

TURNO:	NOTURNO	VERSÃO:	1	ANO / SEMESTRE:	2010.2	Nº	
--------	----------------	---------	---	-----------------	--------	----	--

UNIVERSIDADE REGIONAL DE BLUMENAU
CENTRO DE CIÊNCIAS EXATAS E NATURAIS
DEPARTAMENTO DE SISTEMAS E COMPUTAÇÃO
CURSO DE CIÊNCIA DA COMPUTAÇÃO — BACHARELADO
COORDENAÇÃO DE TRABALHO DE CONCLUSÃO DE CURSO

PROPOSTA PARA O TRABALHO DE CONCLUSÃO DE CURSO

TÍTULO: FERRAMENTA PARA AUXÍLIO NA ANÁLISE DE IMPACTO E RASTREABILIDADE DE REQUISITOS NA GESTÃO DE MUDANÇAS

ÁREA: Engenharia de Software

Palavras-chave: Gestão de mudanças. Rastreabilidade. Análise de impacto

1 IDENTIFICAÇÃO

1.1 ALUNO

Nome: José Alberto Zimmermann		Código/matricula: 87381	
Endereço residencial:			
Rua: Luiz Eleodoro da Silva		nº: 485	Complemento:
Bairro: Ponta Aguda	CEP: 89050-501	Cidade: Blumenau	UF: SC
Telefone fixo: 3322-9892		Celular: 9912-5548	
Endereço comercial:			
Empresa: Dynamix Software			
Rua: República Argentina		nº: 704	Bairro: Ponta Aguda
CEP: 89050-100	Cidade: Blumenau	UF: SC	Telefone: 3037-7006
E-Mail FURB:		E-Mail alternativo: jazimmermann@gmail.com	

1.2 ORIENTADOR

Nome: Marcel Hugo	
E-Mail FURB: marcel@furb.br	E-Mail alternativo:

2 DECLARAÇÕES

2.1 DECLARAÇÃO DO ALUNO

Declaro que estou ciente do Regulamento do Trabalho de Conclusão de Curso de Ciência da Computação e que a proposta em anexo, a qual concordo, foi por mim rubricada em todas as páginas. Ainda me comprometo pela obtenção de quaisquer recursos necessários para o desenvolvimento do trabalho, caso esses recursos não sejam disponibilizados pela Universidade Regional de Blumenau (FURB).

Assinatura: _____ Local/data: _____

2.2 DECLARAÇÃO DO ORIENTADOR

Declaro que estou ciente do Regulamento do Trabalho de Conclusão do Curso de Ciência da Computação e que a proposta em anexo, a qual concordo, foi por mim rubricada em todas as páginas. Ainda me comprometo a orientar o aluno da melhor forma possível de acordo com o plano de trabalho explícito nessa proposta.

Assinatura: _____ Local/data: _____

3 AVALIAÇÃO DA PROPOSTA

3.1 AVALIAÇÃO DO(A) ORIENTADOR(A)

Acadêmico(a): José Alberto Zimmermann

Orientador(a): Marcel Hugo

ASPECTOS AVALIADOS		atende	atende parcialmente	não atende
ASPECTOS TÉCNICOS	1. INTRODUÇÃO 1.1. O tema de pesquisa está devidamente contextualizado/delimitado?			
	1.2. O problema está claramente formulado?			
	2. OBJETIVOS 2.1. O objetivo geral está claramente definido e é passível de ser alcançado?			
	2.2. São apresentados objetivos específicos (opcionais) coerentes com o objetivo geral? Caso não sejam apresentados objetivos específicos, deixe esse item em branco.			
	3. RELEVÂNCIA 3.1. A proposta apresenta um grau de relevância em computação que justifique o desenvolvimento do TCC?			
	4. METODOLOGIA 4.1. Foram relacionadas todas as etapas necessárias para o desenvolvimento do TCC?			
	4.2. Os métodos e recursos estão devidamente descritos e são compatíveis com a metodologia proposta?			
	4.3. A proposta apresenta um cronograma físico (período de realização das etapas) de maneira a permitir a execução do TCC no prazo disponível?			
	5. REVISÃO BIBLIOGRÁFICA 5.1. As informações apresentadas são suficientes e têm relação com o tema do TCC?			
	5.2. São apresentados trabalhos correlatos, bem como comentadas as principais características dos mesmos?			
	6. REQUISITOS DO SISTEMA A SER DESENVOLVIDO 6.1. Os requisitos funcionais e não funcionais do sistema a ser desenvolvido foram claramente descritos?			
	7. CONSIDERAÇÕES FINAIS 7.1. As considerações finais relacionam os assuntos apresentados na revisão bibliográfica com a realização do TCC?			
ASPECTOS METODOLÓGICOS	8. REFERÊNCIAS BIBLIOGRÁFICAS 8.1. As referências bibliográficas obedecem às normas da ABNT?			
	8.2. As referências bibliográficas contemplam adequadamente os assuntos abordados na proposta (são usadas obras atualizadas e/ou as mais importantes da área)?			
	9. CITAÇÕES 9.1. As citações obedecem às normas da ABNT?			
	9.2. As informações retiradas de outros autores estão devidamente citadas?			
	10. AVALIAÇÃO GERAL (organização e apresentação gráfica, linguagem usada) 10.1. O texto obedece ao formato estabelecido?			
	10.2. A exposição do assunto é ordenada (as idéias estão bem encadeadas e a linguagem utilizada é clara)?			
<p>A proposta de TCC deverá ser revisada, isto é, necessita de complementação, se:</p> <ul style="list-style-type: none"> qualquer um dos itens tiver resposta NÃO ATENDE; pelo menos 4 (quatro) itens dos ASPECTOS TÉCNICOS tiverem resposta ATENDE PARCIALMENTE; ou pelo menos 4 (quatro) itens dos ASPECTOS METODOLÓGICOS tiverem resposta ATENDE PARCIALMENTE. <p>PARECER: () APROVADA () NECESSITA DE COMPLEMENTAÇÃO</p>				

Assinatura do(a) avaliador(a): _____

Local/data: _____

CONSIDERAÇÕES DO(A) ORIENTADOR(A):

Caso o(a) orientador(a) tenha assinalado em sua avaliação algum item como “atende parcialmente”, devem ser relatos os problemas/melhorias a serem efetuadas.

Na segunda versão, caso as alterações sugeridas pelos avaliadores não sejam efetuadas, deve-se incluir uma justificativa.

[illegible]

Assinatura do(a) avaliador(a):

Local/data:

3.2 AVALIAÇÃO DO(A) COORDENADOR DE TCC

Acadêmico(a): José Alberto Zimmermann

Avaliador(a): José Roque Voltolini da Silva

ASPECTOS AVALIADOS		atende	atende parcialmente	não atende
ASPECTOS TÉCNICOS	1. INTRODUÇÃO 1.1. O tema de pesquisa está devidamente contextualizado/delimitado?			
	1.2. O problema está claramente formulado?			
	2. OBJETIVOS 2.1. O objetivo geral está claramente definido e é passível de ser alcançado?			
	2.2. São apresentados objetivos específicos (opcionais) coerentes com o objetivo geral? Caso não sejam apresentados objetivos específicos, deixe esse item em branco.			
	3. RELEVÂNCIA 3.1. A proposta apresenta um grau de relevância em computação que justifique o desenvolvimento do TCC?			
	4. METODOLOGIA 4.1. Foram relacionadas todas as etapas necessárias para o desenvolvimento do TCC?			
	4.2. Os métodos e recursos estão devidamente descritos e são compatíveis com a metodologia proposta?			
	4.3. A proposta apresenta um cronograma físico (período de realização das etapas) de maneira a permitir a execução do TCC no prazo disponível?			
	5. REVISÃO BIBLIOGRÁFICA 5.1. As informações apresentadas são suficientes e têm relação com o tema do TCC?			
	5.2. São apresentados trabalhos correlatos, bem como comentadas as principais características dos mesmos?			
	6. REQUISITOS DO SISTEMA A SER DESENVOLVIDO 6.1. Os requisitos funcionais e não funcionais do sistema a ser desenvolvido foram claramente descritos?			
	7. CONSIDERAÇÕES FINAIS 7.1. As considerações finais relacionam os assuntos apresentados na revisão bibliográfica com a realização do TCC?			
ASPECTOS METODOLÓGICOS	8. REFERÊNCIAS BIBLIOGRÁFICAS 8.1. As referências bibliográficas obedecem às normas da ABNT?			
	8.2. As referências bibliográficas contemplam adequadamente os assuntos abordados na proposta (são usadas obras atualizadas e/ou as mais importantes da área)?			
	9. CITAÇÕES 9.1. As citações obedecem às normas da ABNT?			
	9.2. As informações retiradas de outros autores estão devidamente citadas?			
	10. AVALIAÇÃO GERAL (organização e apresentação gráfica, linguagem usada) 10.1. O texto obedece ao formato estabelecido?			
	10.2. A exposição do assunto é ordenada (as idéias estão bem encadeadas e a linguagem utilizada é clara)?			
<p>A proposta de TCC deverá ser revisada, isto é, necessita de complementação, se:</p> <ul style="list-style-type: none"> qualquer um dos itens tiver resposta NÃO ATENDE; pelo menos 4 (quatro) itens dos ASPECTOS TÉCNICOS tiverem resposta ATENDE PARCIALMENTE; ou pelo menos 4 (quatro) itens dos ASPECTOS METODOLÓGICOS tiverem resposta ATENDE PARCIALMENTE. <p>PARECER: () APROVADA () NECESSITA DE COMPLEMENTAÇÃO</p>				

OBSERVAÇÕES:

Assinatura do(a) avaliador(a):

Local/data:

3.3 AVALIAÇÃO DO(A) PROFESSOR(A) DA DISCIPLINA DE TCCI

Acadêmico(a): José Alberto Zimmermann

Avaliador(a): Joyce Martins

ASPECTOS AVALIADOS		atende	atende parcialmente	não atende
ASPECTOS TÉCNICOS	1. INTRODUÇÃO 1.1. O tema de pesquisa está devidamente contextualizado/delimitado?			
	1.2. O problema está claramente formulado?			
	2. OBJETIVOS 2.1. O objetivo geral está claramente definido e é passível de ser alcançado?			
	2.2. São apresentados objetivos específicos (opcionais) coerentes com o objetivo geral? Caso não sejam apresentados objetivos específicos, deixe esse item em branco.			
	3. RELEVÂNCIA 3.1. A proposta apresenta um grau de relevância em computação que justifique o desenvolvimento do TCC?			
	4. METODOLOGIA 4.1. Foram relacionadas todas as etapas necessárias para o desenvolvimento do TCC?			
	4.2. Os métodos e recursos estão devidamente descritos e são compatíveis com a metodologia proposta?			
	4.3. A proposta apresenta um cronograma físico (período de realização das etapas) de maneira a permitir a execução do TCC no prazo disponível?			
	5. REVISÃO BIBLIOGRÁFICA 5.1. As informações apresentadas são suficientes e têm relação com o tema do TCC?			
	5.2. São apresentados trabalhos correlatos, bem como comentadas as principais características dos mesmos?			
	6. REQUISITOS DO SISTEMA A SER DESENVOLVIDO 6.1. Os requisitos funcionais e não funcionais do sistema a ser desenvolvido foram claramente descritos?			
	7. CONSIDERAÇÕES FINAIS 7.1. As considerações finais relacionam os assuntos apresentados na revisão bibliográfica com a realização do TCC?			
ASPECTOS METODOLÓGICOS	8. REFERÊNCIAS BIBLIOGRÁFICAS 8.1. As referências bibliográficas obedecem às normas da ABNT?			
	8.2. As referências bibliográficas contemplam adequadamente os assuntos abordados na proposta (são usadas obras atualizadas e/ou as mais importantes da área)?			
	9. CITAÇÕES 9.1. As citações obedecem às normas da ABNT?			
	9.2. As informações retiradas de outros autores estão devidamente citadas?			
	10. AVALIAÇÃO GERAL (organização e apresentação gráfica, linguagem usada) 10.1. O texto obedece ao formato estabelecido?			
	10.2. A exposição do assunto é ordenada (as idéias estão bem encadeadas e a linguagem utilizada é clara)?			
PONTUALIDADE NA ENTREGA			atraso de _____ dias	
<p>A proposta de TCC deverá ser revisada, isto é, necessita de complementação, se:</p> <ul style="list-style-type: none"> qualquer um dos itens tiver resposta NÃO ATENDE; pelo menos 4 (quatro) itens dos ASPECTOS TÉCNICOS tiverem resposta ATENDE PARCIALMENTE; ou pelo menos 4 (quatro) itens dos ASPECTOS METODOLÓGICOS tiverem resposta ATENDE PARCIALMENTE. <p>PARECER: () APROVADA () NECESSITA DE COMPLEMENTAÇÃO</p>				

OBSERVAÇÕES:

Assinatura do(a) avaliador(a): _____

Local/data: _____

3.4 AVALIAÇÃO DO(A) PROFESSOR(A) ESPECIALISTA NA ÁREA

Acadêmico(a): José Alberto Zimmermann

Avaliador(a):

ASPECTOS AVALIADOS		atende	atende parcialmente	não atende
ASPECTOS TÉCNICOS	1. INTRODUÇÃO 1.1. O tema de pesquisa está devidamente contextualizado/delimitado?			
	1.2. O problema está claramente formulado?			
	2. OBJETIVOS 2.1. O objetivo geral está claramente definido e é passível de ser alcançado?			
	2.2. São apresentados objetivos específicos (opcionais) coerentes com o objetivo geral? Caso não sejam apresentados objetivos específicos, deixe esse item em branco.			
	3. RELEVÂNCIA 3.1. A proposta apresenta um grau de relevância em computação que justifique o desenvolvimento do TCC?			
	4. METODOLOGIA 4.1. Foram relacionadas todas as etapas necessárias para o desenvolvimento do TCC?			
	4.2. Os métodos e recursos estão devidamente descritos e são compatíveis com a metodologia proposta?			
	4.3. A proposta apresenta um cronograma físico (período de realização das etapas) de maneira a permitir a execução do TCC no prazo disponível?			
	5. REVISÃO BIBLIOGRÁFICA 5.1. As informações apresentadas são suficientes e têm relação com o tema do TCC?			
	5.2. São apresentados trabalhos correlatos, bem como comentadas as principais características dos mesmos?			
	6. REQUISITOS DO SISTEMA A SER DESENVOLVIDO 6.1. Os requisitos funcionais e não funcionais do sistema a ser desenvolvido foram claramente descritos?			
	7. CONSIDERAÇÕES FINAIS 7.1. As considerações finais relacionam os assuntos apresentados na revisão bibliográfica com a realização do TCC?			
ASPECTOS METODOLÓGICOS	8. REFERÊNCIAS BIBLIOGRÁFICAS 8.1. As referências bibliográficas obedecem às normas da ABNT?			
	8.2. As referências bibliográficas contemplam adequadamente os assuntos abordados na proposta (são usadas obras atualizadas e/ou as mais importantes da área)?			
	9. CITAÇÕES 9.1. As citações obedecem às normas da ABNT?			
	9.2. As informações retiradas de outros autores estão devidamente citadas?			
	10. AVALIAÇÃO GERAL (organização e apresentação gráfica, linguagem usada) 10.1. O texto obedece ao formato estabelecido?			
	10.2. A exposição do assunto é ordenada (as idéias estão bem encadeadas e a linguagem utilizada é clara)?			
<p>A proposta de TCC deverá ser revisada, isto é, necessita de complementação, se:</p> <ul style="list-style-type: none"> qualquer um dos itens tiver resposta NÃO ATENDE; pelo menos 4 (quatro) itens dos ASPECTOS TÉCNICOS tiverem resposta ATENDE PARCIALMENTE; ou pelo menos 4 (quatro) itens dos ASPECTOS METODOLÓGICOS tiverem resposta ATENDE PARCIALMENTE. <p>PARECER: () APROVADA () NECESSITA DE COMPLEMENTAÇÃO</p>				

OBSERVAÇÕES:

Assinatura do(a) avaliador(a):

Local/data:

UNIVERSIDADE REGIONAL DE BLUMENAU
CENTRO DE CIÊNCIAS EXATAS E NATURAIS
CURSO DE CIÊNCIA DA COMPUTAÇÃO – BACHARELADO

**FERRAMENTA PARA AUXÍLIO NA ANÁLISE DE IMPACTO
E RASTREABILIDADE DE REQUISITOS NA GESTÃO DE
MUDANÇAS**

JOSÉ ALBERTO ZIMERMANN

BLUMENAU
2010

JOSÉ ALBERTO ZIMERMANN

**FERRAMENTA PARA AUXÍLIO NA ANÁLISE DE IMPACTO
E RASTREABILIDADE DE REQUISITOS NA GESTÃO DE
MUDANÇAS**

Proposta de Trabalho de Conclusão de Curso
submetida à Universidade Regional de
Blumenau para a obtenção dos créditos na
disciplina Trabalho de Conclusão de Curso I
do curso de Ciência da Computação —
Bacharelado.

Prof. Marcel Hugo - Orientador

**BLUMENAU
2010**

1 INTRODUÇÃO

A crescente busca pela qualidade de um produto exige cada vez mais que as empresas fabricantes de software busquem um processo ou um modelo no qual se siga uma metodologia de trabalho de conhecimento de todos os envolvidos. Segundo Oliveira (2006, p. 14), diversas empresas optam por processos que contribuam no gerenciamento de mudanças. O objetivo do controle de mudanças, desta forma, é assegurar que as alterações feitas em um projeto sejam consistentes e passíveis de mensurar o impacto que irão gerar. No mercado, existem diversas ferramentas que contemplam as fases de elaboração e desenvolvimento de um sistema, auxiliando inclusive a implantação e uso de processos de software bem definidos.

Comumente se nota que na área de Tecnologia da Informação (TI) as empresas encontram dificuldades em gerir o processo de mudança, em virtude de fatores como a velocidade e dinâmica com que elas ocorrem. As mudanças no software são feitas em resposta a solicitações de modificação de requisitos de um projeto e isso deve ser feito de maneira que a estrutura fundamental já existente permaneça estável.

Sommerville (2003, p. 518) afirma que 65% da manutenção de um sistema está relacionada à implementação de novos requisitos, 18% na modificação de requisitos já existentes e 17% na correção de defeitos de um sistema. Desta maneira, a manutenção pode ser considerada como uma extensão do processo de desenvolvimento de software, com atividades associadas às de especificação, projeto, implementação e testes.

Neste sentido, dentro do processo de manutenção, Pressman (1995, p. 883) afirma que um dos problemas clássicos associados à manutenção é a dificuldade em se rastrear a evolução do software, uma vez que as mudanças não estão devidamente documentadas. Já Borges (2003, p. 59) defende que para se obter o controle sobre a estabilidade de requisitos de software, durante o projeto de desenvolvimento, é necessário acompanhar quantitativamente todas as alterações em requisitos ocorridas, permitindo inclusive mensurar o curso das alterações, em termos de esforço empregado na manutenção dos artefatos.

Diante do exposto, neste trabalho será desenvolvida uma ferramenta em forma de *plugin* para ser integrado ao ambiente de desenvolvimento Eclipse ao qual, através de uma fonte de dados no formato *eXtensible Markup Language* (XML), gerado com o apoio de bibliotecas da ferramenta *Computer-Aided Software Engineering* (CASE) Enterprise Architect (EA), permitirá ao usuário efetuar o rastreamento dos requisitos de software implementados nos códigos fontes do projeto.

Este mapeamento¹ será realizado com a inserção de anotações, após a leitura e interpretação do arquivo XML com os dados do projeto, nos códigos fontes do sistema. Estas anotações, por sua vez, identificarão quais são os requisitos que atendem determinada classe ou método. Isto possibilitará que os códigos fontes sejam rastreáveis pela ferramenta. O usuário também poderá incluir, alterar e remover anotações, uma vez que será disponibilizada o conjunto de anotações anteriormente descritas.

Com o desenvolvimento da ferramenta também será possível efetuar a análise de impacto que uma mudança pode gerar no projeto. Será possível ponderar qual é o custo que uma determinada alteração gera para o sistema, apresentando desta forma, quais requisitos que serão alterados e qual a representação desses em relação ao projeto.

1.1 OBJETIVOS DO TRABALHO

Este trabalho tem como objetivo desenvolver uma ferramenta em forma de *plugin* para o ambiente de desenvolvimento Eclipse, que permita ao usuário efetuar a rastreabilidade entre os artefatos produzidos na fase de elaboração do projeto de software e os códigos fontes da aplicação, bem como possibilitar a análise de impacto dos requisitos que serão alterados.

Os objetivos específicos podem ser detalhados da seguinte maneira:

- a) disponibilizar uma funcionalidade que efetue o mapeamento dos requisitos de software nos códigos fontes do sistema;
- b) definir um conjunto de anotações que possibilitarão ao usuário marcar nos códigos fontes quais trechos do código atendem um requisito;
- c) ponderar os custos que as mudanças em requisitos podem representar no projeto, através do processo *Analytical Hierarchy Process* (AHP)²;
- d) disponibilizar uma interface de consulta que permita ao usuário efetuar a análise de impacto e custo, relativo às alterações no projeto.

¹O relacionamento entre requisitos (artefatos gerados na fase de projeto) e códigos fontes (artefatos gerados na fase de implementação) deverá ser feito através de *links* entre os artefatos na fase de modelagem do projeto. A ferramenta EA possui funcionalidades que permitem esta ligação e caberá ao usuário efetuar o relacionamento correto entre os artefatos.

²Para se quantificar os valores das alterações dos requisitos é necessária a utilização de uma métrica em conjunto com o AHP, como o *Function Point Analysis* (FPA) ou *Use Case Points* (UCP). A quantificação de valores e adoção de métricas não será abrangida pelo escopo deste trabalho.

1.2 RELEVÂNCIA DO TRABALHO

Hamilton e Beeby (1991, p. 1) afirmam que a rastreabilidade de requisitos é um procedimento que provê facilidades para assegurar a efetiva eliminação de erros em uma aplicação. Isso se dá pelo fato de que a implementação da rastreabilidade em um projeto permite que a equipe de analistas efetue um levantamento de quais impactos uma alteração pode causar na aplicação, antes mesmo de ela ser realizada. Segundo Hazan e Leite (2003), os requisitos de software são voláteis e podem evoluir, adicionando uma maior complexidade nas suas funcionalidades. Uma das maneiras para se diminuir e prever o impacto desta evolução é adotar um processo de medição, visando dimensionar o tamanho dos requisitos em relação ao projeto.

Um dos fatores de complexidade do trabalho é a construção de um mecanismo de mapeamento dos códigos fontes, que possibilite ao usuário obter a rastreabilidade de requisitos. Para que isso ocorra é requerida a disponibilização de um conjunto de instruções que sejam capazes de ser interpretadas por uma biblioteca e também que sirvam como identificadores para os procedimentos de marcação escolhidos pelo usuário. Outro fator que pode ser relacionado é a dificuldade em avaliar as características qualitativas dos requisitos de software.

Desta maneira, o presente trabalho possibilita a contribuição da melhora da etapa de análise de impacto nas alterações constantes que um software pode sofrer, provendo a rastreabilidade de artefatos ao longo dos códigos fontes que serão alterados, bem como a ponderação dos custos de determinada alteração.

1.3 METODOLOGIA

O trabalho será desenvolvido observando as seguintes etapas:

- a) levantamento bibliográfico: realizar levantamento bibliográfico sobre processos de desenvolvimento de software, gestão de mudanças, rastreabilidade de requisitos de software e metodologia AHP para estimação de custos dos requisitos de software;
- b) elicitação de requisitos: detalhar e reavaliar os requisitos da ferramenta, observando as necessidades levantadas durante a revisão bibliográfica;

- c) especificação da ferramenta: especificar a ferramenta com análise orientada a objetos utilizando a *Unified Modeling Language* (UML). Será utilizada a ferramenta EA 7.0 para o desenvolvimento dos diagramas de casos de uso, classes e de sequência (do gerador do arquivo XML, do manipulador de anotações e do analisador de custos);
- d) implementação do gerador do arquivo XML: implementar a geração do XML com os dados do projeto. Será utilizada a biblioteca Eaapi e SSJavaCom (SPARX SYSTEMS, 2010), que permitem acessar a funcionalidade de exportação disponibilizada pelo EA. A leitura do arquivo XML será efetuada com a biblioteca *Java Document Object Model* (JDOM) (HUNTER; MCLAUGHLIN, 2007) que permite a leitura e manipulação das informações contidas no arquivo. Será utilizado o ambiente de programação Eclipse 3.5 e linguagem de programação Java na versão 1.6;
- e) implementação do manipulador de anotações: implementar um conjunto de anotações (*annotations*) que serão inseridas no código fonte, a fim de marcar a implementação dos requisitos. Para esta implementação será utilizada a biblioteca AspectWerkz (BON; VASSEUR, 2005), que provê uma *Application Programming Interface* (API) de definição de anotações em um arquivo no formato XML e também a recuperação das anotações em arquivos de código fonte. Será utilizado o ambiente de programação Eclipse 3.5 e linguagem de programação Java na versão 1.6;
- f) implementação do analisador de custos: implementar a ponderação do valor dos requisitos, com base na metodologia AHP. Será criada uma API de apoio à metodologia, que será responsável por analisar e fornecer o valor dos requisitos. Será utilizado o ambiente de programação Eclipse 3.5 e linguagem de programação Java na versão 1.6;
- g) criação de *template* para o EA: criar um *template* do EA com o intuito do usuário poder informar o valor de complexidade dos requisitos do projeto. Este valor de complexidade será utilizado pela API de apoio ao AHP, descrito acima. Para a criação deste *template*, o próprio EA fornece um mecanismo para a personalização de modelos e campos, de acordo com a necessidade do usuário;
- h) testes: utilizar especificações e implementação de projetos já existentes, aplicando os conceitos de rastreabilidade abordados e avaliar a ponderação do custo dos requisitos (análise de impacto).

As etapas serão realizadas nos períodos relacionados no Quadro 1.

etapas / quinzenas	2011									
	fev.		mar.		abr.		maio		jun.	
	1	2	1	2	1	2	1	2	1	2
levantamento bibliográfico										
elicitação de requisitos										
especificação da ferramenta										
implementação do gerador do arquivo XML										
implementação do manipulador de anotações										
implementação do analisador de custos										
criação de <i>template</i> para o EA										
testes										

Quadro 1 - Cronograma

2 REVISÃO BIBLIOGRÁFICA

A seção 2.1 descreve as fases de um processo de desenvolvimento de software e trata brevemente sobre a gerência de requisitos. Na seção 2.2 é explicada de maneira objetiva os benefícios da gestão de mudanças em um projeto, mostrando os estágios em que ela deve ser executada. Na seção 2.3 é explicado o conceito e as funcionalidades da rastreabilidade de requisitos e artefatos. A seção 2.4 apresenta a metodologia AHP, explicando de maneira simplificada, o seu funcionamento. Por fim, na seção 2.5 são descritos os trabalhos correlatos.

2.1 PROCESSOS DE DESENVOLVIMENTO DE SOFTWARE

Sommerville (2003, p. 7) define um processo de software como sendo uma série de atividades e resultados que tem como finalidade um produto de software. Os processos de softwares são complexos e devem seguir uma série de procedimentos, a fim de garantir um produto que atenda as diversas necessidades do cliente. Neste sentido, existem diversos processos, que têm como características comuns as seguintes fases:

- a) especificação: é onde ocorre a definição das funcionalidades do software, indicando inclusive, as suas restrições;
- b) projeto e implementação de software: nesta etapa é onde ocorre o desenvolvimento do software, atendendo a especificação anteriormente descrita;
- c) validação do software: o software necessita ser validado, garantindo que ele atenda as necessidades do cliente;
- d) evolução do software: o software pode evoluir para atender as necessidades mutáveis do cliente.

Ao se colocar um software em uso, novos requisitos são detectados como necessários e os já existentes são alterados à medida que o sistema é utilizado. Partes dos requisitos podem ser modificadas para corrigir erros encontrados na operação ou até mesmo para melhorar o desempenho da aplicação.

É importante que o projeto de software seja atualizado com a evolução do sistema a partir da inclusão, alteração ou exclusão de requisitos, a fim de se identificar de maneira clara quais são os componentes envolvidos nas alterações de um sistema.

2.1.1 Gerência de requisitos

Sommerville (2003, p. 117) afirma que os requisitos nos grandes sistemas estão em constante modificação.

A gerência de requisitos está associada ao processo de controle do processo de desenvolvimento tendo como referência a base de requisitos. Segundo Tocchetto (2007, p. 20), o principal objetivo do gerenciamento de requisitos é “descobrir, organizar, comunicar e administrar o impacto das mudanças de requisitos de um software”. Desta maneira, Tocchetto (2007, p. 20) afirma que os principais benefícios do gerenciamento de requisitos são:

- a) maior controle em projetos de grande porte;
- b) aumento da qualidade de software e maior satisfação do cliente;
- c) diminuição dos custos de projeto e atrasos.

2.2 GESTÃO DE MUDANÇAS

O processo de manutenção é explicado por Sommerville (2003, p. 521) ao se iniciar um conjunto de pedidos de mudança por parte dos usuários, gerentes ou clientes. Custo e impacto devem ser analisados para se verificar quanto do sistema será afetado pela alteração e quanto poderá custar para desenvolver esta mudança.

Assim sendo, em uma situação ideal, o estágio de desenvolvimento de alterações deverá modificar especificação, projeto e implementação do sistema, com o objetivo de refletir no software. Desta maneira, os novos requisitos que refletem as mudanças no sistema são propostos, analisados e validados, para que a mudança seja consistente e não impacte de maneira negativa no restante do sistema.

Neste contexto, a gerência de mudança tem o objetivo de garantir que as mudanças sejam realizadas com sucesso, sem que haja perda da qualidade do software. Para que isso ocorra, é necessário que esta fase da manutenção do software controle as solicitações de manutenção, aprove as solicitações e a partir daí, estabeleça-se como a modificação será implementada e quais restrições que deverão ser respeitadas, definindo-se de maneira clara qual o escopo da alteração. Segundo Sommerville (2003, p. 121), existem três estágios em um processo de gerenciamento de mudanças:

- a) análise do problema e especificação da mudança;
- b) análise e custo da mudança;
- c) implementação de mudanças.

Nestes estágios são utilizadas informações de rastreabilidade para se estimar o tamanho e o custo da mudança. O custo da mudança é estimado em termos das modificações no documento de requisitos e, se apropriado, no projeto de sistemas e na implementação. Desta maneira, entende-se que é de extrema importância possuir o documento de requisitos e o projeto de software constantemente atualizado.

2.3 RASTREABILIDADE DE REQUISITOS

A rastreabilidade é uma técnica que provê o relacionamento entre diversos requisitos, projeto e a sua implementação. É ela quem auxilia no entendimento dos relacionamentos que existe entre os requisitos e o projeto de software desenvolvido. Para Hamilton e Beeby (1991, p. 1), a rastreabilidade é a habilidade de descobrir o histórico de cada funcionalidade do sistema.

Segundo Hazan e Leite (2003), a rastreabilidade permite garantir como e por que os artefatos atendem os requisitos dos clientes, sendo uma forma fundamental para entender os relacionamentos existentes entre requisitos e outros artefatos que fazem parte do processo de software. Desta maneira, a elaboração de um projeto de software deve produzir requisitos que sejam rastreáveis, ou seja, que sejam capazes de serem rastreados a partir da sua origem.

Um dos maiores desafios da engenharia de requisitos é criar um mecanismo que possibilite a criação de *links* entre os requisitos e os códigos fontes. Desta maneira, um dos benefícios que a rastreabilidade oferece é a possibilidade de se cruzar as informações especificadas na fase de projeto (requisitos) e os itens desenvolvidos na fase de implementação, que são os códigos fontes.

A rastreabilidade permite que os custos das alterações em requisitos de um projeto possam ser mais precisos. Também permite que as mudanças possam ocorrer sem a dependência do engenheiro ou programador de conhecerem as áreas afetadas por tais mudanças.

É possível afirmar que a manutenção da rastreabilidade pode ser um trabalho penoso, extenso e até mesmo inviável, quando não auxiliado por uma ferramenta especializada. Isso é

explicado por Richardson e Green (2004, p. 24) que afirmam que programadores são relutantes em manter os documentos do projeto atualizados, quebrando assim, o mecanismo que possibilita ocorrer a rastreabilidade.

2.4 METODOLOGIA AHP

A metodologia AHP foi criada por Saaty (1991) com o objetivo de ponderar as características qualitativas, permitindo assim, a ponderação e priorização de cada um dos requisitos de um projeto que estão inseridos neste contexto. O funcionamento deste processo se dá pela atribuição de pesos a fatores individuais, do menos influente ao mais influente. AHP utiliza-se de uma matriz quadrada que permite avaliar a importância de uma característica sobre a outra. A partir disso, o processo permite obter o valor percentual de cada um dos requisitos e assim, mapear o valor das fases do projeto.

Segundo Tocchetto (2007, p. 21) o método segue a seguinte sequência de passos para estabelecer o valor dos requisitos:

- a) montar uma matriz quadrada com os requisitos do sistema;
- b) comparar os requisitos dois a dois, preenchendo a relação linha X coluna de acordo com a complexidade do requisito i (linha) com o requisito j (coluna), de acordo com a escala de intensidade de importância proposta pelo método;
- c) estimar o valor do auto-vetor;
- d) normalizar o auto-vetor;
- e) multiplicar o valor do auto-vetor por $1/n$, onde n é quantidade requisitos.

2.5 TRABALHOS CORRELATOS

Existem disponíveis ferramentas comerciais e acadêmicas que abordam a temática da rastreabilidade e análise de impacto com a mudança de requisitos, baseados em uma métrica. Desta maneira, é possível encontrar desde trabalhos de conclusão de curso até dissertações de mestrado. Dentre estes trabalhos, foi selecionada a ferramenta para o auxílio à rastreabilidade de Santos e Aragão (2009), a ferramenta CASE de gerência de Batista (2007) e o ambiente de

acompanhamento de custos de requisitos proposto por Tocchetto (2007), aos quais são descritos a seguir.

2.5.1 Ferramenta para auxílio à rastreabilidade de artefatos de software

A partir da identificação de que no processo de criação de um software era necessário associar os artefatos (códigos fontes) aos requisitos descritos na fase de projeto, Santos e Aragão (2009) desenvolveram uma ferramenta (chamada de *Add-In*) que se integra com o ambiente de desenvolvimento Visual Studio 2008. Nesta, o documento de especificação é escrito através da ferramenta Rational Requisite Pro, a qual pertence a IBM³.

Para que ocorra o mapeamento entre o documento de especificação e os códigos fontes (*links* de rastreabilidade) é necessário que o usuário importe os requisitos especificados manualmente para a ferramenta. Após, o usuário pode associar os requisitos importados com os artefatos do projeto, onde o nível de abstração é feito até os métodos de uma classe. Os dados da associação são gravados em um banco de dados MS SQL Server⁴. Também é disponibilizada uma interface de consulta, na qual é possível verificar quais requisitos estão associados aos métodos de uma classe. É possível também gerar relatórios destas consultas.

2.5.2 Ferramenta CASE de gerência de requisitos de software integrada com EA

A ferramenta proposta por Batista (2007) tem como objetivo o gerenciamento de requisitos de software, propondo que se tenha um controle mais eficiente dos requisitos, focando na produção de um documento de especificação completo. A ferramenta possui integração com o EA, dando a possibilidade de se incorporar requisitos desenvolvidos por esta ferramenta.

Um dos diferenciais da ferramenta CASE em relação a outras, que possuem as mesmas finalidades, é a possibilidade de vinculação de requisitos com outros requisitos e

³ *Internacional Business Machines* (IBM).

⁴ *Microsoft Structured Query Language Server* (MS SQL Server).

casos de uso, provendo desta maneira a rastreabilidade entre os elementos do projeto. A ferramenta possui também uma funcionalidade que possibilita a configuração de diversos tipos de matriz de rastreabilidade, como a referência cruzada entre tipos de requisito e casos de uso.

A linguagem de programação utilizada para a criação da ferramenta CASE é o Borland Delphi 6.

2.5.3 AspectCost: ambiente de gerência e acompanhamento de custos e requisitos baseados em AOP⁵

Tocchetto (2007) propõe uma metodologia e disponibiliza uma ferramenta para estimar o custo total de um projeto, mensurar e controlar os custos dos requisitos no paradigma de desenvolvimento orientado a aspectos, estendendo o modelo de estimativa UCP.

As características da metodologia de Tocchetto (2007, p. 22) são explicadas da seguinte maneira:

- a) estimativa do custo total do projeto: o autor engloba os casos de uso aspectuais para propiciar uma estimativa de projeto para os que utilizam a programação orientada a aspectos;
- b) estimativa dos custos de requisitos: é baseado no processo AHP, proposto por Saaty (1991). O autor utiliza este processo para ponderar as características qualitativas, no caso, os requisitos. Como resultado final, é obtido o valor percentual do custo que cada requisito representa no projeto;
- c) controle dos custos dos requisitos: é utilizada uma tabela comparativa de relacionamentos dos requisitos e os fatores de relacionamento entre os mesmos para controlar as alterações de valores ao longo dos requisitos, e por consequência, do projeto.

Desta maneira, é possível verificar que a metodologia proposta por Tocchetto (2007) permite efetuar o controle dos valores percentuais dos requisitos, bem como efetuar o controle dos custos de um projeto, partindo da ideia que a evolução dos requisitos é constante.

⁵ *Aspect-Oriented Programming* (AOP) é um paradigma que tem como objetivo a separação do código segundo a sua importância para a aplicação, permitindo que o programador encapsule o código secundário em módulos.

3 REQUISITOS DO SISTEMA A SER DESENVOLVIDO

A ferramenta deverá:

- a) efetuar a exportação de um projeto especificado na ferramenta EA, disponibilizando estes dados em um arquivo no formato XML (Requisito Funcional – RF);
- b) interpretar o arquivo descrito no item (a), buscando quais as dependências dos requisitos descritos (RF);
- c) permitir ao usuário inserir anotações que identifiquem quais requisitos estão sendo atendidos nos códigos fontes, durante o desenvolvimento de um projeto (RF);
- d) efetuar o mapeamento dos códigos fontes de um projeto, inserindo anotações de acordo com a interpretação descrita no item (a) (RF);
- e) disponibilizar uma interface que mostre ao usuário quais as dependências (códigos fontes) associadas a um requisito do projeto (RF);
- f) ponderar qual o custo que a alteração de um requisito representa para o projeto, através do processo AHP (RF);
- g) disponibilizar uma interface que permita ao usuário consultar qual o custo que uma alteração pode causar no projeto (RF);
- h) ser implementada utilizando o ambiente de desenvolvimento Eclipse 3.5 (Requisito Não-Funcional - RNF);
- i) ser implementada utilizando a linguagem de programação Java 1.6 (RNF);
- j) utilizar as bibliotecas Eaapi e SSJavaCom para a geração do arquivo XML (RNF);
- k) utilizar a biblioteca JDOM para a leitura e manipulação do arquivo XML (RNF);
- l) utilizar a biblioteca AspectWerkz para a manipulação das anotações (RNF).

4 CONSIDERAÇÕES FINAIS

Após o término da análise da problemática que envolve a rastreabilidade de requisitos de software e análise de impacto na gestão de mudanças, verificou-se que quando estes dois processos ocorrem de maneira manual, os custos e os riscos de imprecisão numa decisão final são altos. Isto pode levar uma empresa à tomada de decisão incorreta, a um levantamento errôneo na avaliação de custos e análise de alterações no código fonte imprecisa.

Assim, o desenvolvimento da referida ferramenta será voltado para a automatização do processo de rastreabilidade e análise de impacto, auxiliando para que a tomada de decisão ocorra de maneira mais precisa e segura e contribuindo para que o gerenciamento de requisitos controle com maior eficiência a evolução dos requisitos em função da manutenção.

O uso da ferramenta efetuará o mapeamento dos requisitos de um projeto nos códigos fontes de maneira automática. Desta maneira, com o mapeamento realizado nos códigos fontes a partir de um arquivo XML do projeto, será possível efetuar a rastreabilidade de requisitos, avaliando desta maneira quais os locais que deverão ser alterados com a mudança proposta. Também serão possíveis, com o uso do processo AHP, avaliar quais são os custos desta alteração. O processo AHP fornecerá os valores correspondentes ao tamanho dos requisitos em um projeto, sendo possível, dimensionar de maneira mais precisa o impacto que isso irá gerar em um projeto.

Em relação aos trabalhos correlatos apresentados, pode-se afirmar que a ferramenta proposta se diferencia em alguns aspectos das descritas anteriormente. No que diz respeito à utilização da rastreabilidade de requisitos, a ferramenta que será desenvolvida diferencia-se dos demais trabalhos por efetuar as marcações nos códigos fontes (artefatos produzidos pela especificação), avaliando os *links* criados entre os requisitos e classes na fase de elaboração do projeto. Este processo ocorrerá sem intervenção do usuário. Outro diferencial será a disponibilização do conjunto de anotações para o usuário efetuar as marcações no código fonte. Já o uso do arquivo XML como fonte de dados é sustentado com base na experiência descrita por Batista (2007, p. 56), que encontrou limitações nas bibliotecas que efetuam a leitura do repositório do EA. Já em relação à análise de impacto e a ponderação de custos dos requisitos, será adaptado o método descrito por Tocchetto (2007), onde serão utilizadas as fases de estimativa de custos de requisitos e controle de custos dos requisitos.

Por fim, a ferramenta proposta diferencia-se dos trabalhos correlatos em função da linguagem de programação utilizada e pelo ambiente ao qual a ferramenta será integrada.

REFERÊNCIAS BIBLIOGRÁFICAS

BATISTA, Raphael M. **Ferramenta de gerência de requisitos de software integrada com Enterprise Architect**. 2007. 67 f. Trabalho de Conclusão de Curso (Bacharelado em Ciências da Computação) – Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau. Disponível em: <<http://campeche.inf.furb.br/tccs/2007-I/2007-1raphaelmarcosbatistavf.pdf>>. Acesso em: 08 set. 2010.

BORGES, Eduardo P. **Um modelo de medição para processos de desenvolvimento de software**. 2003. 154 f. Dissertação (Mestrado em Ciência da Computação) – Departamento de Ciência da Computação, Universidade Federal de Minas Gerais, Belo Horizonte. Disponível em: <<http://homepages.dcc.ufmg.br/~wilson/pesquisa/DissertacaoEduardo.pdf>>. Acesso em: 08 set. 2010.

BON, Jonas; VASSEUR, Alexandre. **AspectWerkz: plain Java AOP 2.0 API**. [S.l.], 2005. Disponível em: <<http://aspectwerkz.codehaus.org>>. Acesso em: 14 set. 2010.

HAMILTON, Vivien L.; BEEBY, Martin. Issues of traceability in integration tools. In: IEE COLLOQUIUM ON TOOLS AND TECHNIQUES FOR MAINTAINING TRACEABILITY DURING DESIGN, 1., 1991, Londres. **Proceedings...** Londres: IEEE, 1991. p. 4/1-p.4/2. Disponível em: <<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=182212>>. Acesso em: 30 ago. 2010.

HAZAN, Claudia; LEITE, Julio C. S. P. Indicadores para a gerência de requisitos. In: WORKSHOP EM ENGENHARIA DE REQUISITOS, 6., 2003, Piracicaba. **Anais eletrônicos...** Rio de Janeiro: PUC-RIO, 2003. Não paginado. Disponível em: <http://wer.inf.puc-rio.br/WERpapers/artigos/artigos_WER03/claudia_hazan.pdf>. Acesso em: 30 ago. 2010.

HUNTER, Jason; MCLAUGHLIN, Brett. **JDOM v1.1.1: API specification**. [S.l.], 2007. Disponível em: <<http://www.jdom.org>>. Acesso em: 14 set. 2010.

OLIVEIRA, Fabricio. **Software de apoio à gerência de solicitação de mudanças**. 2006. 72 f. Trabalho de Conclusão de Curso (Bacharelado em Ciências da Computação) – Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau.

PRESSMAN, Roger S. **Engenharia de software**. Tradução José Carlos Barbosa dos Santos. São Paulo: Makron Books, 1995.

RICHARDSON, Julian; GREEN, Jeff. Automating traceability for generated software artifacts. In: INTERNACIONAL CONFERENCE ON AUTOMATED SOFTWARE ENGINEERING, 19., 2004, Moffett Field. **Proceedings...** Moffett Field: IEEE, 2004. p. 24-p.33. Disponível em: <<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1342721>>. Acesso em: 30 ago. 2010.

SAATY, Thomas L. **Método de análise hierárquica**. Tradução Wainer da Silveira e Silva. São Paulo: McGraw-Hill, 1991.

SANTOS, Eder M. dos; ARAGÃO, João P. B. de. **Rastreabilidade de artefatos de software**. 2009. 43 f. Trabalho de Conclusão de Curso (Bacharelado em Informática) – Universidade Católica de Salvador, Salvador. Disponível em: <http://info.ucsal.br/banmon/Arquivos/Mono_12122009.pdf>. Acesso em: 30 ago. 2010.

SOMMERVILLE, Ian. **Engenharia de software**. 6. ed. Tradução Maurício de Andrade. São Paulo: Addison Wesley, 2003.

SPARX SYSTEMS. **Reference**: Enterprise Architect user guide. [S.l.], 2010. Disponível em: <http://www.sparxsystems.com/enterprise_architect_user_guide/automation_and_scripts/reference.html>. Acesso em: 14 set. 2010.

TOCCHETTO, André L. **AspectCost**: um ambiente de gerência e acompanhamento de custos de requisitos baseados em AOP. 2007. 90 f. Dissertação (Mestrado em Computação Aplicada) – Programa Interdisciplinar de Pós-Graduação em Computação Aplicada, Universidade do Vale do Rio dos Sinos, São Leopoldo. Disponível em: <http://bdtd.unisinos.br/tde_arquivos/1/TDE-2007-06-27T110125Z-306/Publico/aspectcost.pdf>. Acesso em: 30 ago. 2010.