias para mitigar este risco.

3.4.5. Refinamento da definição do sistema

As características do sistema definidas no documento visão devem ser transformadas em requisitos de software em nível de detalhes suficiente para guiar um desenvolvimento eficiente, garantindo que a equipe de desenvolvimento possa implementar todas as funcionalidades que suportem as necessidades dos usuários.

Segundo Fowler (2004) o texto de um use-case descreve as ações do usuário, neste contexto chamado de ator, para completar um objetivo no sistema. A UML não especifica formato padrão na descrição textual do use-case, mas Larman (2003) sugere a utilização de três formatos conforme o nível de detalhe requerido durante o processo de desenvolvimento do software. Cada formato possui um nível de detalhe diferente. O formato resumido descreve através de um parágrafo qual o principal objetivo do use-case, sem maiores detalhes. O formato casual apresenta as operações um pouco mais detalhadas e o formato detalhado apresenta o maior nível de detalhe, conforme apresentado na Figura 11. No início do processo de desenvolvimento a equipe registra todos os use-cases no formato resumido, que são reescritos para os outros formatos conforme o andamento do projeto e a prioridade do use-case no contexto do desenvolvimento.

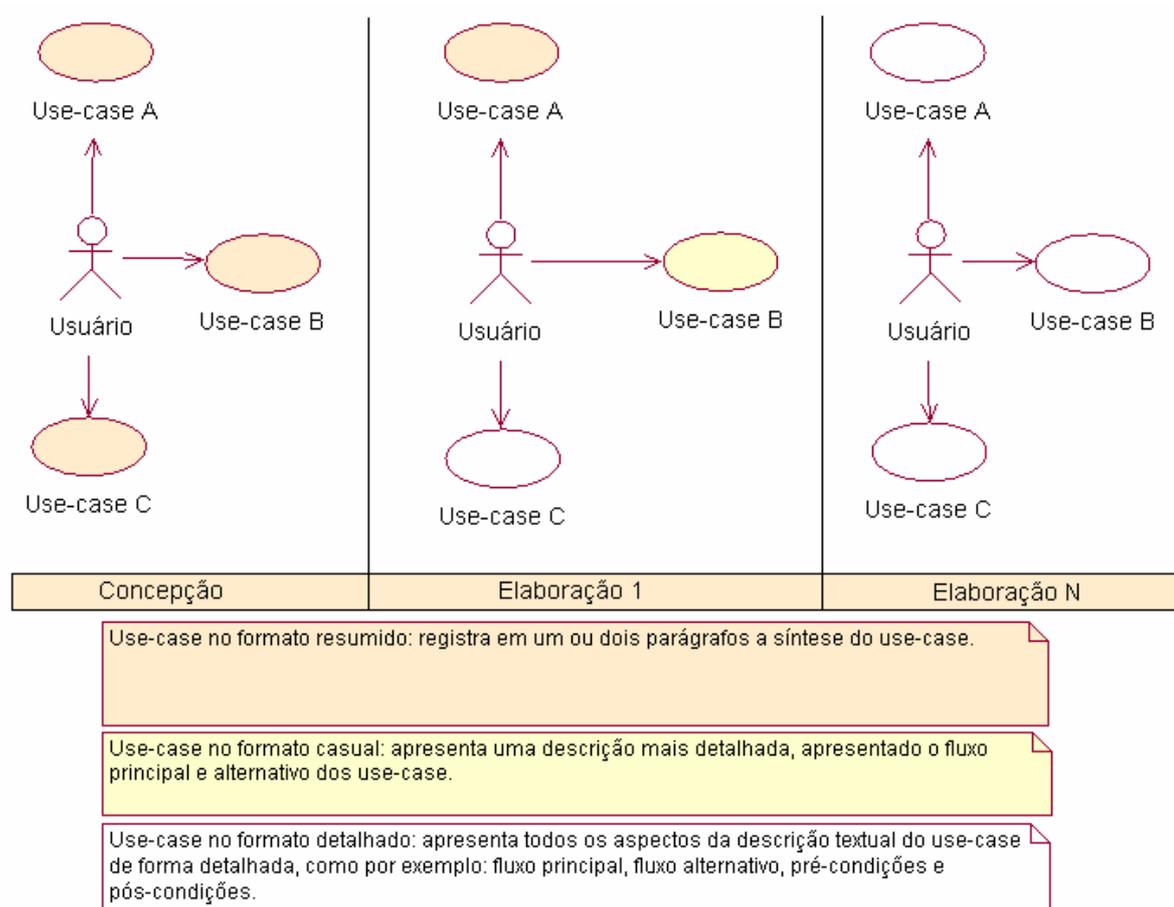


Figura 11: Detalhamento dos use-cases no decorrer das iterações

Larman (2003) sugere que durante as primeiras iterações apenas 10% dos use-cases estejam no formato completo, e que sejam gradativamente escritos no formato detalhado, de forma que ao final da última iteração de elaboração cerca de 90% esteja escrito no formato completo.

Kulak (2003) nomeia três iterações chamadas *Facade*, *Filled* e *Focused* para representar o nível de detalhe utilizado nos use-cases durante o processo de desenvolvimento. Embora a nomenclatura seja outra, o princípio de não se ater muito aos detalhes na primeira iteração e ir detalhando os use-cases com o passar das iterações permanece o mesmo.

3.4.6. Verificação e validação dos requisitos

Embora a engenharia de requisitos represente um grande desafio no processo de desenvolvimento de software, a não utilização de mecanismos de

avaliação e validação do produto resultante, podem comprometer o produto final gerado.

A verificação é uma atividade que avalia se um determinado sistema ou componente de software está de acordo com algum critério pré-definido no início da atividade (WIEGERS, 2003). As atividades de verificação auxiliam a equipe de desenvolvimento a conferir se o sistema que está sendo construído atende os requisitos de software definidos. Leffingwell (2003) ainda define uma lista mínima de itens a serem verificados:

1. Se as características realmente expressam as necessidades dos usuários;
2. Se os use cases e os requisitos derivados das características realmente suportam as características;
3. Se os use cases foram considerados no design;
4. Se o design suporta os aspectos funcionais e não do comportamento do sistema;
5. O código realmente está conforme os objetivos do desenvolvimento;
6. O plano de teste provê cobertura a todos os requisitos e use-case que foram desenvolvidos.

A utilização de mecanismos de rastreamento suporta as atividades de verificação das características implementadas. Desta forma a equipe de desenvolvimento acompanha a evolução de uma necessidade do usuário até se tornar um requisito de software e conseqüente implementação, garantindo assim que as necessidades dos usuários serão todas atendidas. A Figura 12 mostra a relação de rastreamento entre necessidades, características, requisitos de software e use case adaptadas da recomendação de Leffingwell (2003).

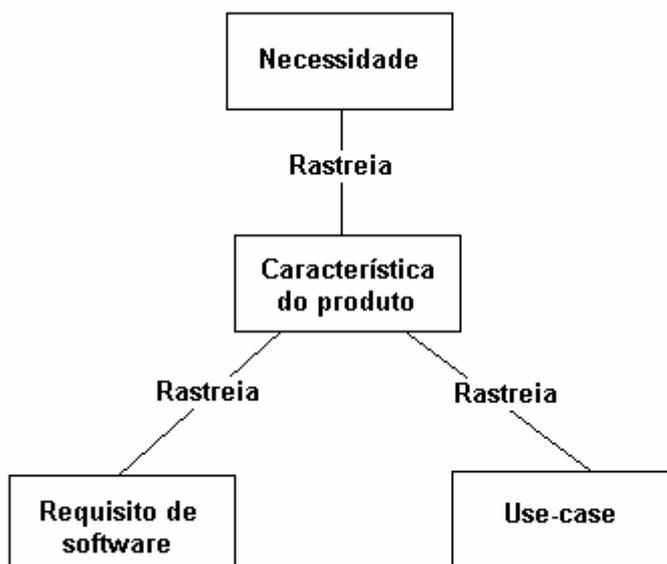


Figura 12: Relação de rastreamento

Fonte: Adaptado de Leffingwell (2003)

A validação confronta a implementação do sistema conferindo se o resultado está de acordo com os requisitos definidos (WIEGERS, 2003). É uma técnica que auxilia a equipe de desenvolvimento a verificar se o sistema que está sendo construído está correto.

Durante o processo de desenvolvimento muitos requisitos de software poderão mudar seja por fatores internos ou externos (LEFFINGWELL, 2003). Os fatores internos normalmente referem-se a problemas ocorridos durante o processo de levantamento dos requisitos e normalmente são administráveis. As mudanças introduzidas por modificações em legislação, surgimento de novas tecnologias, modificação no mercado entre outras são considerados fatores externos e estão fora do controle da equipe de desenvolvimento.

A equipe de desenvolvimento deve estar ciente de que a mudança nos requisitos é inevitável e deve possuir um processo para gerenciamento de mudança. Wieggers (2003) recomenda que pelo menos os seguintes passos devam ser seguidos para gerenciar efetivamente a mudança:

1. Reconhecer e planejar a mudança: a equipe de desenvolvimento precisa estar preparada para aceitar a mudança e deve prever tempo e recursos para isto no planejamento inicial.

2. Criar uma linha base para os requisitos: definir um conjunto inicial de requisitos para facilitar a identificação e gerenciamento da mudança que forem aparecer.

3. Estabelecer um único canal de controle da mudança: deve haver um único canal que controle o impacto e a aceitação ou não da solicitação de mudança.

4. Usar uma ferramenta de controle para capturar as mudanças: a utilização de uma ferramenta formaliza as de solicitação de mudança, auxilia o rastreamento e fornece um depósito de todas as solicitações de alteração.

5. Gerenciar a mudança hierarquicamente: as introduções de modificações no sistema devem ocorrer inicialmente nas características e serem propagadas aos níveis de requisitos e assim por diante, para que se possa visualizar todo impacto da mudança.

3.5. Sumário do capítulo

Neste capítulo foi apresentado o efeito introduzido no desenvolvimento de software por um processo de gerenciamento de requisitos não eficiente. Também foram expostos os principais modelos de ciclo de vida e as considerações do gerenciamento de requisitos no contexto de cada ciclo de vida. Na última parte foram apresentados as melhores práticas da engenharia de requisitos com intuito de identificar e analisar os requisitos de forma eficiente. No próximo capítulo o Processo Unificado (UP) de desenvolvimento é customizado para atender as necessidades específicas do projeto de pesquisa, servindo de esboço onde as melhores práticas da engenharia de requisitos serão aplicadas.

4. MATERIAIS E MÉTODOS: CUSTOMIZAÇÃO DO PROCESSO DE ANÁLISE DE REQUISITOS

Neste capítulo, são apresentados os artefatos e métodos customizados do Processo Unificado (UP) de desenvolvimento para a utilização no levantamento e análise de requisitos da revisão pelos pares na aprendizagem que será abordado no 5.

4.1. Introdução

Conforme Wiegers (2003) todo projeto de desenvolvimento de software deve selecionar e adaptar um processo de desenvolvimento conforme suas necessidades específicas, considerando tamanho e conhecimento da equipe, tipo do software a ser desenvolvido, entre outros. O objetivo deste capítulo é apresentar o procedimento de documentação, análise e especificação dos requisitos que será utilizado na próxima seção para a definição de um ambiente que suporte a revisão pelos pares na aprendizagem.

O Processo Unificado (UP) tornou-se um dos processos de desenvolvimento de softwares orientados a objetos mais populares (LARMAN, 2003), combinando o ciclo de vida iterativo e desenvolvimento dirigido ao risco.

Esta dissertação considera o estudo dos assuntos relacionados à análise de requisitos, mas o processo de desenvolvimento de software representa o arcabouço maior onde a análise de requisitos se enquadra. A contextualização dos requisitos dentro do processo de desenvolvimento torna-se necessária, apesar de não ser objetivo da dissertação um estudo aprofundado sobre os processos de desenvolvimento de software.

4.2. Fases e artefatos utilizados

No desenvolvimento iterativo e incremental as atividades são organizadas em mini-projetos de duração fixa, chamadas iterações (LARMAN, 2003). Durante cada iteração diversas disciplinas são visitadas. A cada iteração há um incremento do sistema com o desenvolvimento de determinadas funcionalidades. Como o

aplicação do processo desta dissertação visa pesquisar as atividades relacionadas a disciplina de requisitos foram utilizadas 3 iterações: uma na fase de concepção e duas na fase de elaboração. Nestas iterações as quantidades de trabalho envolvidas na disciplina de análise de requisitos são mais intensas. Cada iteração deve duração fixa de três semanas, como apresenta o Quadro 4.

Quadro 4: Número de iterações por fase

	Concepção	Elaboração
Número de iterações por fase	1	2

Larman (2003) relembra que artefatos gerados durante o processo de desenvolvimento são opcionais, cada projeto deve estabelecer quais artefatos trazem valor ao projeto. O Quadro 5 lista os artefatos a serem utilizado no processo de análise de requisitos e sua definição.

Quadro 5: Conceituação dos artefatos utilizados no processo de desenvolvimento

Artefato	Definição
Documento visão	Apresenta um sumário executivo.
Especificação suplementar	Descreve os requisitos não capturados pelo modelo de use-case, como, por exemplo, os requisitos não funcionais.
Glossário	Registra a definição dos principais termos utilizado no domínio da aplicação e evitar interpretações duvidosas.
Modelo de use-case	Descreve principalmente os requisitos funcionais.
Diagrama entidade-relacionamento	Descreve o modelo de dados
Protótipo descartável	Permite a validação de conceitos a serem utilizados na solução

O Quadro 6 apresenta cada artefato utilizado no processo de desenvolvimento, indicando em qual iteração o artefato é inicialmente criado (C) e em quais iterações o artefato deve ser refinado (R).

Quadro 6: Artefatos criados (C), ou refinados (R) durante as iterações.

Artefato	Concepção	Elaboração 1	Elaboração 2	Elaboração 3
Documento visão	C	R	R	R
Especificação suplementar	C	R	R	R
Glossário	C	R	R	R
Modelo de use-case	C	R	R	R
Diagrama entidade-relacionamento			C	R
Protótipo descartável			C	R

4.3. Formato dos use-cases

Use-cases descrevem de forma textual a interação do ator primário com o sistema (COCKBURN, 2001). Amplamente aceito, representam uma ferramenta excepcional para descobrir e registrar requisitos (LARMAN, 2003). Existe uma série de qualidades de um use-case bem escrito, mas não existe um formato padrão para sua redação.

A flexibilidade da descrição textual dos use-cases permite escrevê-los em diferentes formatos. Cockburn (2001) apresenta em seu livro cerca de cinco formatos diferentes para se descrever use-cases. Neste trabalho foram utilizados três formatos na descrição dos use-cases como forma de representação do nível de refinamento exigido durante o processo de análise de requisitos:

Resumido: um parágrafo sucinto descrevendo o objetivo principal do use-case, normalmente o cenário de sucesso.

Casual: vários parágrafos, sem muita rigidez no formato, apresentam vários cenários.

Detalhado: formato mais elaborado, onde todos os passos são escritos em detalhes. O formato detalhado utilizado na aplicação do processo representa uma adaptação do gabarito apresentando por Cockburn (2001), conforme Quadro 7.

Quadro 7: Gabarito do use-case detalhado

<p>Breve descrição: apresenta em um parágrafo o objetivo principal do use-case;</p> <p>Ator primário: o nome do ator que dispara o use-case;</p> <p>Stakeholders e interesses: listam os <i>stakeholders</i> e os interesses principais que possuem no use-case;</p> <p>Pré-condições: condições que devem ser verdadeiras antes do início do use-case;</p> <p>Fluxo principal: passos necessários para atingir o objetivo do principal cenário do use-case;</p> <p>Fluxo alternativo: representam outros cenários do use-case e exceções. São originados através do fluxo principal.</p> <p>Requisitos especiais: requisitos não funcionais, atributos de qualidade ou restrições específicas ao use-case.</p>
--

Fonte: Adaptado de Cockburn (2001)

4.4. Fluxo de trabalho

No desenvolvimento iterativo e incremental proposto pelo UP os requisitos são descobertos e registrados gradativamente, principalmente durante as fases de concepção e elaboração por meio da disciplina de requisitos (KROLL, 2003). O fluxo de trabalho da disciplina de requisitos é composto por uma série de atividades, que orientam a execução dos trabalhos a serem desenvolvidos, conforme apresentado na Figura 13. As subseções a seguir detalham cada atividade componente do fluxo de trabalho utilizado na aplicação do processo.

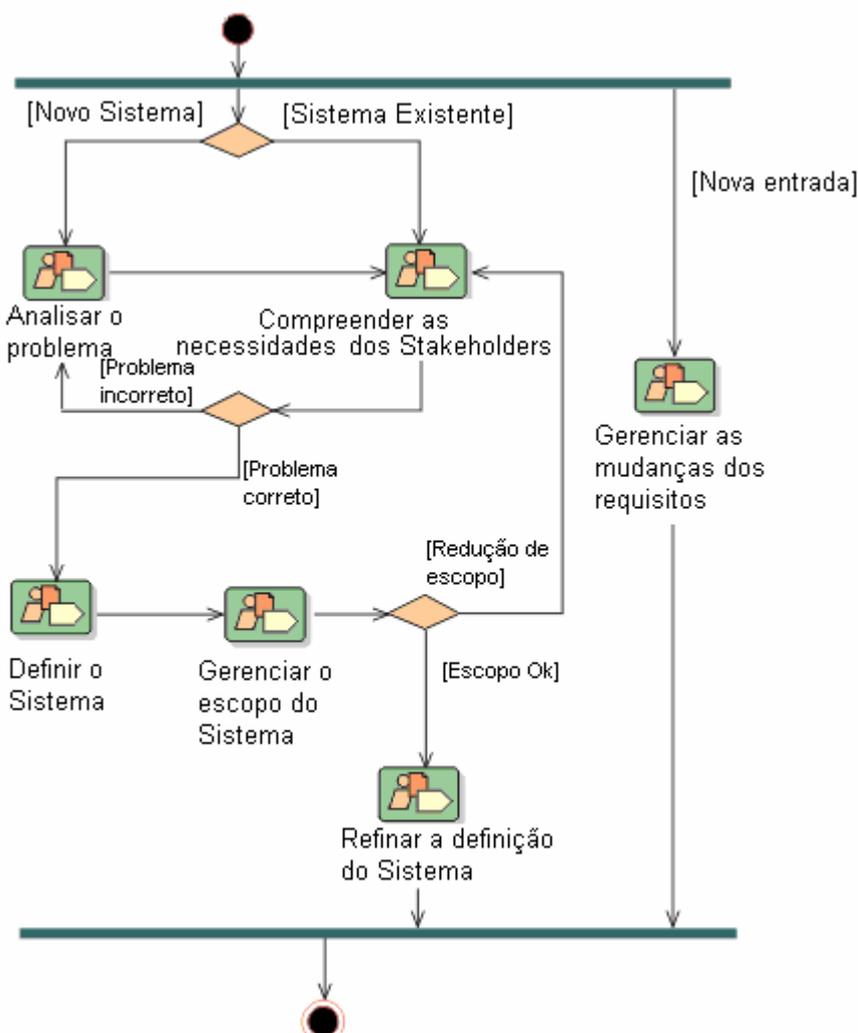


Figura 13: Atividades e artefatos componentes da disciplina de requisitos

Fonte: Adaptado de Kruchten (2003)

4.4.1. Analisar o problema

Algumas aplicações de software são desenvolvidas para aproveitar uma oportunidade de mercado, mas a maioria dos sistemas é construída para resolver um problema em particular. Contudo as equipes de desenvolvimento gastam muito pouco tempo na compreensão dos reais problemas dos negócios e o ambiente onde a aplicação será implantada (WINDLE, 2002).

A principal meta desta atividade é produzir a primeira versão do documento visão contendo uma formulação clara e objetiva dos problemas a serem solucionados, a identificação dos *stakeholders*, bem como definir as fronteiras e as

principais funcionalidades e restrições do sistema. O Quadro 8 apresenta um formato padronizado para declaração do problema.

Quadro 8: Formato padronizado para declaração do problema

Elementos	Descrição
O problema	<descrever o problema>
Afeta	<listar os afetados pelo problema>
O impacto disto é	<listar os impactos do problema>
Benefícios de uma solução	<listas os principais benefícios de uma solução>

Fonte: Adaptado de Leffingwell (2003)

A maior parte dos *stakeholders* serão futuros usuários do sistema, mas a equipe de desenvolvimento deve identificar todos os grupos de *stakeholders* afetados pelo desenvolvimento da solução, principalmente *stakeholders* que não são usuários diretos. A identificação dos principais grupos de *stakeholders* torna-se imprescindível para o sucesso do desenvolvimento de solução efetiva (WIEGERS, 2003). Os grupos de *stakeholders* identificados devem ser registrados no documento visão. O Quadro 9 apresenta um formato padronizado na identificação e registro dos principais grupos de *stakeholders*.

Quadro 9: Formato padronizado para registro de *stakeholders*

Nome	Representa	Papel
Nome do grupo de <i>stakeholder</i> .	Descrição resumida sobre o que o grupo representa.	Descrição resumida do papel desempenhado.

Fonte: Adaptado de Leffingwell (2003)

Durante a atividade de definição problema deve-se criar o glossário com a representação dos termos relativo ao problema em estudo. Sua função é de registrar o vocabulário a ser utilizado durante o projeto, principalmente nas descrições textuais dos use-cases e evitar interpretações dúbias. Na aplicação do processo adotou-se a utilização de um artefato exclusivo para registro do vocabulário contendo os termos utilizados e uma breve descrição. O glossário deve ser um documento atualizado continuamente durante todo o ciclo de desenvolvimento do projeto.

4.4.2. Compreender as necessidades dos *stakeholders*

O principal propósito desta atividade é elicitare as reais necessidades dos *stakeholders* previamente identificados. Esta atividade visa descobrir (identificar, deduzir, extrair, evocar, obter) as necessidades que futuro sistema deverá atender. Segundo Windle (2002) as equipes de desenvolvimento devem utilizar técnicas entrevista, questionário, dentre outras para auxiliar na compreensão das reais necessidades dos usuários.

Com objetivo de entendimento das reais necessidades dos usuários foram utilizadas as seguintes técnicas:

- a) revisão bibliográfica sobre a utilização da revisão pelos pares na ciência e suas peculiaridades na utilização em sala de aula;
- b) levantamento das peculiaridades do processo de revisão pelos pares na educação por meio da observação de aplicações práticas;
- c) investigação e identificação das principais necessidades requeridas por meio de entrevistas com especialistas em revisão pelos pares;
- d) revisar e eleger as principais características das ferramentas de revisão pelos pares mais utilizadas no meio científico;

As necessidades identificadas foram registradas no documento visão, no modelo de use-case e nas especificações suplementares.

4.4.3. Definir o sistema

Segundo Kulak (2003), definir o sistema significa traduzir e organizar as necessidades dos usuários e *stakeholders* em algo útil para o processo de desenvolvimento do sistema. Os requisitos de alto nível previamente registrados devem ser convertidos para requisitos de software com qualidade e riqueza de detalhes suficiente para guiar os desenvolvedores as atividades design, codificação e testes (WIEGERS, 2003).

4.4.4. Gerenciar o escopo do sistema

O desenvolvimento iterativo e incremental abraça a mudança como parte inevitável do processo, mas estas mudanças devem acontecer de uma forma controlada e consciente. Caso toda solicitação de modificação nos requisitos do projeto for aceita, provavelmente o projeto nunca será terminado (WIEGERS, 2003).

Gerenciar o escopo é uma atividade contínua que administra o escopo do projeto, seu planejamento em iterações bem como suas mudanças. Avaliando os atributos dos requisitos a equipe de desenvolvimento pode definir a lista do que será desenvolvido em cada iteração visando minimizar o risco conhecido do projeto e atender as características mais prioritárias. O Quadro 10 apresenta conjunto de atributos utilizados na aplicação do processo para gerenciamento do escopo e definição use-cases e requisitos a serem priorizados para a próxima iteração.

Quadro 10: Atributos utilizados no gerenciamento do escopo

Atributo	Conceito	Valores possíveis
Prioridade	Importância do use-case para o projeto.	indispensável, desejável, acessório
Risco	Risco associado ao use-case. Pode ser risco tecnológico, de negócio ou arquitetural	alto, médio, baixo
Esforço	Estimativa rudimentar sobre o esforço necessário para o desenvolvimento	alta, média, baixa
Dificuldade	Dificuldade na implementação do use-case	alta, média, baixa
Estabilidade	Quão estável o use-case se apresenta	alta, média, baixa
Status	Situação que o use-case encontra-se no momento	Proposto, revisado, validado, fila de espera

Kruchten (2003) define a lista inicial de requisitos a ser implementada em cada iteração como o plano de iteração. Durante o projeto aparecerão novas características a ser alocada a lista de requisitos de determinada iteração. A equipe de desenvolvimento deve estabelecer um procedimento para o registro destas solicitações de mudanças, bem como criar um “comitê” que avalie o impacto da mudança nas solicitações mais complexas que envolvam modificação no escopo do projeto. Esta avaliação deve ser baseada nos atributos de gerenciamento do escopo (WIEGERS, 2003).

4.4.5. Refinar a definição do sistema

Com base nos use-cases e documento visão, a equipe de desenvolvimento precisa detalhar os requisitos de software visando um perfeito entendimento com o cliente sobre quais as funcionalidades que o software deverá possuir (KRUCHTEN, 2003). Esta atividade revisa todo material coletado até o momento e produz use-cases em formato detalhado, permitindo a compreensão mais aprofundada das funcionalidades do futuro sistema. Os requisitos não funcionais identificados devem ser registrados em um documento suplementar.

4.4.6. Gerenciar as mudanças nos requisitos

Segundo Kruchten (2003) as solicitações de mudanças nos requisitos são inevitáveis, seja por fatores internos ou externos ao projeto. A equipe de desenvolvimento deve avaliar o impacto das solicitações e garantir que as modificações aprovadas sejam propagadas para todos os artefatos.

Solicitações de alteração podem ter o custo de implementação muito alto ou contradizer com requisitos já implementados. Conforme Young (2004) deve existir um comitê formado por representantes da equipe de desenvolvimento e do cliente visando avaliar quais solicitações serão atendidas. Este comitê pode basear sua decisão nos atributos dos requisitos.

4.5. Ferramentas utilizadas

Atualmente a Rational Software Corporation é uma das empresas líderes no mercado de ferramentas para modelagem orientada a objetos, possuindo diversos produtos que auxiliam nas atividades de desenvolvimento de software.

Visando organizar os artefatos e suportar o processo de desenvolvimento escolhido, esta dissertação utilizou algumas ferramentas *Rational Software Corporation*. A escolha destas ferramentas não levou em consideração nenhum critério técnico, apenas a facilidade de acesso que o pesquisador teve a ferramenta.

Para documentação e gerenciamento dos requisitos foi utilizada a ferramenta Rational RequisitePro. Esta ferramenta permite registrar os requisitos em um banco de dados além de permitir a definição de atributos e elaboração das matrizes de rastreamentos entre requisitos. O Rational Rose foi utilizado na elaboração dos diagramas de use-cases e integrado com o Rational RequisitePro para garantir o rastreamento dos requisitos.

5. REQUISITOS DE SOFTWARE DA REVISÃO PELOS PARES NA APRENDIZAGEM

Esta dissertação pretende elicitar, analisar e documentar os principais requisitos de um software que suporte a aplicação da revisão pelos pares na aprendizagem utilizando as melhores práticas da engenharia de requisitos dentro do processo desenvolvimento iterativo e incremental do processo UP conforme sugerida no capítulo 4. O escopo do trabalho está voltado para as atividades relacionadas a análise de requisitos visando melhor compreender e documentar as necessidades dos usuários.

Durante a elaboração da pesquisa alguns artefatos do processo de desenvolvimento foram criados e atualizados de forma paralela, conforme preconiza a estratégia iterativa e incremental. Entretanto, na elaboração deste relatório de pesquisa há necessidade de se descrever a aplicação do processo de forma linear, sem muita repetição. Neste contexto, este relatório de pesquisa apresenta os artefatos de desenvolvimento da forma mais fidedigna possível sem torná-lo muito maçante e repetitivo.

Nas próximas seções deste capítulo os resultados obtidos na aplicação do processo customizado de análise de requisitos são detalhados. Na primeira seção o escopo do projeto é definido, seguido pela aquisição de conhecimento específico por meio de estudo das fontes de informação disponíveis, conforme apresentado nas seções 5.2 e 5.3. Com base neste conhecimento e interação com os envolvidos no projeto, definiram-se claramente os principais problemas a serem solucionados, bem como os limites e restrições de projeto. As seções 5.7 e 5.8 apresentam o registro e a definição dos *stakeholders* envolvidos, para que a seção 5.10 possa expor as principais características de alto nível identificadas. As seções seguintes apresentam como as características foram transcritas para use-case e o seu refinamento gradativo, bem como ilustra o processo de rastreamento dos requisitos utilizado. Por final, são apresentados os requisitos não funcionais, glossário de termos e artefatos complementares utilizados para auxiliar o processo análise de requisitos.

5.1. Escopo do sistema a ser desenvolvido

O sistema a ser modelado deve permitir a simulação de um congresso científico em sala de aula, utilizando o processo de revisão pelos pares fundamentado nas etapas sugeridas por Kern et. al. (2002) e detalhado em 2.2. O desenvolvimento visa prover um software que apóie o trabalho administrativo, permitindo a exploração pedagógica e divulgação da abordagem.

5.2. Origem dos requisitos identificados

A equipe de desenvolvimento deve inicialmente adquirir conhecimento específico na área objeto do desenvolvimento (WINDLE, 2002). A maneira mais usual é a realização de uma revisão bibliográfica visando dominar o linguajar específico, sem se tornar um perito no assunto. Além de prover background necessário, as informações coletadas podem ser utilizadas como fonte para a análise de requisitos.

No problema em estudo inicialmente foi realizada uma pesquisa bibliográfica buscando informações referentes a utilização da revisão pelos pares e sua aplicação na aprendizagem. As informações resultantes da pesquisa e pertinentes ao escopo do projeto foram documentadas de forma textual em um artefato chamado "Solicitação dos usuários". Neste documento as informações coletadas foram registradas e para cada fonte de informação realizou-se a elicitación das necessidades dos usuários (válidas ou inválidas), que começaram a serem rastreadas dentro do processo de análise de requisitos. A Figura 14 apresenta todas as fontes de dados utilizadas como origem para os requisitos.



Figura 14: Fontes utilizadas na elicitação dos requisitos

A lista a seguir descreve as fontes de informação utilizadas no processo de análise de requisitos:

- revisão bibliográfica: revisão de artigos esclarecendo a abordagem de revisão pelos pares;
- participação de processo: anotações realizadas através da participação da simulação de um congresso em sala de aula, no papel de autor e revisor;
- entrevista com *stakeholders*: levantamento de uma lista de necessidades por meio de entrevista com especialistas;
- estudo exploratório: identificação das principais características que a abordagem na aprendizagem deve possuir com base nos principais softwares de gerência de conferência existentes no mercado.

As entrevistas com *stakeholders* foram realizadas através de um questionário com perguntas fechadas e algumas abertas enviadas através de e-mail a vinte especialistas na organização de congressos científicos. O resultado desta ação foram apenas três respostas, que pouco contribuíram aos requisitos de software

previamente identificados. Por este motivo estas entrevistas foram desconsideradas na elaboração deste relatório de pesquisa.

5.3. Estudo exploratório

Existem diversos softwares desenvolvidos para suportar a gerência de conferências científicas (SNODGRASS, 1999). Apesar de não atenderem plenamente as necessidades específicas da revisão pelos pares na aprendizagem, constituem excelente fonte de estudo para identificação de requisitos (WINDLE, 2002).

Realizou-se uma busca na Internet por meio do sítio de pesquisa Google (www.google.com.br) visando identificar alguns softwares nesta área e quais principais características que possuem no contexto da revisão pelos pares na aprendizagem. Percebe-se que existem poucos softwares de gerência de conferência comerciais, a maioria são softwares acadêmicos desenvolvidos inicialmente para atender as necessidades de uma conferência, que acabam evoluindo para atender as necessidades de outros eventos, tornando-se plenos na gerência de conferências. O Quadro 11 apresenta os softwares encontrados através da busca, bem como as características pertinentes à aplicação da revisão pelos pares na aprendizagem. A realização desta investigação objetiva a identificação de requisitos atendidos pelos softwares de gerência de conferências e encontrar características interessantes a serem consideradas na aplicação do processo.

Quadro 11: Sumário das principais características de softwares de gerência de conferências no que tange a abordagem na aprendizagem

Software	Submissão de artigos	Alocação de revisores	Submissão da revisão	Comentário de revisores	Encontro do comitê de avaliação	Comunicação aos autores
AAA S/W	S	S	-	-	-	-
ACM CR DB	-	S	S	-	-	S
ConfMan	S	S	S	S	S	S
CyberChair	S	S	S	S	S	S
EDAS	-	S	S	S	-	S

ICSE S/W	S	S	S	-	-	S
munge+	-	-	S	-	-	S
PCA	-	S	-	-	-	S
PUMA	S	S	S	-	-	S
ReviewIt	-	S	S	-	-	S
SAC/WWW S/W	S	-	S	-	?	S
SIGACT EPC	-	-	S	S	S	S
SIGACT Sub	S	-	-	-	-	-
SIGDA S/W Conf	S	S	S	S	-	S
SIGDA S/W Jour	S	S	-	-	-	-
SIGGRAPH Online	S	S	S	-	?	?
SIGPLAN S/W	S	S	S	-	-	S
START v1	S	S	S	S	S	S
WIMPE	S	S	S	?	S	?
WitanWeb	S	S	S	S	P	S

Legenda:

S: Sim

P: Parcial

?: Não foi possível identificar

-: Não

Fonte: Adaptado de Snodgrass (1999)

Além da avaliação das características implementadas pelos softwares, algumas características chamaram a atenção do pesquisador, são apresentadas a seguir:

1. Seleção voluntária de novos artigos: permitir ao revisor escolher novos artigos para revisar;
2. Indicação de preferência para revisão: permitir ao revisor indicar preferência para realizar a revisão de um artigo de seu interesse. Esta indicação pode auxiliar no momento da alocação dos artigos aos revisores;
3. Discussão sobre conflito de revisões: permite que revisores discutam anonimamente sobre disparidades entre suas avaliações sobre um artigo.

5.4. Acordo na definição do problema

Conforme apresentado no Capítulo 3.4.1, todos os envolvidos em um projeto de desenvolvimento de software precisam compartilhar uma visão em comum sobre os principais problemas que se pretende resolver. Young (2004) afirma que se deve escrever o problema de forma clara e concisa para garantir que todos os envolvidos conheçam o objetivo em comum pelo qual está se trabalhando. O Quadro 12 apresenta a declaração do problema no formato estruturado conforme sugerido por Leffingwell (2003).

Quadro 12: Definição do problema no formato estruturado

O problema	O excesso de burocracia na aplicação da revisão pelos pares na aprendizagem
Afeta	Alunos, professores, comunidade científica em geral e a qualidade da formação acadêmica.
O impacto	A não utilização da abordagem na aprendizagem
Uma solução de sucesso	Permitir a simulação de um congresso, no qual professores e alunos possuam um ambiente que suporte a revisão pelos pares, facilitando e minimizando o trabalho administrativo envolvido além de facilitar a colaboração entre os participantes.

5.5. Definição do limite da solução

Wiegiers (2003) enfatiza a necessidade de elaboração de um diagrama de contexto para representar os limites impostos à solução, bem como as correlações com outros sistemas. Também apresenta de forma simples e com alto nível de abstração as entidades externas ao sistema, como usuários, dispositivos de hardware, e outros sistemas de informação. A Figura 15 apresenta o diagrama de contexto elaborado para a representação dos limites da solução que está sendo proposta.

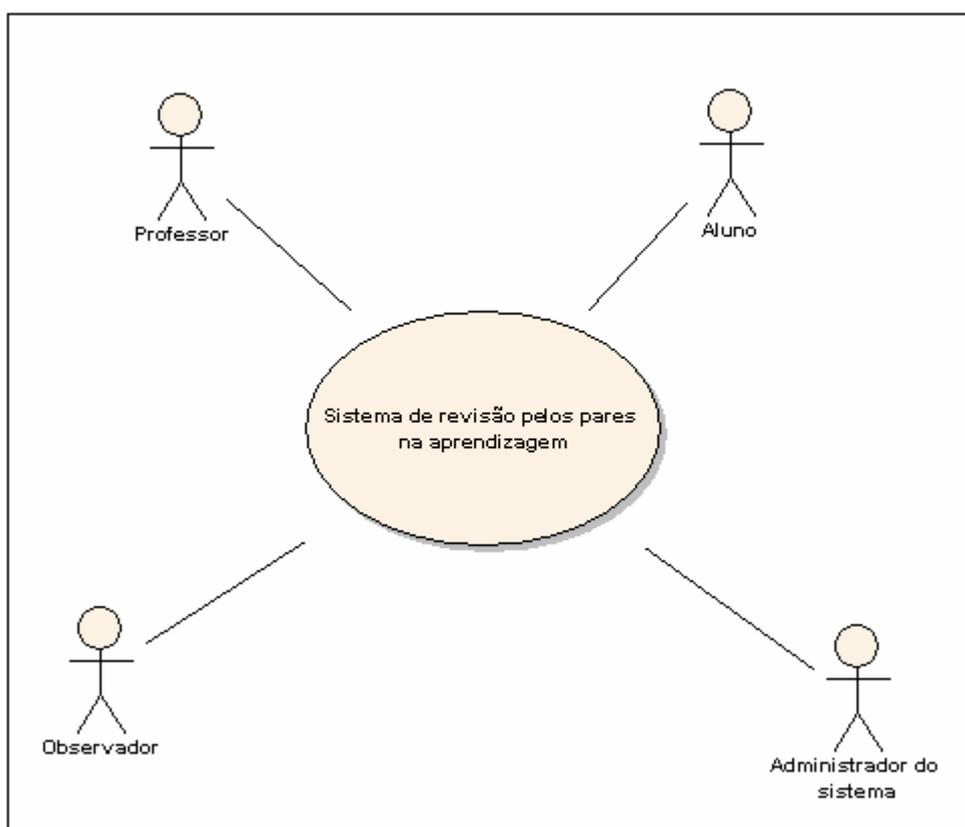


Figura 15 : Diagrama de contexto do sistema

No caso da solução que está sendo inicialmente proposta não foi priorizada a integração com outros sistemas de informação. A integração com um sistema de matrícula, por exemplo, pode permitir a transferência dos dados dos alunos, evitando a necessidade do cadastramento no ambiente. A análise e o desenvolvimento destas características ficaram postergados para evoluções futuras do produto.

5.6. Restrições impostas à solução

No problema em estudo houve algumas decisões de projeto que restringiram as escolhas disponíveis para o design e construção do produto. O Quadro 13 lista as restrições a serem impostas ao produto e o motivo pelas quais elas existem.

Quadro 13: Restrições impostas à solução

Restrição	Justificativa
Ser compatível com banco de dados padrão SQL ANSI 92.	Diferentes instituições de ensino utilizam diferentes bancos de dados. Uma aplicação compatível com SQL ANSI 92 irá permitir a portabilidade para qualquer banco de dados compatível com o padrão.
Não requerer licenças comerciais para a utilização do ambiente	A necessidade de licenças comerciais para a execução do software pode ser uma barreira na adoção da plataforma no meio acadêmico.
Tecnologia de acesso menos restrita possível	Torna-se quase impossível exigir a todos os participantes um software ou hardware específico para acessar o ambiente.

5.7. Identificação dos usuários e *stakeholders*

Diferentes grupos de usuários possuem diferentes necessidades (WINDLE, 2002). Neste contexto há necessidade de se identificar e registrar os principais tipos de usuários e *stakeholders* em um formato padronizado. O documento visão completo desenvolvido para o projeto encontra-se no apêndice A. O Quadro 14 apresenta os principais usuários e *stakeholders* identificados e registrados no documento visão.

Quadro 14: *Stakeholders* e usuários do futuro sistema

Nome	Representa	Papel na análise de requisitos
Professores	Professores que aplicam a abordagem da revisão pelos pares na aprendizagem.	Fonte principal para a análise de requisitos.
Alunos	Participam da simulação do congresso em sala de aula.	Possuem, principalmente, requisitos não funcionais.
Pedagogos e entidades externas	Pessoas externas ao processo, mas que possuem interesse na abordagem.	Possuem necessidades de informações pedagógicas e resumidas sobre os resultados das simulações de congresso.

5.8. Atores

Na solução proposta foram identificados cinco atores, conforme ilustrado na Figura 16. O ator Administrador do Sistema representa um ou mais usuários que executam as funções administrativas do software, por exemplo, a gestão de usuários. Três atores (Revisor, Editor, Autor) representam os principais envolvidos na simulação do congresso em sala de aula.

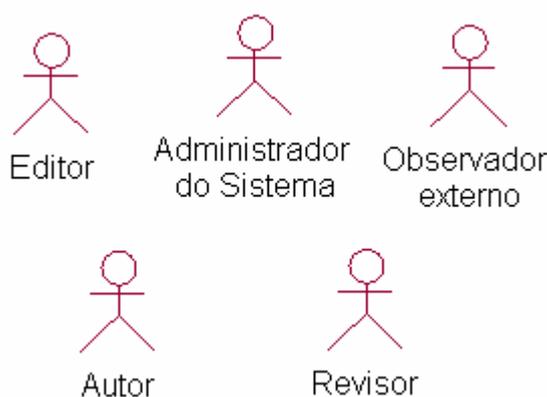


Figura 16: Atores do sistema

O Quadro 15 descreve os atores e seu principal interesse no ambiente, bem como a abstração do mundo real que está representando.

Quadro 15: Atores e principal interesse

Ator	Principal interesse	Abstração
Editor	Organização e condução da simulação do congresso em sala de aula	Professor
Autor	Publicação do material que será avaliado	Aluno
Revisor	Avaliação do material submetido	Aluno
Observador externo	Representa pessoas e entidades externas ao processo que possuem interesse na observação da aplicação da abordagem	Pedagogos e entidades externas
Administrador do sistema	Manter o processo operacional	Auxiliar administrativo ou o próprio professor

5.9. Priorização dos requisitos

Conforme apresentado, os atributos orientam a equipe de desenvolvimento na elaboração do plano de iteração indicando quais requisitos carecem ser detalhados ou melhor explorados. Segundo Wieggers (2003), a priorização de requisitos é uma maneira de lidar com a competição por recursos limitados, objetivando entregar as funcionalidades de grande importância o mais cedo possível. Por questões didáticas, neste projeto decidiu-se pela utilização dos atributos na priorização dos requisitos.

No processo iterativo e incremental o risco deve dirigir o desenvolvimento, ou seja, os requisitos que possuírem maior risco e prioridade devem ser desenvolvidos nas iterações iniciais. Neste contexto, o Quadro 16 apresenta algumas combinações

de valores possíveis para os atributos dos requisitos e a estratégia adotada pela equipe de desenvolvimento.

Quadro 16: Algumas combinações de valores dos atributos e a estratégia adotada

Atributo	Valor	Valor	Valor	Valor
Prioridade	Indispensável	Desejável	Desejável	Acessório
Risco	Alto	Alto	Baixo	Alto
Esforço	Alto	Baixo	Alto	Alto
Dificuldade	Alta	Baixa	Baixa	Alta
Estabilidade	Alta	Alta	Alta	Alta
Estratégia adotada	Priorizar desenvolvimento	Priorizar desenvolvimento	Desenvolver nas iterações finais	Colocar na fila de espera

Após a elicitação das necessidades dos usuários a equipe de desenvolvimento valorou os atributos prioridade, risco, esforço, dificuldade e estabilidade. Neste momento, a definição de alguns atributos, como por exemplo esforço, pode ser muito rudimentar e suscetível a erros, que se tornavam estimativas mais realísticas no decorrer das iterações. Através da avaliação dos atributos dos requisitos foi possível selecionar e priorizar as necessidades dos usuários que deveriam ser transcritas na forma de características do produto e posteriormente em use-cases. As necessidades que não foram selecionadas para constar no escopo inicial do projeto tinham o atributo situação atualizado para “fila de espera”. Estas necessidades não foram descartadas, pois poderão ser abordadas em um ciclo de desenvolvimento futuro.

Após as características terem sido inferidas, a equipe de desenvolvimento realiza a valoração de seus atributos. O mesmo processo é realizado quando da transcrição das características para use-cases nas etapas seguintes. Isto se faz necessário, pois durante o desenrolar do processo de desenvolvimento pode ser necessário reajustar os valores dos atributos para, por exemplo, refletir uma estimativa de esforço mais realista, ou a utilização de uma nova tecnologia que reduzirá o tempo de desenvolvimento.

5.10. Características de alto nível

A elicitação inicial nas fontes de informação resultou em vinte e seis características de alto nível, registradas no documento visão. Provavelmente existem mais características, contudo a equipe de desenvolvimento deve encontrar os requisitos de forma iterativa e incremental com o decorrer das iterações. O Quadro 17 ilustra as vinte e seis características levantadas e registradas no documento visão.

Quadro 17: Características de alto nível levantadas

<p>Simulação de um congresso em sala de aula</p> <p>Apoio a avaliação dos participantes do congresso</p> <p>Fluxo de trabalho variável, com descrição, objetivo e data limite para cada etapa do fluxo</p> <p>Lembrete de proximidade do encerramento de determinadas etapas</p> <p>Multilíngüe</p> <p>Acessível remotamente</p> <p>Apoio a colaboração entre os participantes, por meio de publicação e consulta a referências</p> <p>Trâmite automático de documentos</p> <p>Inscrição de participantes (autores, revisores, editores)</p> <p>Alocação automática de revisores</p> <p>Alocação manual de revisores</p> <p>Expansão e personalização do ambiente</p> <p>Publicação do material resultante (caderno técnico)</p> <p>Escalonamento de apresentações</p> <p>Apoio a avaliação de apresentações de artigos</p> <p>Sumariar revisões</p> <p>Apoio a elaboração do esboço do artigo e avaliação do esboço</p> <p>Divulgação da abordagem da revisão pelos pares na aprendizagem</p> <p>Convencimento do docente na utilização ferramenta</p> <p>Avaliação do processo pelos participantes</p> <p>Permitir a participação de pedagogos e interessados como observadores externos do processo</p> <p>Definição de documentos padrão (template) para artigos, relatórios de revisão etc.</p> <p>Definição dos tipos de arquivos aceitos por uma conferência</p> <p>Apoio a identificação de inconsistência entre revisões de um mesmo artigo</p> <p>Seleção da modalidade do processo de revisão- cega, ou duplamente cega</p> <p>Seleção de critérios de avaliação a serem utilizados na conferência</p>

Com base nestas características de alto nível percebe-se que além de disponibilizar uma ferramenta que suporte as atividades operacionais da revisão pelos pares na educação há necessidade de funcionalidades que apóiem a divulgação da abordagem e despertem o interesse do docente na sua aplicação. Esta necessidade está fundamentada na peculiaridade que mercado da informação possui, onde é a oferta que determina a demanda por informação (BARRETO, 1999).

5.11. Diagrama de use-case no formato resumido

Os use-cases no formato resumido, embora sucintos e vagos, permitem uma compreensão rápida do grau de complexidade e funcionamento do sistema a ser desenvolvido, sem se ater a detalhes (LARMAN, 2003). As características do produto identificadas e documentadas no documento visão foram transcritas na forma de use-case resumidos visando delinear o escopo do projeto. A Figura 17 apresenta uma ilustração do diagrama de use-case da fase de elaboração, onde se percebem os atores identificados e os principais use-cases que iniciam.

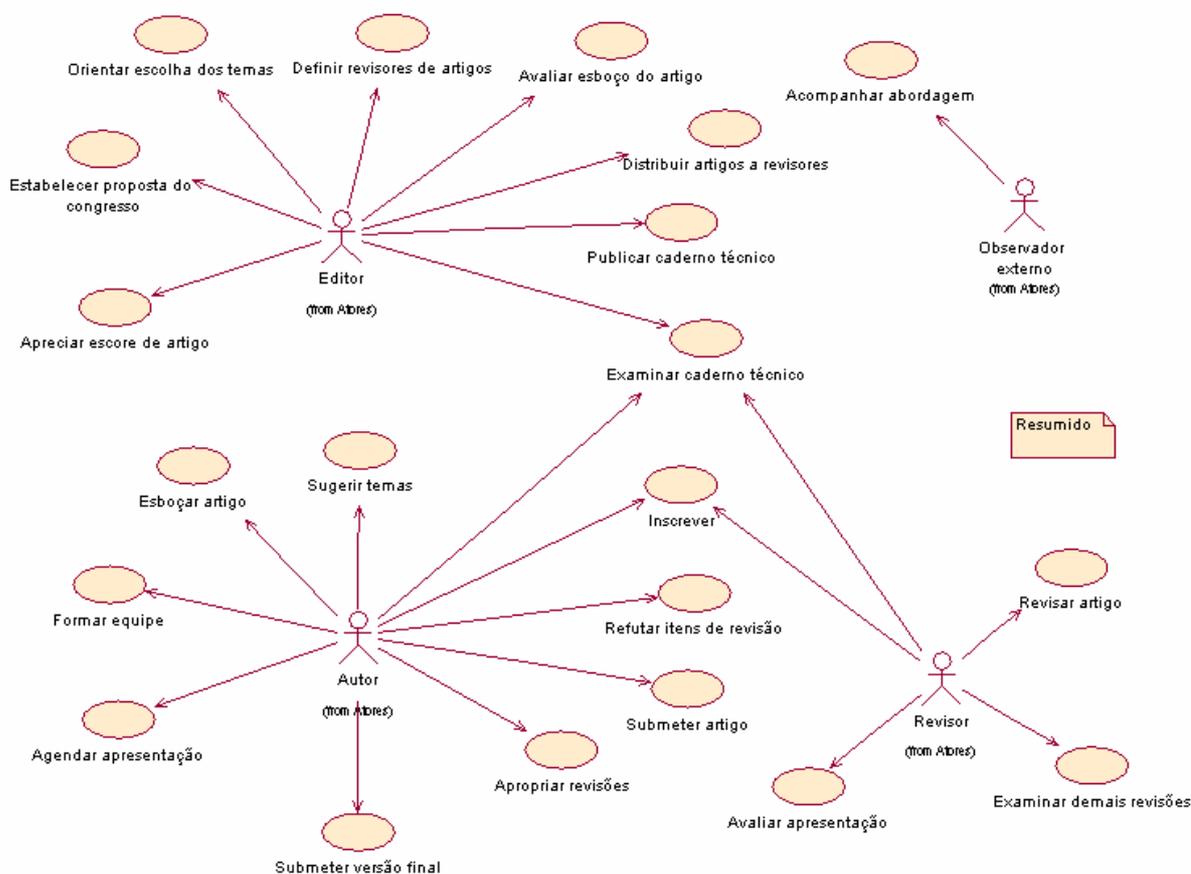


Figura 17: Use-case identificados na fase de elaboração

5.12. Detalhamento dos use-cases

Os use-cases da fase de concepção com maior risco e menor estabilidade foram transcritos do formato resumido para o formato detalhado na primeira iteração da fase de elaboração. Os use-cases mais estáveis e com menor risco foram apenas transcritos para o formato casual, sem se a necessidade de detalhamento. A Figura 18 apresenta o diagrama de use-case no decorrer da primeira iteração da fase elaboração contendo apenas use-cases no formato detalhado ou casual.

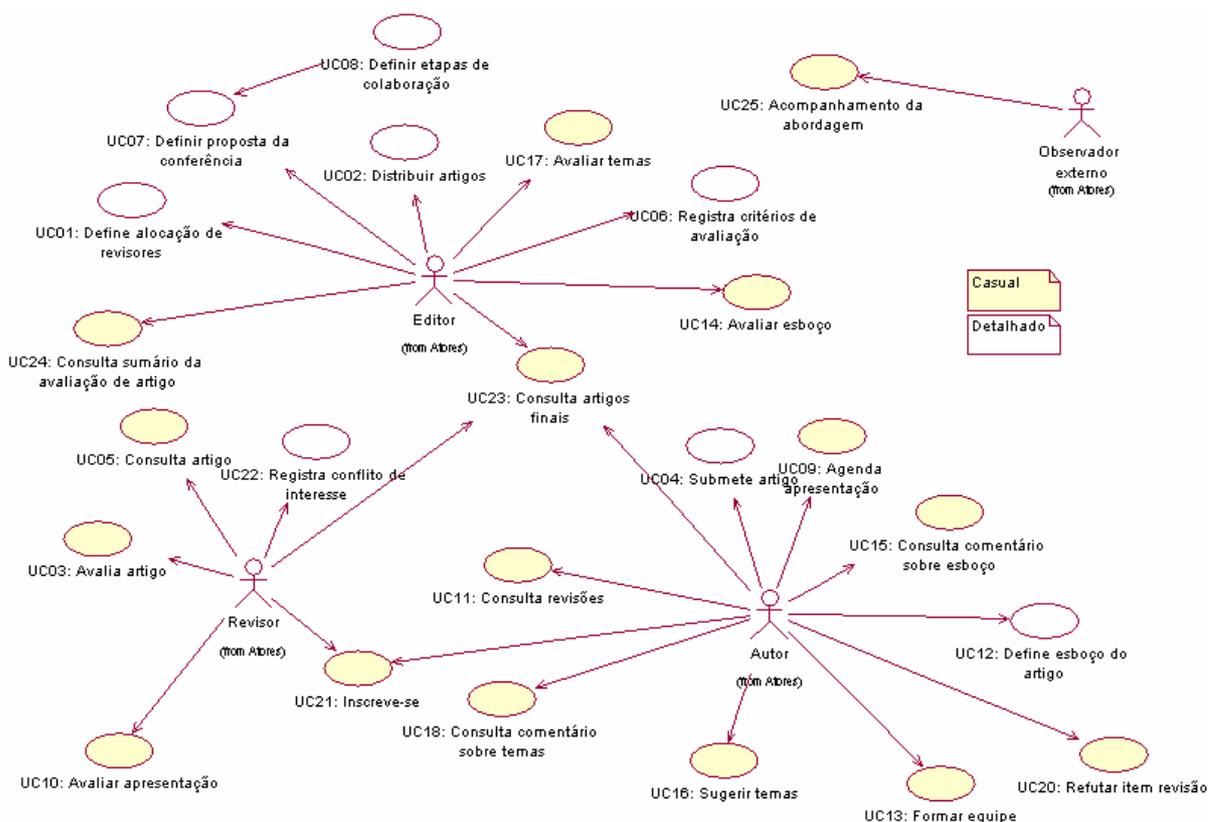


Figura 18: Diagrama de use-case na primeira iteração da fase de elaboração

Na transcorrer da segunda iteração da fase de elaboração a equipe de desenvolvimento transcreveu a maioria dos use-cases que se encontravam no formato casual para o formato detalhado. Neste ponto cerca de noventa e cinco por cento dos use-cases se encontram no formato detalhado. Os poucos use-cases restantes serão detalhados nas iterações seguintes, visto que são use-case simples e não representam risco para o projeto. A Figura 19 apresenta o diagrama de use-cases na segunda iteração da fase de elaboração.

<u>UCE23 UC08:Definir etapas de colaboração</u>	
Breve descrição	Permitir a definição das etapas a utilizar na conferência.
Ator primário	UC07: Definir proposta para conferência
Stakeholders e interesses:	Editor: gostaria de definir as etapas constantes da conferência de forma fácil
Pré-condições	Editor identificado para o sistema.
Fluxo principal	1. O Editor informa: a. uma descrição para a etapa; b. o objetivo da etapa; c. a fase antecessora; d. a data limite da fase; 2. O Editor confirma e pode incluir mais etapas;
Fluxo alternativo	
Requisitos especiais	

Quadro 18: Exemplo de use-case no formato detalhado

5.13. Rastreamento dos requisitos

O objetivo primordial de todo desenvolvimento de software deve ser o atendimento as necessidades dos usuários e *stakeholders*. Para tanto há necessidade de uma estratégia de rastreamento dos requisitos que permita a verificação se uma determinada característica foi implementada, desde a fonte de solicitação ao produto final. Cada projeto deve decidir quais artefatos deverão ser rastreados e de que maneira (LEFFINGWELL, 2003). A Figura 20 apresenta os artefatos com o registro dos requisitos e a ligação de rastreamento existente entre eles. Nesta figura percebe-se que as necessidades dos usuários podem concluir em características do produto, que por sua vez ou geram “Especificação suplementar” ou “use-case”. Os três retângulos dos use-cases representam o rastreamento dos use-cases de uma iteração para outra (concepção, elaboração 1 e elaboração 2). Para cada relação de rastreamento foi elaborada uma matriz de rastreamento, permitindo a equipe de desenvolvimento um melhor mecanismo de validação dos requisitos.

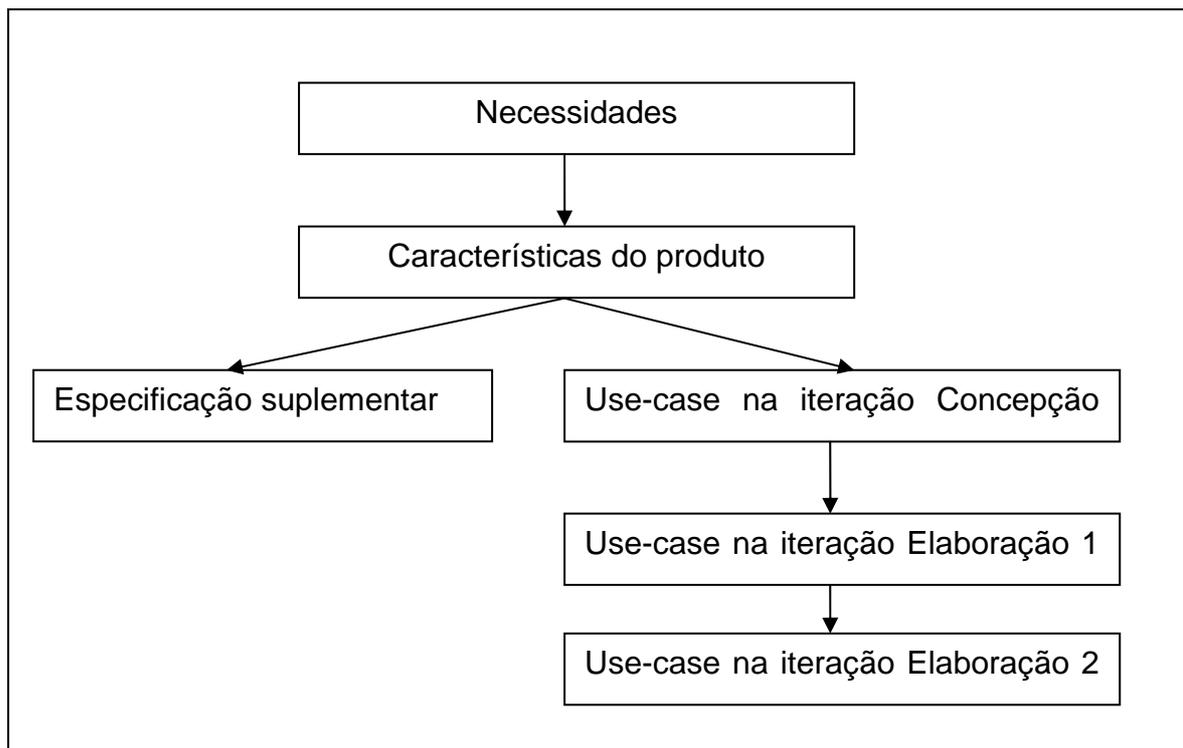


Figura 20: Estrutura de rastreamento

A Figura 21 apresenta uma captura de tela da ferramenta utilizada para o rastreamento dos artefatos mostrando o rastreamento entre use-cases de duas iterações. Na horizontal encontram-se os use-cases da iteração Concepção, e na vertical os use-cases da iteração Elaboração 1. A seta no cruzamento de uma linha com determinada coluna representa uma relação de rastreamento entre os requisitos. Na Figura 21 percebe-se uma seta marcada com uma linha diagonal vermelha no cruzamento entre o use-case “Inscreve-se” da iteração Concepção com o mesmo use-case na fase de Elaboração 1. Este seta representa um relacionamento suspeito, ou seja, a introdução de uma modificação pode ter causado inconsistência no use-case dependente. Neste caso há necessidade de revisão por parte da equipe de desenvolvimento de todos os artefatos dependentes propagando as modificações onde for necessário.

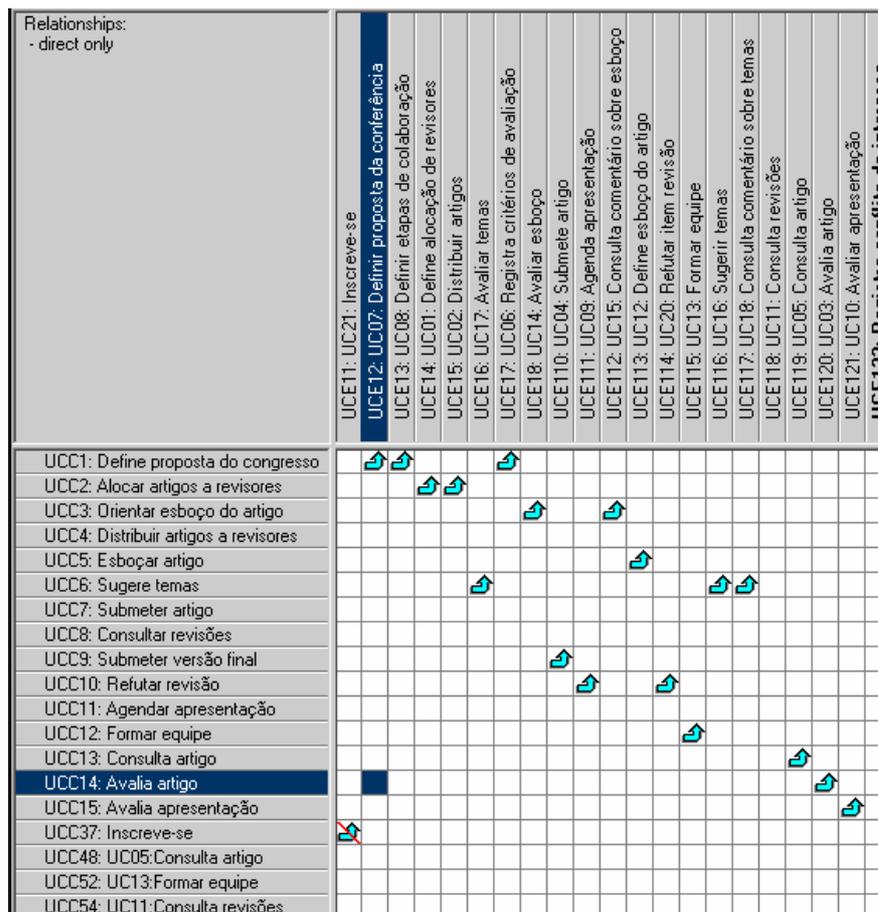


Figura 21: Rastreamento dos requisitos

5.14. Requisitos não funcionais

Os requisitos não funcionais desempenham um papel muito importante na definição de um sistema (WIEGERS, 2003). Na aplicação do processo os requisitos não funcionais identificados foram registrados de forma textual no Documento Suplementar. Foram classificados em quatro categorias: usabilidade, confiabilidade, performance e suporte (LEFFINGWELL, 2003). **O Erro! A origem da referência não foi encontrada.** lista os requisitos não funcionais encontrados.

Quadro 19: Principais requisitos não funcionais

Classificação	Requisito	Descrição
Usabilidade	Aparência	O software deve oferecer um leiaute agradável ao usuário.
	Acesso remoto	Deve haver meios para acesso remoto.
Confiabilidade	Erros de transmissão	Deve haver menos de 5% de erros na submissão dos arquivos à plataforma.
Performance	Tempo de resposta	As operações de resposta de questionário serão as mais freqüentemente utilizadas. Devem ser carregadas em menos de 20 segundos.
	Capacidade	Os recursos de hardware e comunicação devem ser inicialmente dimensionados para atender 120 usuários iniciais (aproximadamente 3 turmas simultâneas).
Suporte	Multilingüe	Necessidade da interface do software em múltiplas línguas permitindo a quebra de fronteiras geográficas ou institucionais.
	Extensibilidade	Permitir a extensão do software

5.15. Glossário

O glossário registra e esclarece os principais termos e conceitos do domínio do problema, reduzindo os problemas na comunicação entre toda a equipe envolvida no projeto (KROLL, 2003). O glossário também auxilia o detalhamento dos use-

cases (ROSENBERG, 2001). Sua criação coincide com o início do projeto, no início da fase de elaboração, sendo constantemente atualizado para refletir novos os termos e conceitos descobertos.

Na aplicação do processo a maioria dos termos foi identificada durante a fase de elaboração e registrada de forma textual em um artefato separado, conforme apresentado no Apêndice C. O glossário utilizado na aplicação do processo não objetivava registrar todos os termos, mas apenas os não claros, ambíguos ou chaves para os use-cases. O Quadro 20 apresenta alguns dos termos definidos no glossário do projeto.

Quadro 20: Amostra de termos registrados no glossário

Termo	Definição
Par	Pessoa que está no mesmo nível hierárquico que outros participantes de um mesmo processo
Esboço do artigo	Delineação inicial do artigo que se pretende escrever
Avaliação recíproca	Avaliação na qual duas pessoas avaliam uma à outra

5.16. Artefatos complementares utilizados

5.16.1. Modelo entidade-relacionamento

Os diagramas entidade-relacionamento (ER) representam o modelo de dados e suas relações (Wieggers, 2003). O modelo ER é amplamente utilizado na fase de design para a representação dos dados a serem armazenados no banco de dados, mas segundo Leffingwell (2003) também representa uma ferramenta adequada na representação e validação de requisitos junto a usuários.

Kulak (2003) afirma que *stakeholders* com perfil técnico tendem a prestar muito mais atenção em um diagrama ER ou em um diagrama de classes do que nos requisitos. Por este motivo na aplicação do processo o diagrama ER foi utilizado somente na segunda iteração da elaboração, visando não desviar a atenção dos *stakeholders*.

Na aplicação do processo foi utilizado o diagrama ER como um artefato complementar para a representação dos dados e suas relações, bem como para auxiliar na validação dos requisitos. As figuras 22 e 23 apresentam o diagrama ER construído para suportar o armazenamento de informações relativas às operações definidas nos use-cases. O diagrama está separado em duas figuras para facilitar a legibilidade.

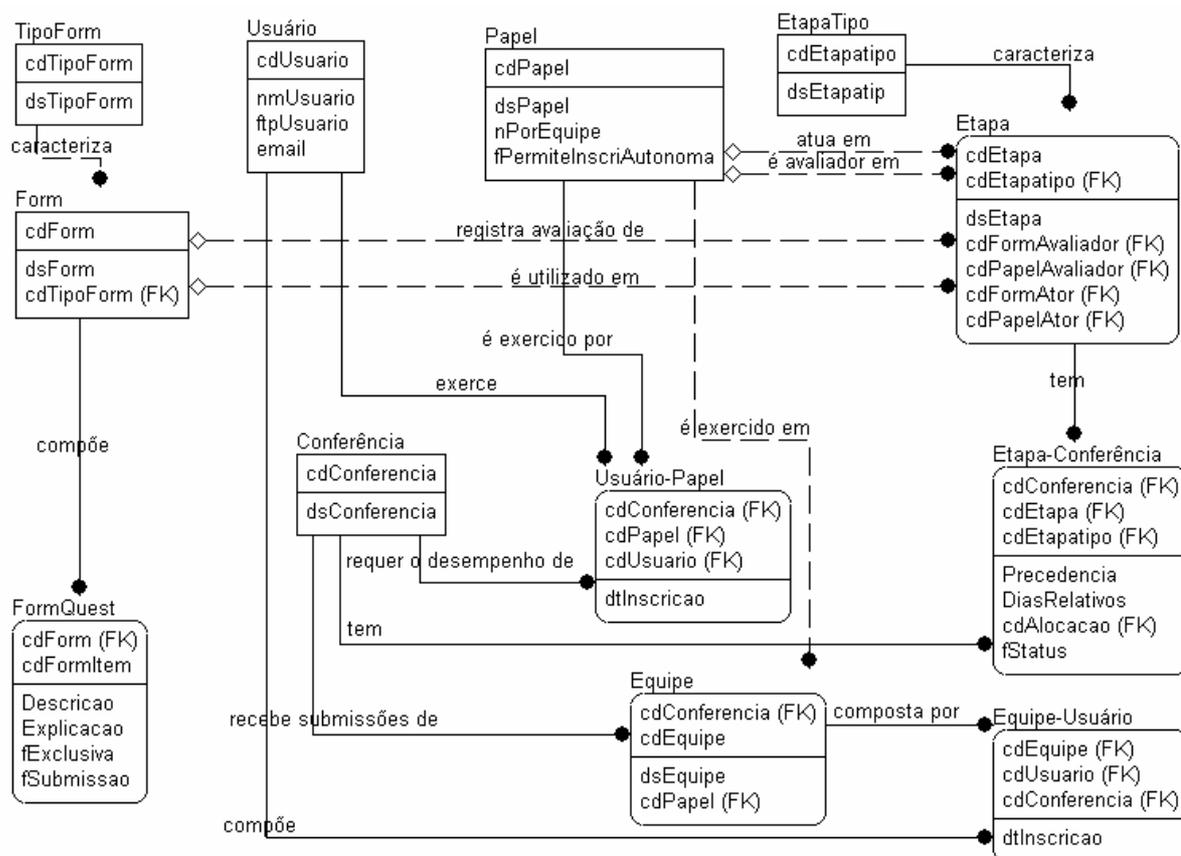


Figura 22: Modelo ER

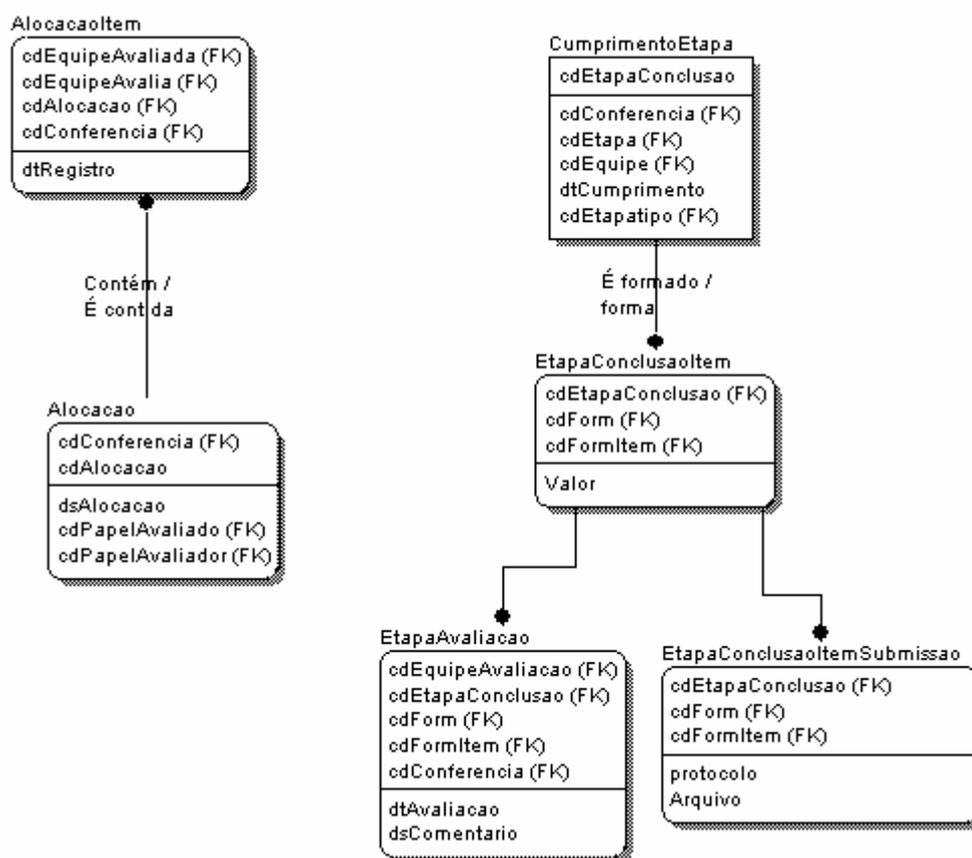


Figura 23: Modelo ER

As Figuras 22 e 23 representam as principais entidades identificadas para suportar os use-cases no formato detalhado da segunda iteração da fase de elaboração. Embora seja um diagrama parcial, alguns atributos das entidades foram documentados para facilitar a validação junto aos usuários. Destacam-se as seguintes entidades:

a) Usuário: representa qualquer participante na simulação do congresso, independente de estar inscrito em alguma conferência e do papel a exercer dentro da conferência.

b) Papel: classifica um participante conforme a sua função numa conferência, por exemplo: editor, autor, revisor.

c) Etapa: registra as etapas. Cada etapa possui um formulário de avaliação e o formulário avaliado, assim como o papel que será avaliado e o papel do avaliador.

d) Formulário e Formulárioitem: estrutura genérica que permite o registro de formulários de avaliação ou submissão de artigo. Os formulários são posteriormente

associados a etapas de conferência, de forma que cada organizador de conferência seleciona as etapas e os formulários de avaliação desejados.

e) EtapaConferência: permite ao organizador selecionar as etapas que farão parte da simulação da conferência. Desta forma cada conferência poderá utilizar apenas as etapas de interesse.

f) Equipe: permite que a cada conferência seja formada equipes, sejam para revisores ou autores. Desta forma também se pode realizar a revisão em grupo.

g) Alocação e AlocaçãoItem: entidade que registra o relacionamento entre as equipes, informando qual equipe avalia e qual equipe será avaliada. Isto garante ao orientador flexibilidade para criar diversas formas de alocação entre as etapas de uma mesma conferência.

5.16.2. Protótipo

Embora o protótipo seja uma ferramenta utilizada para a minimização dos riscos do desenvolvimento, seu uso pode introduzir novos riscos (WIEGERS, 2003). Muitos usuários se deparam com o protótipo e imaginam que o software já está quase pronto, quando na verdade o protótipo possui função de reduzir alguns riscos do projeto. Na aplicação do processo foi desenvolvido um protótipo descartável (WIEGERS, 2003) visando auxiliar na validação de alguns conceitos e conseqüente minimização do risco. A Figura 24 apresenta o protótipo desenvolvido para a aplicação do processo, que foi desenvolvido apenas para validação dos conceitos e será descartado posteriormente.

The screenshot shows a software application window titled "Revisão pelos pares na aprendizagem". The menu bar includes "Definição da proposta", "Preparatório", "Conferência", "Relatórios", "Janelas", and "Ajuda". Below the menu, there are fields for "Usuário:" and "Conferência:". The main area contains a "Formulário" window with two data entry sections, each with a toolbar and a scroll bar.

Formulário 1

Código	1
Descrição	Seleção de tema
Tipo	Avaliação

Formulário 2

Código item	1
Descrição item	Qual a área de interesse de pesquisa?
Explicação	Determinar o interesse da pesquisa: banco de dados, engenharia de requisitos, orientação a objetos

Figura 24: Protótipo descartável para validação de conceitos

6. DISCUSSÃO

A principal contribuição deste trabalho é a identificação, análise e registro dos principais requisitos de software para uma plataforma que suporte a revisão pelos pares na aprendizagem. O documento visão, o glossário e os diagramas de use-cases resultantes do processo de análise de requisitos constituem o principal legado deste projeto de pesquisa para futuros desenvolvimentos.

Durante o processo de análise dos requisitos primou-se pelo registro dos requisitos sem a limitação de tecnologias específicas. Esta independência tecnológica dos use-cases permite liberdade na escolha da linguagem e tecnologia a ser utilizada na implementação. Sendo assim, os artefatos resultantes do processo de análise podem ser utilizados com a tecnologia de desenvolvimento que melhor couber.

Com base na análise efetuada no capítulo 5 pôde identificar-se, que a implementação dos requisitos relativos ao trâmite (submissão e distribuição) de documentos entre os participantes representam as atividades nas quais incide a maior parte do trabalho administrativo. Este trâmite deve acontecer conforme uma tabela de alocação definida pelo organizador, outra tarefa administrativa dispendiosa. Recomenda-se o desenvolvimento destas funcionalidades nas primeiras iterações pois representam as funções mais importantes do software e formam a estrutura básica sobre a qual as outras características serão agregadas.

Além de servirem para uma futura implementação, os artefatos resultantes do processo também representam uma fonte de pesquisa para a comunidade com interesse na aplicação da revisão pelos pares na aprendizagem. Esta contribuição se deve à característica dos use-cases de possuírem fácil compreensão, mesmo para pessoas sem conhecimento técnico. Em virtude disto, esta dissertação também coopera com a disseminação da abordagem e a exploração de técnicas correlatas, como a aprendizagem colaborativa, a avaliação por pares, bem como o tratamento do acervo de informações gerados no processo de revisão pelos pares, insumo para a avaliação pelo docente e para a consolidação da prática do feedback sistemático.

A utilização das melhores práticas na engenharia de requisitos em conjunto com um processo de desenvolvimento são fatores recomendados para o sucesso

dos projetos de software. Embora exista clara necessidade na qualificação das empresas de desenvolvimento de software, muitas desconhecem ou não utilizam processos de engenharia de requisitos. Há necessidade da disseminação das melhores práticas de engenharia de requisitos objetivando a redução na distância entre a pesquisa acadêmica e sua aplicação na cadeia produtiva. Neste contexto, esta dissertação também contribuiu para a propagação da importância do efetivo gerenciamento dos requisitos nos projetos de desenvolvimento de software.

Para que alguns requisitos pudessem ser validados foi desenvolvido um protótipo de um software que suporta algumas funcionalidades do processo de revisão pelos pares. Ele pode ser adaptado para validar novos conceitos ou requisitos que venham a surgir. Neste sentido o protótipo torna-se um modelo auxiliar para a equipe de desenvolvimento na validação de novos requisitos.

A utilização da revisão pelos pares na aprendizagem é uma abordagem pedagógica promissora, pouco explorada no cenário da educação brasileira. Para facilitar sua exploração foram identificados e registrados requisitos que facilitem a disseminação da abordagem. O acompanhamento de observadores externos é uma característica de reduzida importância operacional, mas de suma importância no contexto de divulgação da abordagem. Os mecanismos para simplificar a criação de novas conferências por meio da reutilização da estrutura (questionários e etapas) de conferências passadas, são outro exemplo que visam facilitar a disseminação da abordagem.

A desoneração da carga burocrática é um objetivo primário dos futuros sistemas de apoio à revisão pelos pares na aprendizagem. Pode-se vislumbrar, no entanto, que o uso de técnicas avançadas na construção de sistemas pode criar facilidades importantes, ou mesmo novos requisitos. A aplicação de técnicas avançadas na interpretação dos artigos submetidos por meio da marcação semântico-retórica de texto, detecção de fraude, verificação de formatos e de *choppy writing* permitirá a identificação e exploração de novos requisitos relacionados a engenharia do conhecimento. A implementação destes requisitos em uma plataforma de software facilitará a utilização da abordagem no ensino a distância (EAD).

7. CONCLUSÃO

7.1. Sumário da dissertação

Neste trabalho de pesquisa foi customizado e aplicado o processo de desenvolvimento UP para a identificação, análise e registro dos principais requisitos de um software que suporte a aplicação da revisão pelos pares na aprendizagem. A utilização do processo UP em conjunto com as melhores práticas na engenharia de requisitos facilitou o registro e detalhamento das necessidades indispensáveis sem desprezar as necessidades acessórias que devem ser abordadas e mais detalhadas em iterações futuras como determina o desenvolvimento iterativo e incremental.

Para alcançar o objetivo geral descrito acima foi necessária a revisão da literatura sobre a revisão pelos pares e sua aplicação na aprendizagem como mecanismo capaz de potencializar habilidades não técnicas, tão necessárias aos egressos dos cursos de graduação. Foi apresentada uma adaptação da revisão pelos pares aplicada na aprendizagem conforme proposta de Kern et al. (2002). A burocracia envolvida na aplicação do processo foi exposta como o principal fator desestimulador de sua aplicação, sendo indispensável um ambiente computacional que suporte a prática da revisão pelos pares na aprendizagem.

Foi apresentada a necessidade de profissionalização da atividade de desenvolvimento de software através da utilização de processos que possam ser repetidos e possuam controle de qualidade apurado. Neste contexto, foram destacados que os principais problemas enfrentados pelo desenvolvimento de software estão relacionados aos requisitos. As práticas necessárias a equipe de desenvolvimento para a minimização dos erros no processo de análise de requisitos foram apresentadas como alternativa para redução nos custos de desenvolvimento e atendimento as reais necessidades dos usuários.

O processo de desenvolvimento UP foi customizado no capítulo 4 para adaptá-lo às necessidades específicas do projeto. Foram definidos os artefatos, fases e fluxo de trabalho que foram utilizados no processo de análise de requisitos deste projeto de pesquisa. Além disso, as melhores práticas definidas por Leffingwell (2003) foram exibidas no contexto do processo de desenvolvimento UP.

O processo de desenvolvimento UP customizado foi aplicado no contexto da revisão pelos pares na aprendizagem com objetivo de elicitar, analisar e documentar os principais requisitos necessários. Foram analisadas diferentes fontes de informação visando à coleta do maior número de características possíveis. Com base nestas informações elaborou-se o documento visão contendo a definição e os limites da solução que estava sendo proposta. No documento visão também foram sintetizadas as principais características que a plataforma deverá possuir.

Os use-cases foram inicialmente escritos no formato resumido visando suportar as necessidades identificadas, sem se preocupar com detalhes. Com o decorrer do processo de desenvolvimento os use-cases foram sendo detalhados conforme a prioridade de cada use-case para o processo. Ao final da fase de elaboração cerca de noventa e cinco por cento dos use-cases se encontram no formato detalhado, sendo que o restante dos use-cases será detalhado durante a fase de construção.

As habilidades para um efetivo gerenciamento dos requisitos foram assimiladas neste trabalho. Além disso, o processo de desenvolvimento iterativo e incremental UP foi aplicado para suportar o processo de engenharia de requisitos. Ao final da aplicação do processo diversos artefatos foram gerados registrando os requisitos necessários a um software que suporte a prática da revisão pelos pares na aprendizagem.

Devido à ausência de docentes utilizando a abordagem de revisão pelos pares na educação houve um número reduzido de *stakeholders* disponíveis para o levantamento dos requisitos. Dentro destas circunstâncias, verificou-se que a análise de requisitos foi realizada de forma rápida e simples, sem a existência de requisitos conflitantes. O resultado foi um conjunto conciso de requisitos com as principais funcionalidades, representando uma solução que atende as expectativas dos interessados. A solução proposta representa uma possível solução, provavelmente existam outras e talvez até melhores soluções.

7.2. Trabalhos futuros

Embora o presente trabalho tenha alcançado os objetivos propostos, a partir dos resultados obtidos surgem diversos temas que podem ser explorados. A

primeira proposta de trabalho futuro consiste na implementação dos requisitos especificados neste projeto de pesquisa. Tornar simples e fácil a aplicação do processo de revisão pelos pares na aprendizagem exige o desenvolvimento de um software que suporte esta prática. A implementação dos requisitos especificados neste projeto de pesquisa tornará possível o desenvolvimento de um software que reduza a excesso de trabalho administrativo, facilitando a aplicação da revisão pelos pares em sala de aula.

Através da popularização da revisão pelos pares aplicadas na aprendizagem existirão mais pessoas envolvidas na prática da abordagem. Novos requisitos irão surgir por meio do feedback fornecido. Neste contexto surge a necessidade da criação de uma comunidade com objetivo de evoluir a utilização da abordagem através da sugestão de novos requisitos.

Com a utilização de recursos computacionais adequados, a revisão pelos pares na aprendizagem possui proveito no ensino a distância (EAD), pois permite aplicação com participantes geograficamente distribuídos de forma assíncrona. Sendo assim, recomenda-se explorar a integração da plataforma de revisão pelos pares na aprendizagem com portais de ensino a distância.

Durante o processo de análise dos requisitos foram elicitadas e registradas diversas necessidades dos usuários que não se transformaram em requisitos de software pois foram julgadas como características acessórias ou avançadas, ficando fora do escopo inicial do projeto. Diversas características são merecedoras de investigação futura, permitindo a agregação de novas características que irão auxiliar ainda mais a aplicação da revisão pelos pares na aprendizagem. Dentre as quais pode-se citar:

- a utilização de especificações sintático-semânticas (ontologias) para a representação e intercâmbio de documentos produzidos por estudantes no contexto da revisão pelos pares (e.g. artigos, teses, resenhas);
- verificação automática de referências;
- aplicação de técnicas de inteligência artificial e correlatas, principalmente para a solução do problema de alocação de revisores pares;

- automática verificação metodológica (sintática) e de estilo;
- mecanismo que suporte comentários de revisão no do próprio artigo;
- submissão e manipulação de artigos em XML;

A aplicação da revisão pelos pares na aprendizagem é especialmente promissora no desenvolvimento de habilidades e competências não técnicas. Outras abordagens podem ser utilizadas com intuito de exploração destas habilidades. Diversas atividades de pesquisa podem ser realizadas no sentido de viabilizar um portal de conhecimento que objetive o desenvolvimento de habilidades não técnicas por meio da utilização de técnicas diversas.

REFERÊNCIAS BIBLIOGRÁFICAS

ARMOUR, F.; MILLER, G. **Advanced Use Case Modeling**. New Jersey: Addison-Wesley, 2001.

Association for Computing Machinery. ACM code of ethics and professional conduct. Adotado pelo ACM Council em 16/10/1992. Disponível em: <<http://www.acm.org/constitution/code.html>>. Acesso em: 27/06/2004.

BARRETO, Aldo de Albuquerque. A oferta e a demanda de informação: condições técnicas, econômicas e políticas. **Ciência da Informação** v. 28, n. 2, p. 167, 1999

BUDNY, D. et al. Using writing to meet the needs of freshmen. In: INTL. CONF. ON ENGINEERING AND TECHNOLOGY EDUCATION – INTERTECH, 7., 2002, Santos-SP. **Proceedings...** (CD-ROM), 2002. p. 1-6.

CASTELLS, M. **A sociedade em rede**. São Paulo: Paz e Terra, 1999.

COCKBURN, A. **Writing Effective Use Cases**. New York: Addison-Wesley, 2001.

CORREA NETO, Á. S.; KERN, V. M. Requisitos de uma interface para apoio à aplicação da revisão pelos pares na aprendizagem. In: **CONFERÊNCIA INTERNACIONAL DE EDUCAÇÃO EM ENGENHARIA E COMPUTAÇÃO - ICECE**, 2003, São Vicente / Santos-SP. Anais (CD-ROM), 2003. p. 1-5.

FOWLER, M.; **UML distilled : a brief guide to the Standard object modeling language**. 3. ed. Boston: Pearson Education, 2004.

HORROBIN, D. F. Peer review of grant applications: a harbinger for mediocrity in clinical research? **The Lancet**, v. 348, p. 1293-1295, 1996.

HORROBIN, D. F. Referees and research administrators: barriers to scientific research? **British Medical Journal**, v. 2, p. 216, 1974.

HORROBIN, D. F. Something rotten at the core of science? **Trends in Pharmacological Sciences**, v. 22, n. 2, p. 51-52, February 2001.

HUGHES, J. M. et al. Directions and challenges in informatics education. In: CASSEL, L.; REIS, R. A. L. (Org.). **Informatics curricula and teaching methods**. Boston: Kluwer Academic Publishers, 2003, v. 245, p. 115-123.

KERN, V. M., PERNIGOTTI, J. M., CALEGARO M. M, BENTO M. Peer review in engineering education: speeding up learning, looking for a paradigm shift. In: INTL. CONF. ON ENGINEERING AND TECHNOLOGY EDUCATION – INTERTECH, 7., 2002, Santos-SP. **Proceedings...** (CD-ROM), 2002. p. 1-5. Disponível em: <http://kern.stela.ufsc.br/publ/Ittech02_peerreview.pdf>. Acesso em: 27/06/2004.

KERN, V. M., SARAIVA, L. M., PACHECO, R. C. S. Peer review in education: promoting collaboration, written expression, critical thinking, and professional responsibility. **Education and Information Technologies**, v. 8, n. 1, p. 37-46, 2003. Disponível em: <<http://www.kluweronline.com/issn/1360-2357>>. Acesso em: 27/06/2004.

KERN, V. M.; SARAIVA, L. M. Aplicação da revisão pelos pares no ensino de graduação. **Alcance**, Itajaí, ano VI, n. 3, p. 42-49, nov. 1999. Disponível em: <<http://kern.stela.ufsc.br/publ/AlcancePeer.pdf>>. Acesso em: 27/06/2004.

KOTONYA, G. SOMERVILLE, I.; **Requirements Engineering: processes and techniques**. New Your: John Wiley & Sons Ltd, 1998.

KROLL, P.; KRUCHTEN, P. **Rational Unified Process Made Easy: A Practitioner's Guide to the RUP**. Boston: Addison Wesley, 2003.

KRUCHTEN, P.; **Rational Unified Process, The: An Introduction**. 3. ED. Boston: Pearson Education, 2003.

KULAK, D.; GUINEY, E. **Use Cases: Requirements in Context**. 2. ed. Boston: Addison Wesley, 2003.

LARMAN, C. **Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design**. 2. ed. Upper Saddle River, NJ: Prentice-Hall, 2003.

LEFFINGWELL D.; WIDRIG D. **Managing Software Requirements: A Use Case**

Approach. 2. ed. Boston: Addison Wesley, 2003.

MORAN, G. Review. **Library and Information Science Research**, v. 17, n. 1, p. 90-92, 1995. Resenha de: DANIEL, H.-D. Guardians of science: fairness and reliability of peer review. VCH Verlagsgesellschaft, 1993, 118 p.

MOREIRA, D. A. e SILVA, E. Q. A method to increase student interaction using student groups and peer review over the internet. **Education and Information Technologies**, v. 8, n. 1, p. 47-54, 2003.

MORIN, E.; **Para sair do século XX**. Rio e Janeiro: Nova Fronteira, 1995.

MULDER, F.; VAN WEERT, T.; (Editors). **IFIP/UNESCO Informatics Curriculum Framework 2000, Building effective higher education informatics curricula in a situation of change**. UNESCO: Paris, 2000. Disponível em: <<http://poe.netlab.csc.villanova.edu/ifip32/ICF2000.htm>>. Acesso em: 27/06/2004.

OESTEREICH, B.; **Developing Software with UML**. Oldenbourg: Addison-Wesley, 2002.

PADUAN, R.; Tesouro escondido. **Exame**. Local de publicação, volume, número, página inicial-final do artigo, Junho de 2003(EDIÇÃO 795).

PESSANHA, C. Critérios editoriais de avaliação científica: notas para discussão. **Ciência da Informação**, pp. 226-229, Maio/Agosto, 1998.

PRESSMAN, Roger. S. **Software Engineering: A practitioner's approach**. New York: Ed. McGraw-Hill, 1997.

PIAGET, J. **Educar para o futuro**. Rio de Janeiro: Fundação Getúlio Vargas, 1974.

ROBERTSON, J.; ROBERTSON, S. **Mastering the requirements process**. Addison-Wesley, 1999.

ROOYEN, S. van. **The evaluation of peer-review quality**. Learned Publishing, v. 14, p. 85-91, 2001.

ROSENBERG, D. SCOTT K. **Applying Use Case Driven Object Modeling with**

UML: An Annotated e-Commerce Example. New Jersey: Addison Wesley, 2001.

ROWLAND, F. The peer-review process. **Learned Publishing**, v. 15, p. 247–258, 2002.

SANTOMÉ, J.T. Globalização e interdisciplinaridade. **Artmed**, 1999.

SARAIVA, L. M. **Proposta metodológica de aplicação da revisão pelos pares como instrumento pedagógico para a educação ambiental.** 2002. 103f. Tese (Doutorado em Engenharia de Produção). Universidade Federal de Santa Catarina, Florianópolis, 2002. Disponível em: <http://teses.eps.ufsc.br/Resumo.asp?3931>. Acessado em: 01/05/2004.

SMITH, A. J.; **The task of the referee.** **IEEE Computer**, v. 23, n. 4, p. 46-51, 1990. Disponível em: <http://www.computer.org/tkde/taskoftheferee.pdf>. Acesso em: 27/06/2004.

SNODGRASS, R. **Summary of conference management software.** January, 1999. Disponível em: <http://www.acm.org/sigs/sgb/summary.html>. Acessado em: 27/06/2004.

SOMMERVILLE, Ian. **Software Engineering.** Lancaster University, Lancaster: Ed. Addison-Wesley, 1995.

WIEGERS, E. Karl. **Software Requirements.** 2. ed. Washington: Microsoft Press, 2003.

WINDLE, D. R.; ABREO, L. R. **Software Requirements Using the Unified Process: A Practical Approach.** New Jersey: Prentice Hall PTR, 2002.

YOUNG, R. R.; **The Requirements Engineering Handbook.** Boston: Artech House, 2004.

APÊNDICE

Apêndice A - Documento visão

1. Introdução

O objetivo deste documento é coletar, analisar e definir um conjunto de necessidades de alto nível de um ambiente de revisão pelos pares na aprendizagem. Está focado nas necessidades dos *stakeholders*, e usuários, e por que elas existem. Os detalhes para suportar as necessidades identificadas estão detalhados nos use-case e nas especificações suplementares.

1.1 Propósito

O propósito deste documento é apresentar uma descrição alto-nível do desenvolvimento que se pretende iniciar.

1.2 Visão geral

Explicar as seções seguintes e como o documento visão está organizado.

2. Posicionamento

2.1 Oportunidade de Negócio

A revisão pelos pares na aprendizagem é uma abordagem pedagógica de resultados relevantes no desenvolvimento da prática na expressão escrita e falada, senso de responsabilidade profissional e pensamento crítico. O excesso de burocracia inviabiliza a aplicação da abordagem em sala de aula. Uma ferramenta de revisão pelos pares na aprendizagem é de suma importância.

2.2 Afirmação do Problema

O problema	o excesso de burocracia na aplicação da revisão pelos pares na aprendizagem
Afeta	Alunos, professores, comunidade científica em geral e a qualidade da formação acadêmica.
O impacto	a não utilização da abordagem na aprendizagem
Uma solução de sucesso	Permitir a simulação de um congresso, onde professores e alunos possuam um ambiente que suporte a

	revisão pelos pares, facilitando minimizando o trabalho administrativo envolvido e facilitando a colaboração entre os participantes.
--	--

2.3 Posicionamento do Produto

Para	Cursos de graduação e pós-graduação
Que	Gostariam de utilizar a revisão pelos pares
O	RVPA (Revisão pelos pares na aprendizagem)
Que	cria um ambiente de revisão pelos
Ao contrário	dos softwares de gestão de conferência (Cyberchair,etc...)
Nosso produto	Atende todos os processos envolvidos na aplicação da abordagem

3. Stakeholder e Descrição dos Usuários

Os principais *stakeholders* são as pessoas que já utilizaram e/ou participaram do processo de revisão pelos pares na aprendizagem. Estes *stakeholders* são de grande valia, pois conhecem a abordagem com riqueza de detalhes.

Outra fonte de informação a ser explorada são os organizadores de conferência, devido a seu conhecimento na utilização da revisão pelos pares em congressos.

Educadores, pedagogos e especialistas em educação não possuem participação operacional do ambiente, mas são *stakeholders* que possuem necessidades específicas na avaliação da efetividade do processo como um todo, ou seja, na consulta a informações gerenciais.

3.1 Ambiente do Usuário

A abordagem de revisão pelos pares na aprendizagem é aplicada em turmas de graduação e/ou pós-graduação, sendo que o número de participantes varia conforme o tamanho da sala, ficando algo de 15 a 50 participantes.

Estes usuários irão acessar o sistema dos mais diversos lugares (casa, escritório, laboratórios, cybercafés)

A aplicação da abordagem normalmente dura o tempo de 3 a 6 meses, durante a duração de uma disciplina.

4. Visão Geral do Produto

O produto a ser desenvolvido deve permitir a simulação de um congresso em sala de aula, permitindo a colaboração entre professores e alunos.

4.1 Resumo das Capacidades

Benefício ao Cliente	Funcionalidade relacionada
Simulação de um congresso em sala de aula	Criação de um congresso, com inscrição de participantes.
Redução do trabalho burocrático	Facilitação na alocação e distribuição dos documentos durante o processo
Divulgação da abordagem	
Avaliação da abordagem de revisão pelos pares na educação	Consultas e relatórios sobre evolução da aplicação
Personalizável	Cada simulação de congresso poderá personalizar fases, descrições, etc..
Extensível	Desenvolvimento do software orientado a objetos e utilizando design patterns visando a extensão do ambiente.

5. Características de Alto Nível do Produto

Abaixo um sumário de alto nível com as principais características do produto.

[FEAT1 Simulação de um congresso em sala de aula](#)

[FEAT2 Permitir a avaliação dos participantes do congresso](#)

[FEAT3 Fluxo de trabalho variável, com descrição, objetivo e data limite para cada etapa do fluxo](#)

[FEAT4 Lembrar aos participantes a proximidade do encerramento de determinadas etapas](#)

[FEAT5 Multilíngüe](#)

[FEAT6 Acessível remotamente](#)

[FEAT7 Permitir colaboração entre os participantes, através de publicação e consulta a referências](#)

[FEAT26 Trâmite automático de documentos](#)

[FEAT9 Inscrição de participantes \(autores, revisores, editores, observadores\)](#)

[FEAT10 Alocação automática de revisores](#)

[FEAT11 Alocação manual de revisores](#)

[FEAT12 Permitir a expansão e personalização do ambiente](#)

[FEAT13 Tornar acessível o material resultante \(caderno técnico\)](#)

[FEAT14 Escalonamento das apresentações](#)

[FEAT15 Permitir aos revisores avaliar a apresentação dos artigos](#)

[FEAT27 Sumarização das revisões](#)

[FEAT17 Suportar esboço do artigo e avaliação do esboço](#)

[FEAT18 Facilitar a divulgação da abordagem](#)

[FEAT19 Permitir aos participantes avaliar a aplicação o processo de revisão](#)

[FEAT20 Pedagogos e interessados podem consultar dados gerais sobre a utilização da plataforma.](#)

[FEAT21 Permitir ao organizador definir um arquivo modelo de formatação que será utilizado como base para a redação dos artigos](#)

[FEAT22 O organizador poderá definir os tipos de arquivos aceito pela conferência](#)

[FEAT23 Permitir ao editor a consulta a artigos que possuem divergências na avaliação](#)

[FEAT24 O organizador poderá escolher o tipo de revisão: cega ou duplamente cega](#)

[FEAT25 Permitir ao Editor definir os critérios de avaliação a serem utilizados na conferência](#)

Apêndice B – Use-cases

[UCE21 UC23: Cópia estrutura de conferência](#)

Breve descrição	Permitir ao Editor realizar a cópia da estrutura de uma conferência anterior.
Ator primário	Editor
Stakeholders e interesses:	Editor: iniciar uma conferência rapidamente através da re-utilização de uma estrutura previamente existente.
Pré-condições	Editor identificado para o sistema.
Fluxo principal	1. O Sistema apresenta ao Editor as conferências que estão compartilhadas. 2. O Editor seleciona a conferência origem que deseja importar. 2. O Editor seleciona a conferência destino e confirma a importação. 3. O Sistema importa a estrutura da origem para o destino.
Fluxo alternativo	1.1 O Sistema não encontra nenhuma estrutura de conferência compartilhada 3.1 O Sistema apresenta um erro pois não pode importar devido à conferência destino já possuir uma estrutura.
Requisitos especiais	

[UCE22 UC07: Definir proposta da conferência](#)

Breve descrição	Permitir a definição de fases, datas e fluxo de trabalho que serão utilizados na conferência.
Ator primário	Editor
<i>Stakeholders</i> e interesses:	Editor: gostaria de definir a proposta da conferência de forma fácil e com flexibilidade para utilizar apenas as fases de seu interesse.
Pré-condições	Editor identificado para o sistema.
Fluxo principal	1. O Editor informa a descrição para conferência; 2. O Editor seleciona se a estrutura da conferência será pública a outros editores 3. O Editor informa o número de dias limites as datas limites (relativas ao início da conferência) para a inscrição de participantes e formação de equipes. 2. O Editor pode definir as etapas de colaboração (UC08: Definir etapas de colaboração);
Fluxo alternativo	
Requisitos especiais	

[UCE23 UC08:Definir etapas de colaboração](#)

Breve descrição	Permitir a definição das etapas a utilizar na conferência.
Ator primário	Editor é o Editor ou outro Use-case
<i>Stakeholders</i> e interesses:	Editor: gostaria de definir as etapas constantes da conferência de forma fácil.
Pré-condições	Editor identificado para o sistema.
Fluxo principal	1. O Editor informa: a. uma descrição para a etapa; b. o objetivo da etapa; c. a fase antecessora; d. a data limite da fase; 2. O Editor confirma e pode incluir mais etapas;
Fluxo alternativo	
Requisitos especiais	

[UCE24 UC02:Distribuir artigos](#)

Breve descrição	Permitir a distribuição dos artigos a revisores com base na tabela de alocação previamente definida.
Ator primário	Editor
<i>Stakeholders</i> e interesses:	Editor: gostaria de distribuir os artigos de forma rápida.
Pré-condições	Editor identificado para o sistema. Tabela de alocação estar definida
Fluxo principal	1. O Editor seleciona a conferência e confirma; 2. O sistema irá informar aos Revisores que os artigos já se encontram disponíveis para revisão;
Fluxo alternativo	
Requisitos especiais	

[UCE25 UC06:Registra critérios de avaliação](#)

Breve descrição	Permitir ao Editor a definição de como serão avaliados os participantes da simulação de conferência.
Ator primário	Editor
<i>Stakeholders</i> e interesses:	Editor: elaborar um mecanismo de avaliação eficiente.
Pré-condições	Editor identificado para o sistema.
Fluxo principal	1. O Editor seleciona a conferência e a fase desejada 2. O Editor incluir os quesitos de avaliação para a fase / conferência selecionada.

Fluxo alternativo	
Requisitos especiais	

[UCE26 UC22:Registra conflito de interesse](#)

Breve descrição	Permitir o registro da existência de algum conflito de interesse entre Revisores e artigos.
Ator primário	Revisor
Stakeholders e interesses:	Autor: não gostaria de ser revisado por algum revisor que possuísse conflito de interesse em seu artigo. Revisor: gostaria de informar algum conflito com relação a determinado artigo.
Pré-condições	Revisor identificado para o sistema. Esboço dos artigos elaborados e aprovados
Fluxo principal	1. O Revisor seleciona a conferência 2. O Revisor localiza um artigo e informa a descrição do conflito existente.
Fluxo alternativo	
Requisitos especiais	

[UCE27 UC04:Submete artigo](#)

Breve descrição	Permitir aos Autores a submissão de artigos.
Ator primário	Autor
Stakeholders e interesses:	Autor: gostaria de submeter seu artigo e garantir que a submissão ocorresse sem problemas.
Pré-condições	Autor identificado para o sistema. Estar habilitado a submissão de artigos.
Fluxo principal	1. O Autor seleciona a conferência para qual está submetendo; 2. O Autor localiza o arquivo no seu computador e confirma o envio; 3. O Sistema irá transferir o arquivo e indicar ao Autor que a transferência foi efetuada com sucesso;
Fluxo alternativo	3.1. Se ocorrer algum erro na transmissão o sistema tenta novamente e envia retorna uma mensagem de erro caso não consiga enviar por 3 vezes.
Requisitos especiais	

[UCE28 UC12:Define esboço do artigo](#)

Breve descrição	Permitir ao Autor a elaboração do esboço de seu artigo.
Ator primário	Autor
Stakeholders e interesses:	Autor: gostaria de informar a pretensão de pesquisa de seu artigo para que o Editor possa avaliar o mérito.
Pré-condições	Autor identificado para o sistema. Estar habilitado a fase de definição de esboço.
Fluxo principal	1. O Autor seleciona a conferência para qual está participando; 2. O Autor preenche o formulário de esboço e confirma;
Fluxo alternativo	
Requisitos especiais	

[UCE29 UC03:Avalia artigo](#)

Breve descrição	Permitir aos Revisores realizar a avaliação dos artigos conforme os quesitos definidos pelo Editor para a conferência.
Ator primário	Revisor
Stakeholders e	Revisor: avaliar o artigo de forma justa.

interesses:	
Pré-condições	Revisor identificado para o sistema. Estar habilitado a etapa de revisão de artigos.
Fluxo principal	1. O Revisor seleciona a conferência 2. O Sistema apresenta os artigos alocados para receber seu parecer. 3. O Revisor seleciona um artigo e preenche os quesitos de avaliação. 4. O Revisor pode confirmar e realizar a avaliação dos outros artigos.
Fluxo alternativo	1.1 Caso a conferência não esteja em etapa de revisão de artigos o sistema apresenta uma mensagem de erro e retorno.
Requisitos especiais	

[UCE216 UC05:Consulta artigo](#)

Breve descrição	Permitir aos Revisores realizar a consulta e download dos arquivos alocados para receber seu parecer.
Ator primário	Revisor
Stakeholders e interesses:	Revisor: consultar e baixar o arquivo para avaliação
Pré-condições	Revisor identificado para o sistema. Estar habilitado a etapa de revisão de artigos.
Fluxo principal	1. O Revisor seleciona a conferência 2. O Sistema apresenta os artigos alocados para receber seu parecer disponíveis para consulta ou download. 3. O Revisor pode selecionar um arquivo para abrir ou download.
Fluxo alternativo	1.1 Caso a conferência não esteja em etapa de revisão de artigos o sistema apresenta uma mensagem de erro e retorno.
Requisitos especiais	

[UCE210 UC21:Inscreve-se](#)

Breve descrição	Permitir aos Usuários realizar a inscrição para a participação da simulação de conferência.
Ator primário	Usuário
Stakeholders e interesses:	Usuário: realizar sua inscrição no congresso. Editor: permitir que os Usuários realizem sua própria inscrição, cabendo ao Editor apenas a confirmação da mesma.
Pré-condições	
Fluxo principal	1. O Usuário registra seus dados 2. O Sistema apresenta as conferências em andamento 3. O Usuário seleciona as conferências desejadas e confirma 2. O Sistema registra os dados e informa ao Usuário que sua inscrição foi submetida à confirmação do Editor.
Fluxo alternativo	1.1 O Usuário já é cadastrado no Sistema: 1.2 Deverá informar o usuário e senha
Requisitos especiais	

[UCE211 UC24: Confirma inscrição](#)

Breve descrição	Permitir ao Editor confirmar a inscrição dos Usuários.
Ator primário	Editor
Stakeholders e interesses:	Editor: confirmar a inscrição dos usuários. Usuário: ter sua inscrição confirmada para iniciar os trabalhos.
Pré-condições	Usuário ter solicitado sua inscrição
Fluxo principal	1. O Editor informa os parâmetros de consulta das inscrições pendentes de aprovação.

	<ol style="list-style-type: none"> 2. O Sistema apresenta as inscrições pendentes. 2. O Editor pode selecionar uma inscrição visualizar detalhes. 3. O Editor aprova a inscrição. 4. O Sistema comunica ao Usuário.
Fluxo alternativo	3.1 O Editor informa um motivo e reprova a inscrição
Requisitos especiais	

[UCE212 UC16:Sugeri temas](#)

Breve descrição	Permitir ao Autor sugerir temas que possui interesse em pesquisar.
Ator primário	Autor
<i>Stakeholders</i> e interesses:	Editor: gostaria que os autores realizassem o artigo sobre temas interessantes e desafiantes. Autor: sugerir temas afins com sua área de pesquisa.
Pré-condições	Usuário estar identificado no sistema. Estar na etapa de sugestão de temas.
Fluxo principal	<ol style="list-style-type: none"> 1. O Autor seleciona a conferência. 2. O Autor descreve os temas que possui interesse em pesquisar e confirma.
Fluxo alternativo	
Requisitos especiais	

[UCE217 UC13:Formar equipe](#)

Breve descrição	Permitir ao Autor formar grupos para a participação das atividades.
Ator primário	Autor
<i>Stakeholders</i> e interesses:	Editor: gostaria de equipes formadas rapidamente. Autor: formar equipes com interesse afinidades.
Pré-condições	Usuário estar identificado no sistema. Estar na etapa de formação de equipes.
Fluxo principal	<ol style="list-style-type: none"> 1. O Autor seleciona a conferência. 2. O Autor registra um nome para a equipe e seleciona os participantes e confirma. 3. O Sistema comunica aos participantes a criação da equipe e seus participantes.
Fluxo alternativo	
Requisitos especiais	

[UCE213 UC20:Refuta item revisão](#)

Breve descrição	Permitir ao Autor discordar de um determinado quesito de avaliação.
Ator primário	Autor
<i>Stakeholders</i> e interesses:	Autor: discordar de um determinado quesito de avaliação.
Pré-condições	Autor estar identificado no sistema. Estar na etapa de apropriação das revisões.
Fluxo principal	<ol style="list-style-type: none"> 1. O Autor seleciona a conferência. 2. O Autor seleciona o item que está discordando, escreve a justificativa e confirma a refutação.
Fluxo alternativo	
Requisitos especiais	

[UCE218 UC11:Consulta revisões](#)

Breve descrição	Permitir ao Autor consultar as revisões que seu artigo recebeu.
Ator primário	Autor
<i>Stakeholders</i> e	Autor: consultar as revisões, para evoluir seu artigo e gerar a versão para submissão final.

interesses:	
Pré-condições	Autor estar identificado no sistema. Estar na etapa de apropriação das revisões.
Fluxo principal	1. O Autor seleciona a conferência. 2. O Sistema apresenta as revisões que o artigo recebeu. 3. O Autor pode selecionar uma determinada revisão e visualizar os detalhes.
Fluxo alternativo	
Requisitos especiais	

[UCE214 UC18:Consulta comentário sobre temas](#)

Breve descrição	Permitir ao Autor consultar os comentários recebidos sobre os temas propostos ao Editor.
Ator primário	Autor
Stakeholders e interesses:	Autor: consultar os temas visando saber a opinião do Editor.
Pré-condições	Autor estar identificado no sistema. Estar na etapa de submissão de temas.
Fluxo principal	1. O Autor seleciona a conferência. 2. O Sistema apresenta os comentários recebidos aos temas propostos.
Fluxo alternativo	
Requisitos especiais	

[UCE215 UC17:Avaliar temas](#)

Breve descrição	Permitir ao Editor a avaliação dos temas sugeridos pelos Autores.
Ator primário	Editor
Stakeholders e interesses:	Editor: garantir que Autores pesquisem sobre temas desafiantes.
Pré-condições	Editor identificado no sistema.
Fluxo principal	1. O Editor seleciona a conferência. 2. O Sistema apresenta as equipes e os temas sugeridos. 3. O Editor pode aprovar um tema e tecer seus comentários. 4. O Sistema comunica aos Autores informando sobre os comentários realizados.
Fluxo alternativo	3.1 O Editor pode reprovar um tema e tecer seus comentários.
Requisitos especiais	

Apêndice C - Glossário

Glossário

1. Introdução

Este Glossário apresenta os termos e conceitos no contexto do projeto.

2. Definições

Abaixo são apresentados os principais termos e definições relativas ao contexto do projeto.

2.1 Conflito de interesses

O conflito de interesses é uma restrição existente entre o revisor e determinado artigo (Questões pessoais ou interesse de pesquisa).

2.2 Estrutura de conferência

Representa a parte estrutural dos dados de uma conferência, compreende os dados da conferência, como: descrição, etapas e datas limites, fichas de avaliação.

2.3 Simulação de congresso/conferência em sala de aula

Reprodução do funcionamento de um congresso em sala de aula visando utilizar o processo de revisão pelos pares nesta simulação.

2.4 Avaliação

Determinação do valor através do julgamento de algum participante.

2.5 Referências

Links, artigos e material de apoio aos participantes.

2.6 Trâmite

Fluxo que um documento deve seguir na simulação de um congresso em sala de aula.

2.7 Alocação de artigos

Associação entre artigos e revisores, definindo responsáveis pela avaliação do artigo.

2.8 Caderno técnico

Versão final de todos os artigos que participaram de um processo de simulação determinado.

2.9 Esboço do artigo

Delineação inicial do artigo.

2.10 Template

Arquivo contendo as diretrizes de formatação e orientações na elaboração do artigo.

2.11 Tipo de arquivo

Formato de documento que pode ser submetido eletronicamente.

2.12 Revisão cega

Revisão onde o revisor conhece a identidade do autor, mas o autor desconhece o nome do revisor.

2.13 Revisão duplamente cega

Revisão onde a identidade de autor e revisor permanece anônima.

2.14 Critérios de avaliação

Quesitos utilizados na avaliação dos artigos.

2.15 Par

Pessoa que possui conhecimento semelhante a outro participante

2.16 Avaliação recíproca

Avaliação onde duas pessoas, avaliam uma à outra.

2.17 Escore

Parecer emitidos pelos participantes para avaliar algo.

2.18 Tema

Proposição que vai ser tratada ou demonstrada.

2.19 Refutar

Negar algum item de avaliação recebido.

2.20 Distribuição

Disponibilizar determinado artigo para ser acessível a outra participante.

2.21 Apresentação de artigo

Manifestação no decurso do qual se apresenta o artigo escrito aos participantes da simulação do congresso.