



UNIVERSIDADE FEDERAL DO ESTADO DO RIO DE JANEIRO
CCET - CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA
ESCOLA DE INFORMÁTICA APLICADA
CURSO DE BACHARELADO EM SISTEMAS DE INFORMAÇÃO

GERÊNCIA DO CICLO DE VIDA DE CASOS DE USO BASEADA NA
FERRAMENTA REDMINE

DIOGO BARRETO SALEH
JOSE CARLOS MARTINS BRITZ

Orientador

Prof. Dr. Gleison Santos

2013

**GERÊNCIA DO CICLO DE VIDA DE CASOS DE USO BASEADA NA
FERRAMENTA REDMINE**

DIOGO BARRETO SALEH

JOSÉ CARLOS MARTINS BRITZ

Projeto de Graduação apresentado à
Escola de Informática Aplicada da Universidade
Federal do Estado do Rio de Janeiro (UNIRIO)
para obtenção do título de Bacharel em Sistemas
de Informação

Orientador

Prof. Dr. Gleison Santos

Rio de Janeiro

Abril, 2013

**GERÊNCIA DO CICLO DE VIDA DE CASOS DE USO BASEADA NA
FERRAMENTA REDMINE**

DIOGO BARRETO SALEH

JOSÉ CARLOS MARTINS BRITZ

Orientador

Prof. Dr. Gleison Santos

Aprovado em ___/ ___/ ___

Prof. Dr. Gleison Santos

Prof. Dra. Claudia Cappelli

Prof. Dra. Renata Araujo

Os autores deste Projeto autorizam a ESCOLA DE INFORMÁTICA APLICADA da UNIRIO a divulgá-lo, no todo ou em parte, resguardados os direitos autorais conforme legislação vigente.

Rio de Janeiro, ___ de Abril de 2013.

Diogo Barreto Saleh

José Carlos Martins Britz

RESUMO

GERÊNCIA DO CICLO DE VIDA DE CASOS DE USO BASEADA NA FERRAMENTA REDMINE

Diogo Barreto Saleh

José Carlos Martins Britz

Resumo da Monografia submetida ao corpo docente do curso de Graduação em Sistemas de Informação – Universidade Federal do Estado do Rio de Janeiro – UNIRIO, como parte dos requisitos necessários à obtenção do título de Bacharel em Sistemas de Informação.

Esse trabalho apresenta uma abordagem para a gerência do ciclo de vida de casos de uso. Esta abordagem está baseada na definição de um fluxo de trabalho com elementos de gerência de requisitos e de gerência de configuração para controle do ciclo de vida de casos de uso. Além disso, essa abordagem é completada pela automatização desde fluxo de trabalho em uma instância da ferramenta Redmine.

Palavras-chave: casos de uso, gerência de configuração, Redmine.

Rio de Janeiro

Abril, 2013

Agradecimentos

Agradecemos às nossas famílias pelo apoio, à nossa diretora Leila Andrade que sempre nos auxiliou quando precisamos, às professoras Claudia Cappelli e Renata Araujo que participaram de nossa banca, ao nosso orientador Gleison Santos que sempre fora prestativo durante a realização do trabalho e, por fim, aos nossos amigos pela compreensão do momento de tanta dedicação.

Sumário

1	Introdução	1
1.1	Contexto	1
1.2	Objetivo	3
1.3	Estrutura do Trabalho	4
2	Fundamentação Teórica	5
2.1	Levantamento de requisitos	5
2.2	Casos de uso	6
2.3	Avaliação de casos de uso	7
2.4	Construção de casos de uso	7
2.5	Aprovação de codificação	8
2.6	Gerência de configuração de software	8
2.7	Elementos de um sistema de Gerência de Configuração.....	10
2.8	Itens de configuração de software	10
2.9	Repositório	11
2.10	Controle de Versão	12
2.11	Controle de modificação	13
2.12	Wiki	14
2.13	Trabalhos relacionados	16
3	Utilização do Redmine para gerência de casos de uso	18
3.1	Motivação pelo uso da ferramenta Redmine	18
3.2	Processo de apoio para gerência de casos de uso	21
3.2.1	Levantar requisitos	22
3.2.2	Definir casos de uso.....	23

3.2.3	Avaliar casos de uso pela equipe do projeto.....	23
3.2.4	Avaliar casos de uso pelo cliente.....	24
3.2.5	Construir caso de uso.....	26
3.2.6	Aprovar construção.....	26
3.2.7	Alterar casos de uso.....	26
3.3	Configuração do Redmine.....	27
3.4	Exemplo do uso do workflow na ferramenta Redmine.....	28
3.4.1	Fluxo de sucesso.....	28
3.4.2	Fluxo alternativo.....	35
3.5	Considerações finais.....	38
4	Conclusão	39
4.1	– Principais Contribuições	39
4.2	– Trabalhos Futuros.....	40
	Referências Bibliográficas	41

Índice de Tabelas

TABELA 3-1: COMPARATIVO TÉCNICO ENTRE AS FERRAMENTAS [WIKIPÉDIA, 2012]..	20
TABELA 3-2: COMPARATIVO GERAL ENTRE AS FERRAMENTAS.....	21
TABELA 3-3: CRITÉRIOS DE AVALIAÇÃO DOS CASOS DE USO PELA EQUIPE DE PROJETO	24
TABELA 3-4: CRITÉRIOS DE AVALIAÇÃO DOS CASOS DE USO PELO CLIENTE	25

Índice de Figuras

FIGURA 1: PROCESSO DE CONTROLE DE MODIFICAÇÕES (PRESSMAN, 2006).....	14
FIGURA 2: PROCESSO DE APOIO PARA GERÊNCIA DE CASOS DE USO.....	22
FIGURA 3: DIAGRAMA DE ESTADOS.....	22
FIGURA 4: TELA DA CRIAÇÃO DE NOVOS ESTADOS.....	27
FIGURA 5: CRIAÇÃO DO WORKFLOW NO REDMINE.....	29
FIGURA 6: TELA DE CRIAÇÃO DE CASO DE USO, ESTADO “PRONTO PARA INICIAR”.....	30
FIGURA 7: TELA DO ESTADO “EM ELABORAÇÃO”.....	30
FIGURA 9: CASO DE USO INSERIDO NA WIKI.....	31
FIGURA 11: TELA DO ESTADO “APROVADO PELA EQUIPE”.....	32
FIGURA 13: TELA DO ESTADO “APROVADO PELO CLIENTE”.....	33
FIGURA 15: TELA DO ESTADO “EM AVALIAÇÃO DA CONSTRUÇÃO”.....	34
FIGURA 16: TELA DO ESTADO “CONSTRUÇÃO APROVADA”.....	35
FIGURA 17: TELA DE CRIAÇÃO DE CASO DE USO NO FLUXO ALTERNATIVO, ESTADO “PRONTO PARA INICIAR”.....	36
FIGURA 18: TELA DO ESTADO “EM ELABORAÇÃO” NO FLUXO ALTERNATIVO.....	36
FIGURA 19: TELA DO ESTADO “DEFINIDO” NO FLUXO ALTERNATIVO.....	37
FIGURA 20: TELA DO ESTADO “EM AVALIAÇÃO PELA EQUIPE” NO FLUXO ALTERNATIVO	37

1 Introdução

O objetivo deste capítulo é apresentar o contexto e o conteúdo do trabalho, além de apresentar a estrutura dos capítulos contidos nele.

1.1 Contexto

A análise de requisitos, na engenharia de software, engloba a execução de tarefas que lidam com investigação, definição e escopo de novos sistemas ou alterações em sistemas existentes. Ao final da execução destas tarefas, é criado um documento consolidando as informações obtidas. Dessa forma, a análise de requisitos é uma parte importante do processo de projeto de sistemas, na qual a equipe de implantação e a equipe de produtos identificam as necessidades ou requisitos de um cliente. Uma vez que os requisitos do sistema tenham sido identificados, os projetistas de sistemas estarão preparados para projetar a solução (BOHM, 2010).

A atividade de análise de requisitos é relevante para o sucesso de um projeto de desenvolvimento. Deverá estar condizente com o que foi pedido pelo cliente para que seja passada para a fase de projeto de maneira esclarecedora. Quando o projeto de software é iniciado, é essencial que esses requisitos estejam explícitos para todos os envolvidos no processo. Além disso, é muito importante também a participação dos *stakeholders* no levantamento dessas informações, visto que nesta fase irão ser discutidas as necessidades do cliente.

Com as ações e os objetivos das funcionalidades do software bem definidos, as atividades realizadas posteriormente tornam-se mais simples de serem realizadas, evitando um retrabalho maior caso os requisitos não fossem especificados de maneira clara. Para que este processo ocorra com sucesso, os requisitos devem ser, além de claros: completos, sem ambiguidade, consistentes e testáveis. Os requisitos que não apresentem estas qualidades são problemáticos: devem ser revistos e renegociados com os clientes e usuários. Já que todo o processo é imprescindível para o projeto,

deve-se, portanto, compreender a importância da análise de requisitos, para garantir que a necessidade seja atendida e a solução encontrada.

Uma forma de representar o resultado da análise de requisitos é na forma de casos de usos (BEZERRA, 2006). Um caso de uso deve especificar as expectativas de comportamento de um sistema, quando este apoia uma transação de negócio do cliente. Essas expectativas de comportamento são divididas em cenários. Partindo do cenário principal de sucesso, que é aquele em que tudo transcorre conforme o esperado e deve corresponder à grande maioria das transações, chegando a cada cenário em que alguma barreira ou dificuldade pode provocar comportamentos específicos no sistema ou até mesmo impedir a realização da transação iniciada. Estes artefatos são de suma importância para um projeto de desenvolvimento de software, sendo possível identificar as necessidades da organização, traçar um plano efetivo, além de reduzir os custos com a adoção de ferramentas que implementem essas atividades. Neste contexto, a especificação por casos de uso é uma ferramenta para driblar as dificuldades de captação, identificação, descoberta, registro, validação e manutenção de requisitos de software (OLIVEIRA, 2009).

No contexto de um projeto de desenvolvimento, outro fator que merece destaque é a gerência de configuração, que possui o propósito de manter a integridade de todos os produtos de trabalho de um processo ou projeto e disponibilizá-los a todos os envolvidos. Além disso, garante que os desenvolvedores estarão trabalhando com a versão correta dos artefatos. A gerência de configuração tem importância no âmbito de desenvolvimento do projeto, pois consegue identificar quem mudou determinada parte do programa, o que mudou e quando mudou. Pode também recuperar uma versão anterior que estava instalada em um cliente, ou, então, evoluir uma versão do software, adicionando novas funcionalidades, além de gerenciar mudanças feitas por outros desenvolvedores, etc. Ao longo do ciclo de vida de um software, diversas informações são adicionadas e alteradas. Tratando-se do processo de desenvolvimento, a gerência de configuração pode ser mais um componente importante e eficaz que, além de muitas outras vantagens, organiza e disponibiliza de maneira simples as mudanças realizadas no processo, com diversas

informações sobre a alteração efetuada, como por exemplo: responsável, hora da mudança, o que mudou, fornecendo um histórico das alterações, etc. A Gerência de Configuração de Software surgiu da necessidade de controlar essas alterações para evitar a perda de controle do projeto de software. Assim, ela pode ser definida como o controle da evolução de sistemas complexos ou, de forma mais pragmática, como a disciplina que permite manter produtos de software em evolução sob controle, e, portanto, contribuindo para satisfazer restrições de qualidade e de cronograma (FIGUEIREDO *et al.*, 2004).

1.2 Objetivo

O objetivo deste trabalho é utilizar uma ferramenta de gerência de workflow para a gerência de casos de uso desde o início de sua elaboração até sua codificação. O workflow a ser utilizado deve incluir procedimentos de gerência de configuração de forma a aumentar o controle sobre as tarefas relacionadas à gerência de requisitos. Como resultado secundário, os casos de usos serão disponibilizados para as partes envolvidas no processo, gerando logs como: data, responsável pelo requisito, solicitante, quem modificou e porquê etc. Assim, será criado um mecanismo de rastreamento dos requisitos para os desenvolvedores que utilizarão e trabalharão com a documentação armazenada. É utilizado também um suporte a *wiki*, para gerenciar os requisitos de uma forma mais eficiente. A ferramenta adotada para o projeto é o Redmine¹.

Para possibilitar esse controle, será utilizada a ferramenta Redmine, que possui integrados o controle de fluxo de trabalho e um sistema interno de *wiki*.

¹<http://www.redmine.org>

1.3 Estrutura do Trabalho

Este texto está estruturado como se segue. O Capítulo 1 apresenta a introdução, mencionando os componentes necessários para os processos de software no trabalho em questão – análise de requisitos; casos de usos; gerência da configuração. Além de destacar também a importância desses componentes para os projetos. No Capítulo 2 são discutidos gerência da configuração, análise de requisitos, casos de uso e *wiki*. O Capítulo 3 aborda a ferramenta utilizada no trabalho – o Redmine, justificando sua escolha e apresentando sua configuração, o processo de apoio definido e o funcionamento do workflow criado. Por fim, o Capítulo 4 apresenta um resumo do trabalho, finalizando o texto e sintetizando o tema que será tratado na monografia.

2 Fundamentação Teórica

O objetivo desse capítulo é apresentar os conceitos abordados que são: levantamento de requisitos em um projeto de software, casos de uso e sua construção, gerência de configuração e wiki.

2.1 Levantamento de requisitos

A tarefa de desenvolvimento de software engloba uma série de fases e atividades que, independentemente da metodologia escolhida, ocorrem para a realização do seu objetivo maior: entregar software funcionando corretamente dentro do orçamento e prazos previstos para o seu desenvolvimento.

É fundamental definir corretamente o que vem a ser um requisito: é uma especificação de uma característica ou propriedade que um sistema deve possuir ou fazer, assim como sua restrição de operação (SILVA, 2009). Os requisitos podem ser definidos por diversas classificações tais como: requisitos de negócio, funcionais, não funcionais, etc.

A atividade de levantamento de requisitos corresponde à etapa de compreensão do problema aplicada ao desenvolvimento de software. O principal objetivo do levantamento de requisitos é que usuários e desenvolvedores tenham a mesma visão do problema a ser resolvido. Nessa etapa, os desenvolvedores, juntamente com os clientes, tentam levantar e definir as necessidades (requisitos) dos futuros usuários do sistema a ser desenvolvido. Durante o levantamento de requisitos, a equipe de desenvolvimento tenta entender o domínio do negócio que deve ser automatizado pelo sistema de software, além de incluir um estudo exploratório das necessidades dos usuários e da situação atual do sistema, caso exista (BEZERRA, 2006).

Algumas empresas ignoram a etapa de levantamento de requisitos devido a diversos fatores como prazos curtos, prioridade da etapa de codificação, execução de outras atividades em paralelo etc. Essa é uma característica muito comum em empresas com pouca maturidade em desenvolvimento de software. Os produtos

dessas organizações, em geral, não possuem um padrão de qualidade relevante, podendo apresentar defeitos em suas distribuições finais.

Pode-se afirmar que sem uma correta definição e gestão dos requisitos é possível que o projeto não alcançará o sucesso comprometido, frustrando as expectativas do cliente e comprometendo as metas e planos da empresa contratante.

Uma das formas de se descrever requisitos em um projeto de software é utilizando casos de uso.

2.2 Casos de uso

Um caso de uso (do inglês *use case*) é a especificação de uma sequência de interações entre um sistema e os agentes externos que utilizam esse sistema. Um caso de uso deve definir o uso de uma parte da funcionalidade de um sistema, sem revelar a estrutura e o comportamento interno desse sistema (BEZERRA, 2006). Cada caso de uso deve ser definido através da descrição narrativa das interações que ocorrem entre o(s) elemento(s) externo(s) e o sistema. A UML (Linguagem de Modelagem Unificada) não define o formato e o grau de abstração a serem utilizados na descrição de um caso de uso. Conseqüentemente há vários formatos de descrição propostos na literatura, assim como vários são os graus de abstração utilizados. Na descrição de casos de uso de um sistema podem ser utilizados diferentes formatos, graus de detalhamento e graus de abstração, independentes um do outro.

É muito comum empresas discutirem e proporem soluções para que os seus produtos finais atinjam uma qualidade de excelência reconhecida no mercado. Para isso, investem muito dos seus recursos nas melhorias de seus processos, assim como em bons desenvolvedores, plataformas de desenvolvimento atuais, etc. Talvez o erro principal destas empresas seja o não reconhecimento da importância da fase de levantamento de requisitos e definição dos casos de uso para criação de seus produtos finais.

2.3 Avaliação de casos de uso

A variação que os requisitos sofrem durante o ciclo de vida de um projeto em relação a uma média aceitável de variações é o que se chama de volatilidade de requisitos. São muitos os fatores que podem determinar se os requisitos de um determinado projeto são voláteis ou não, como, por exemplo, a imaturidade do cliente, a sonegação de informações, a inexperiência dos analistas de requisitos, o mercado em que o projeto está inserido etc. Quanto mais sensível um requisito é a um fator externo, mais volátil ele é; e quanto mais volátil é o requisito, maior o risco de não se entregar o projeto no prazo estipulado ou dentro da faixa de custos previamente aprovados (CASTRO *et al.*,2005).

Portanto, esta etapa é de suma importância para o prosseguimento do projeto de software, já que não é aconselhável a adição e alterações de requisitos nas fases que se seguem. É esperado que as partes interessadas no projeto colaborem, concentrando energia nesta fase para evitar um retrabalho desnecessário futuramente com o surgimento de novos requisitos.

2.4 Construção de casos de uso

Nesta fase os casos de uso são codificados, ou seja, ocorre a tradução computacional obtida nas fases anteriores de levantamento de requisitos e definição dos casos de uso, em código executável mediante o uso de uma ou mais linguagens de programação.

Em um processo de desenvolvimento orientado a objetos, a implementação envolve a criação do código-fonte correspondente às classes de objetos do sistema utilizando linguagens de programação como C#, C++, Java etc. Além da codificação desde o início de toda documentação, a implementação pode também reutilizar componentes de software, bibliotecas de classes e frameworks para agilizar a atividade (BEZERRA, 2006).

2.5 Aprovação de codificação

Após os casos de uso serem codificados na fase anterior, é necessário que se realize a aprovação da codificação. Se na fase da avaliação dos casos de uso pela equipe do projeto as revisões de software foram tratadas como uma proposta modelo para realizar a avaliação da documentação, aqui ela deve ser seguida com rigor.

Nesta fase as revisões de software devem ser realizadas pelos codificadores e é extremamente aconselhável realizar reuniões formais como inspeções e *walkthroughs* para a discussão do código e aprovação posteriormente. É importante a participação do gerente de projeto e de um desenvolvedor experiente para aumentar a qualidade do produto gerado nesta fase.

Além disso, diversas atividades de testes são realizadas para verificação do código, levando-se em conta a especificação feita na fase de levantamento de requisitos e de definição dos casos de uso. O principal produto desta etapa é um relatório de testes, com informações sobre erros detectados na construção do software.

2.6 Gerência de configuração de software

Durante o processo de desenvolvimento de um software, é gerada uma grande quantidade de itens de informação que podem ser alterados durante o processo. Para que não haja inconsistência nos itens de informação importantes para o projeto, a criação e as alterações desses itens devem ser acompanhadas e controladas pelo gerente do projeto. O processo de gerência de configuração é um processo de ciclo de vida de software que permite que esse controle seja realizado (FIGUEIREDO *et al.*, 2004).

Gerência de configuração de software é um conjunto de atividades desenvolvidas para administrar modificações ao longo do ciclo de vida do software de computador, pode ser vista também como uma atividade de garantia de qualidade de software, que é aplicada ao longo de todo o processo de software (PRESSMAN, 2006).

Os itens que compreendem toda a informação produzida como parte do processo de software são chamados coletivamente de configuração de software. Segundo PRESSMAN (2006), os itens de configuração de software podem ser divididos em três amplas categorias:

- Programas de computador (tanto na forma de código fonte quanto executável);
- Produtos de trabalho que descrevem programas de computador (voltados tanto para profissionais técnicos quanto para usuários);
- Dados (contidos em um programa ou externos a ele).

Se cada item de configuração, simplesmente levasse outros itens, pouca confusão resultaria. Infelizmente outra variável entra no processo: a modificação. As modificações podem ocorrer em qualquer época, por qualquer motivo, a origem delas é tão variada quanto às próprias modificações. No entanto, há quatro fontes fundamentais de modificação (PRESSMAN, 2006):

- Novas condições do negócio ou do mercado ditam modificações nos requisitos do produto ou nas regras do negócio.
- Novas necessidades do cliente exigem modificação dos dados produzidos por sistemas de informação, da funcionalidade incorporada a produtos ou dos serviços realizados por um sistema baseado em computador.
- Reorganização ou crescimento/diminuição dos negócios causa modificações nas prioridades do projeto ou na estrutura da equipe de engenharia de software.
- Restrições de orçamento ou cronograma causam redefinição do sistema ou produto.

2.7 Elementos de um sistema de Gerência de Configuração

DART (2001) identifica quatro importantes elementos que devem estar presentes quando um sistema de gestão de configuração é desenvolvido:

- Elementos de componente – um conjunto de ferramentas acoplado a um sistema de gestão de arquivos (por exemplo, um banco de dados) que possibilita acesso a gestão de cada item de configuração de software.
- Elementos de processo – uma coleção de procedimentos e tarefas que definem uma abordagem efetiva para a gestão de modificação (e atividades relacionadas) para todas as partes envolvidas na gestão, na engenharia e no uso de software de computador.
- Elementos de construção – um conjunto de ferramentas que automatizam a construção de software garantindo que o conjunto adequado de componentes validados (isto é, versão correta) foi montado.
- Elementos humanos – para implementar a gerência de configuração, a equipe de software utiliza um conjunto de ferramentas e características de processo.

2.8 Itens de configuração de software

Um item de configuração de software é a informação criada como parte do processo de engenharia de software, pode-se considerar também como sendo uma única seção de uma especificação grande ou um caso de teste em uma sequência de testes grande. Com mais realismo, um item de configuração é um documento, toda uma sequência de casos de teste ou um componente de programa que tem nome (um *applet* Java, por exemplo) (PRESSMAN, 2006).

Além dos itens de configuração derivados dos produtos do trabalho de software, muitas organizações de engenharia de software colocam também ferramentas de software sob controle de configuração, isto é, compiladores, navegadores e outras

ferramentas automatizadas são “congeladas” como parte da configuração de software. Essas ferramentas precisam estar disponíveis quando modificações na configuração de software tiverem que ser feitas. Apesar de serem raras, é um possível que uma nova versão de uma ferramenta utilizada produza resultados diferentes que os da versão original. Por esse motivo, as ferramentas podem se tornar referencial como parte de um processo abrangente de gerência de configuração (PRESSMAN, 2006).

2.9 Repositório

O repositório para gerência de configuração de software é o conjunto de mecanismos e estruturas de dados que permite a uma equipe de software gerir modificação de modo efetivo, fornecendo as funções de um sistema de gestão de banco de dados. Para apoiar a gerência de configuração, o repositório precisa ter um conjunto de ferramentas que forneça apoio as seguintes características (PRESSMAN, 2006):

- **Determinação de versão:** A medida que um projeto progride, várias versões de produtos de trabalho individuais são criadas. O repositório precisa ser capaz de salvar todas essas versões para permitir a gestão efetiva das entregas do produto e permitir aos desenvolvedores voltar a versões anteriores durante o tempo da depuração.
- **Acompanhamento dos requisitos:** Essa função fornece a capacidade de acompanhar todos componentes e produtos passíveis de entrega de projeto e construção que resultam de uma especificação de requisitos específica (acompanhamento avante). Além disso, fornece a capacidade de identificar qual requisito gerou um produto de trabalho definido (retro acompanhamento).
- **Acompanhamento de dependência e gerência de modificação:** O repositório gera uma grande variedade de relacionamentos entre os objetos de configuração armazenados nele. Eles incluem relacionamentos

entre entidade e processos da empresa, entre as partes de um projeto de empresa, entre componentes de projeto e a arquitetura da informação da empresa, etc. A capacidade de acompanhar todos esses relacionamentos é essencial para a integridade da informação armazenada no repositório e para a geração de produtos de trabalho baseados nela.

- **Pista de auditoria:** Uma pista de auditoria estabelece informação adicional sobre quando, por que e por quem modificações foram feitas. Informação sobre a fonte das modificações pode ser introduzida como atributos de objetos específicos do repositório.

2.10 Controle de Versão

Controle de versão é uma combinação de procedimentos e ferramentas para gestão de diferentes versões de objetos de configuração (um modelo de análise de especificação de requisitos, por exemplo) que são criados durante o processo de software. Um sistema de controle de versão possui quatro capacidades principais: (1) um repositório que armazena todas as versões de um objeto de configuração; (2) uma capacidade de gestão de versão que armazena todos os objetos de configuração relevantes; (3) um método de construção fácil que permite ao engenheiro de software coletar todos os objetos de configuração relevantes e construir uma versão específica do software. Além disso, o sistema de controle de versão e controle de modificação geralmente implementam uma capacidade de acompanhamento de tópicos que permite a equipe registrar e acompanhar o estado de todos os tópicos importantes associados a cada objeto de configuração (PRESSMAN, 2006).

Segundo MOLINARI (2007), alguns colocam que a principal missão de um sistema de controle de versões é permitir a edição colaborativa e o compartilhamento de dados, outros dizem que o sistema de controle de versões rastreia e controla todos os artefatos do projeto e desse modo consegue coordenar o trabalho paralelo de desenvolvedores.

Em empresas de grande porte, onde existem vários profissionais trabalhando concorrentemente nos mesmos arquivos, faz-se necessário uma política para ordenar e integrar estas alterações, a fim de evitar a perda de um trabalho ocasionada quando uma pessoa sobrescreve o trabalho de outra.

Os sistemas de controle de versão implementam esta política, assegurando que os desenvolvedores não trabalhem diretamente nos arquivos do repositório, mas sim em cópias de trabalho desses arquivos em suas próprias máquinas. Isso permite que um desenvolvedor faça seu trabalho de forma independente e sem interferir no trabalho de outro. Após o término do trabalho, o arquivo deve ser colocado novamente no repositório e o sistema de controle de versão deve fazer uso de políticas de versionamento para que alterações não sejam inadvertidamente sobrepostas.

2.11 Controle de modificação

A realidade do controle de modificação foi descrita no contexto da engenharia de software por BACH (1998): “O controle de modificação é vital, mas as forças que o tornam necessário também o tornam desagradável. Preocupamo-nos com a modificação porque uma pequena perturbação no código pode criar uma grande falha no produto. Mas ela também pode reparar uma falha grande ou proporcionar magníficas capacidades novas. Preocupamo-nos com as modificações porque um único desenvolvedor preguiçoso pode afundar o projeto, no entanto, ideias brilhantes originalmente nas mentes desses preguiçosos e um processo de controle de modificação complicado pode efetivamente desencorajá-los a fazer trabalho criativo”.

Para um grande projeto de engenharia de software, modificações sem controle podem levar rapidamente ao caos. Para esses projetos, o controle de modificação combina procedimentos humanos e ferramentas automatizadas. O processo do controle de modificação é ilustrado na Figura 1.

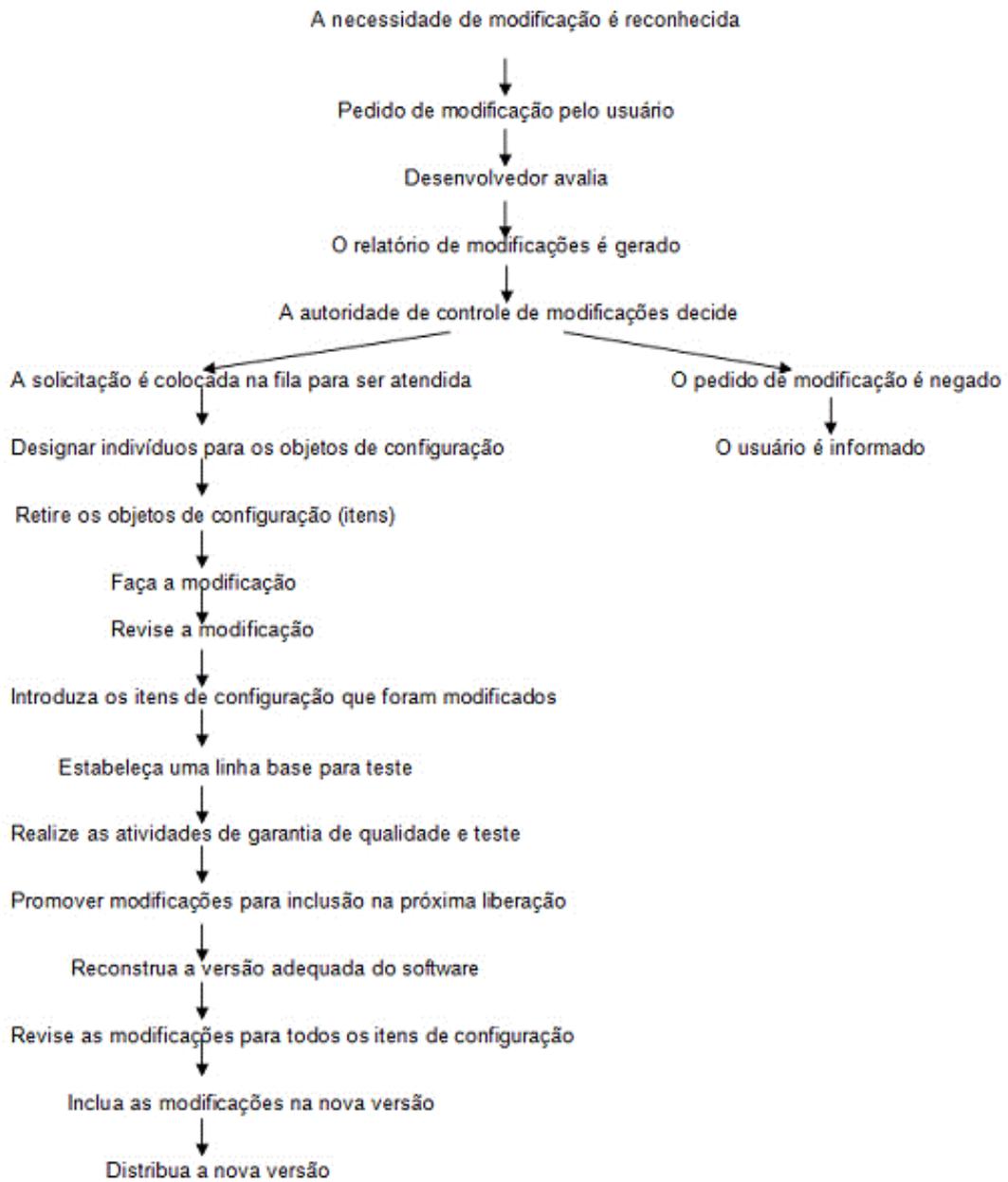


Figura 1: Processo de controle de modificações (Pressman, 2006).

2.12 Wiki

Wiki é um termo havaiano que significa “rápido” e que tem se estabelecido como um eficiente método para a edição colaborativa de textos. Ele foi criado por Ward Cunningham, autor do primeiro *Wiki* disponibilizado na web em 1995 e chamado de

Portland Pattern Repository. Cunningham pretendia desenvolver um site onde os próprios usuários poderiam gerar conteúdo. Com o sucesso do sistema que desenvolveu, vários clones de sua ideia foram surgindo como alternativa para a construção participativa de textos e até mesmo como ferramenta para a gestão do conhecimento em empresas e escolas.

No livro *The Wiki Way*, Leuf e Cunningham (2001) definem um *Wiki* como “uma coleção livremente expansível de páginas Web interligadas num sistema de hipertexto para armazenar e modificar informação – um bando de dados, onde cada página é facilmente editada por qualquer usuário com um browser”. A ideia central do sistema *wiki* é que qualquer texto original possa ser alterado, de modo que novos conhecimentos sejam incorporados aos já existentes, ou seja, em *wikis* abertos, sendo possível a qualquer pessoa editar uma página. Já em *wikis* com acesso restrito para edição, somente usuários cadastrados ou com permissão do administrador podem contribuir com novos conteúdos ou acrescentar um novo texto.

O funcionamento de um sistema *wiki* pode ser dividido em cinco partes (LEUF; CUNNINGHAM, 2001):

- Interface - A interface dos *wikis* é muito intuitiva. A página inicial reúne todos os temas tratados e, uma lista de páginas atualizadas permite o acompanhamento da evolução do *wiki* e o acesso direto às últimas modificações.
- Editor - As páginas são criadas e editadas com um editor de texto. Como com um software do tipo do Word, é possível formatar o seu texto (negrito, itálico, sublinhado, tamanho e seleção de fonte). Também é possível adicionar imagens, tabelas, listas, links para outras páginas do *wiki* e anexar documentos para enriquecer a sua página. Antes de baixar a página, pode-se pré-visualizá-la.
- Histórico - Toda vez que uma página é criada ou alterada, ela é listada no histórico com a data e o nome do autor.
- Direitos de acesso - Certos *wikis* propõem um sistema de gestão dos direitos de acesso, que definem quem tem acesso àquela página e qual ação o usuário está autorizado a efetuar (editar, comentar, ler).

- Busca - Um motor de busca ajuda a encontrar rapidamente uma informação, uma página, uma alteração. A triagem é feita por data de criação, autor, título, data de modificação etc.

O sistema *wiki* privativo possibilita criar um espaço interativo para que ocorra a gestão do conhecimento em uma organização de forma ampla entre os colaboradores. O seu funcionamento pode servir como uma via de acesso para a aprendizagem colaborativa e a integração entre os funcionários de diferentes setores, níveis hierárquicos e segmentos em torno do crescimento coletivo de um determinado assunto, criando soluções simples para problemas complexos através da construção hipertextual colaborativa. A utilização de *wikis* em ambientes que concentram equipes de trabalho pode servir para criar e compartilhar conhecimentos internamente em uma organização criando redes em torno de áreas de interesse em comum.

2.13 Trabalhos relacionados

A ferramenta GConf (FIGUEREDO *et al.*, 2004) foi proposta para apoiar as atividades de gerência de configuração em uma organização. A GConf apoia todas as atividades do processo de gerência de configuração proposto pelo artigo: identificação da configuração, controle da configuração, relato da situação, auditoria da configuração e gerência de liberação e entrega. A ferramenta é disponibilizada num Ambiente de Desenvolvimento de Software Orientado a Organização.

A GConf, assim como o Redmine, é um ferramenta genérica para apoiar a gerência de configuração de qualquer artefato de software e não tem como objetivo principal apoiar diretamente a gerência de casos de uso. Na ferramenta é utilizado um sistema de arquivos onde estão localizados os itens de configuração a serem controlados. A ferramenta não possui sistema de controle de versão como no Redmine nem pode ter o seu fluxo de trabalho configurado de acordo com as necessidades da organização. Também não permite a divulgação dos itens de configuração na forma de hipertexto ou *wiki*.

Em Barbaresco (2000) foi apresentado uma ferramenta de apoio ao processo de gerência de configuração. A ferramenta, que foi implementada em Delphi, não

incorpora de maneira satisfatória todas as funcionalidades exigidas pelos modelos estudados, como, por exemplo, o controle de versões, porém atende aos pedidos de modificações dos itens de configuração, juntamente com histórico e relatórios das alterações dos artefatos.

3 Utilização do Redmine para gerência de casos de uso

Este capítulo aborda a ferramenta que servirá de apoio para o desenvolvimento do trabalho proposto – o Redmine. É proposto também um workflow, com a descrição detalhada das etapas do processo a ser utilizado em projeto de software para a gerência dos casos de uso. Será apresentado, também, como implementar este workflow com o auxílio do Redmine.

3.1 Motivação pelo uso da ferramenta Redmine

Inicialmente, na fase de elaboração do projeto final, foram cogitadas três ferramentas para desenvolver o objetivo do tema: Trac², Jira³ e Redmine. Uma série de pesquisas, estudos e comparações foram realizadas sobre essas três ferramentas para decidir qual seria a mais adequada ao trabalho. Os requisitos para a definição da ferramenta foram: integração com repositórios - como o SVN; integração com *wiki*; controle de tarefas e licença gratuita. Esta seção discute os motivos da escolha do Redmine como a ferramenta para a implementação do trabalho proposto.

O Redmine é um software de gerenciamento de projetos criado em 2006 e desenvolvido na linguagem Ruby on Rails⁴. Sua instalação é relativamente fácil, porém é necessário instalar outros conteúdos, como, por exemplo, o banco de dados MySQL⁵ e o servidor *Apache*⁶. Este caracteriza-se por ser uma ferramenta para gerência de projetos e gerência de tickets, integrada com diferentes sistemas de controle de versões, multilíngua e altamente personalizável. Para a gestão de utilizadores deve ser destacado o controle flexível dos seus papéis, ao poder criar grupos onde é possível customizar os acessos, suporte para autenticação LDAP e auto-registro no sistema. Abaixo segue algumas das vantagens do Redmine que atende ao que é proposto pelo trabalho:

²<http://trac.edgewall.org/>

³<http://www.atlassian.com/software/jira/overview>

⁴<http://rubyonrails.org/>

⁵<http://www.mysql.com/>

⁶<http://www.apache.org/>

- Suporte a múltiplos projetos;
- Controle de tarefas;
- Calendários e gráficos;
- Notícias, documentos e arquivos (gerenciamento);
- Wiki e fórum;
- Integração com repositórios (SVN, CVS, Git, Mercurial, Bazaar e Darcs);
- Multilinguagem;
- Suporte a diversos bancos;
- Controle de tempo nas tarefas;
- Uso de plugins.
- Licença de uso e de distribuição livre;
- Atende à maioria dos requisitos técnicos de gerência de projetos;
- Instalação simples;
- Interface simples e agradável; e
- Equipe ativa de desenvolvimento e de melhoria do produto.

Algumas exigências foram estabelecidas para a escolha da ferramenta. Deveriam conter integração com o SVN, já que o caso de uso deve ser versionado, e também integração com *wiki* para a descrição dos casos de uso. Todas as três atendiam tais requisitos. Apesar de possuírem funcionalidades semelhantes entre elas, o Redmine se destacava por ser uma ferramenta gratuita e muito usada em trabalhos acadêmicos, sendo assim mais fácil encontrar dados, opiniões e suporte.

Outro fator positivo foi a grande disponibilidade de plugins. Diversas atualizações são lançadas constantemente, corrigindo bugs e adicionando funcionalidades essenciais, algumas até percebidas e sugeridas pelos próprios usuários. Isso se deve pelo fato de possuir uma comunidade bastante ativa, já que sua distribuição é gratuita. A Tabela 3.1 mostra uma comparação técnica entre as ferramentas estudadas:

Sistema	Criador	Licença	Linguagem de implementação	Banco de dados
Jira	Atlassian	Proprietária, Gratuita para uso não comercial	Java	MySQL, PostgreSQL, Oracle, SQL Server
Redmine	Jean-Philippe Lang	GPLv2	RubyonRails	MySQL, PostgreSQL, SQLite
Trac	Edgewall Software	New BSD	Python	SQLite, PostgreSQL, MySQL

Tabela 3-1: Comparativo técnico entre as ferramentas [Wikipédia, 2012]

O Trac é uma ferramenta semelhante ao Redmine com propósitos em comum, pode ser utilizado tanto para desenvolvimento como manutenção e não apenas gerencia o ciclo de vida das solicitações como também se integra ao Subversion (SVN), tem um *wiki* associado onde os casos de uso podem ser documentados e alguns plugins que ajudam tornar o Trac uma ferramenta útil e simples de usar. Sua instalação não é simples. O usuário deve ter conhecimento em *shell script* e como configurar o servidor *Apache*. Por possuir muitas dependências, o processo de instalação pode se tornar muito trabalhoso, especialmente no Windows já que ele possui algumas dependências que tem origem em sistemas UNIX.

O Jira é uma ferramenta utilizada para gestão de projetos e defeitos. É altamente flexível e se adapta a diversas necessidades. Sua proposta é facilitar o desenvolvimento de aplicações servindo como uma plataforma. Assim como foi sugerido na monografia, o Jira também é capaz de criar *workflow* configurável, possibilitando a gerência de processos. Outra vantagem é a possibilidade de reaproveitamento de recursos que demandariam muito esforço para criação, como, por exemplo, mecanismo de autenticação do usuário. Ao contrário dos outros dois sistemas analisados, sua licença não é gratuita.

De todas as ferramentas estudadas, o Redmine é o que mais se aproxima do que foi estabelecido para o tema da monografia, além de ser a ferramenta mais completa, como pode ter ser observado na Tabela 3.2.

O foco deste trabalho é na gerência de configuração dos casos de uso. Portanto, esta seção não destacará fatores técnicos ou aprofundar discussões sobre as

ferramentas estudadas, limitando-se a explicar como o Redmine apoiará o desenvolvimento deste trabalho.

	Redmine	Trac	Jira
Licença de uso e distribuição livre	SIM	SIM	NÃO
Atender à maioria dos requisitos técnicos de gerência de projetos	SIM	SIM	SIM
Instalação simples	SIM	NÃO	SIM
Incluir recursos como <i>wiki</i> e integração com SVN	SIM	SIM	SIM
Interface simples e intuitiva	SIM	SIM	SIM
Equipe e comunidade ativa/disponibilização de plugins	SIM	SIM	SIM

Tabela 3-2: Comparativo geral entre as ferramentas

Com o apoio dos recursos necessários para a conclusão do projeto – como *wiki*, gerência de configuração etc. – o Redmine consegue atender o que foi proposto pelo tema, rastreando e disponibilizando os casos de uso em seu repositório, reunindo os elementos necessários para definir o *workflow* do projeto.

3.2 Processo de apoio para gerência de casos de uso

Para uma adequada gerência dos casos de uso é necessária a implementação de um workflow que controle seu ciclo de vida desde o levantamento de requisitos até a conclusão da codificação, passando pela aprovação interna e pela aprovação do cliente.

O workflow será implementado pela ferramenta Redmine, seguindo um processo planejado e estabelecido previamente. O objetivo desta seção é descrever as etapas deste processo que será apoiado pela ferramenta. A Figura 2 apresenta o esquema do processo definido.

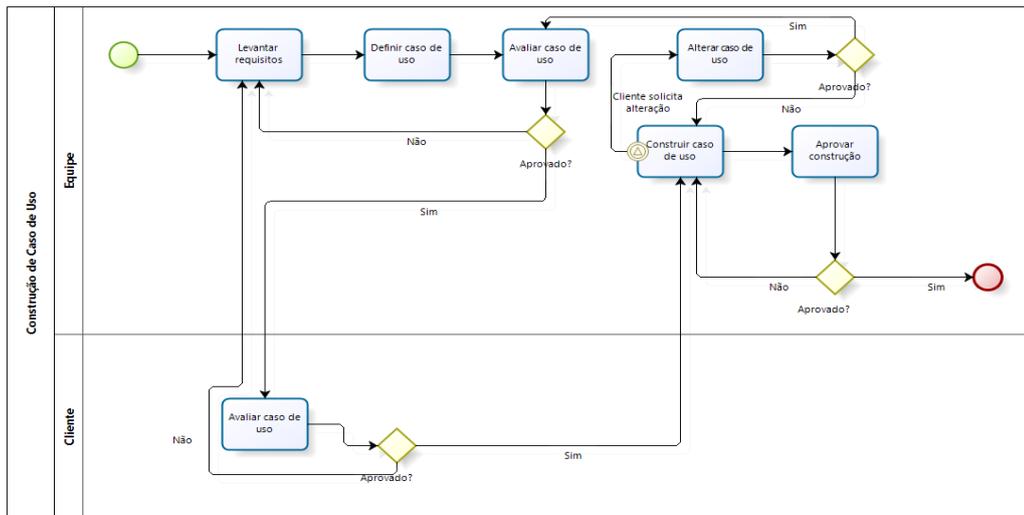


Figura 2: Processo de apoio para gerência de casos de uso

Cada atividade pressupõe a alteração do estado da tarefa criada para controlar a gerência do ciclo de vida dos casos de uso. Dessa forma, os estados da tarefa deverão seguir as transições presentes no diagrama de estados que pode ser visualizado na Figura 3.

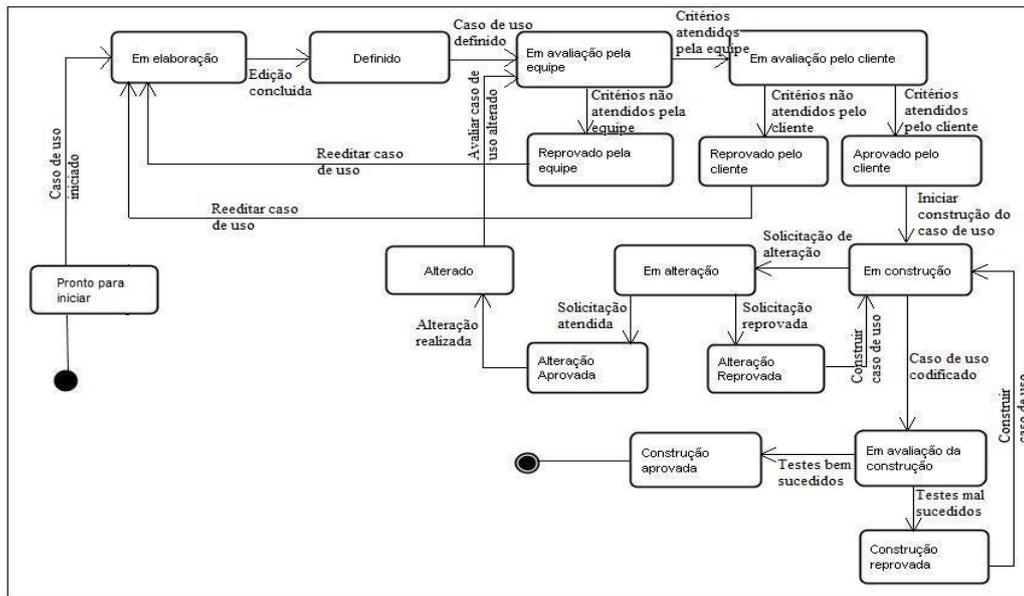


Figura 3: Diagrama de estados

3.2.1 Levantar requisitos

A primeira etapa do processo tem foco no levantamento de requisitos, que serão definidos para o prosseguimento do *workflow*. Apenas após esta atividade ser

concluída é recomendado seguir para a atividade seguinte de definição dos casos de uso. Os requisitos informados serão utilizados também nas atividades posteriores. Com essas informações, é possível atingir um entendimento mais claro em relação aos objetivos do escopo do projeto.

O primeiro estado da tarefa (caso de uso) foi definido como “Pronto para iniciar”, equivalente à abertura de um novo ticket na ferramenta Redmine. Esse estado significa que o caso de uso pode começar a ser editado e seus requisitos começarem a ser levantados. Em seguida, o estado sucessor é denominado “Em elaboração” e corresponde à edição de fato do caso de uso a partir dos requisitos da fase anterior, no caso da ferramenta, criado como uma tarefa. É neste estado que os analistas editam e modificam o caso de uso. Feito isso o estado passa para “Definido”.

3.2.2 Definir casos de uso

Nesta etapa são definidos os casos de uso que serão utilizados. Os requisitos serão disponibilizados de forma que sejam visíveis a todos os participantes do processo. O Redmine controla um repositório específico para documentos e imagens sendo possível armazenar os casos de uso criados nesta atividade.

É também nesta atividade que ocorre a transformação dos requisitos elaborados na atividade anterior de levantamento de requisitos, em casos de uso. É a fase mais crítica para o analista, pois um erro de comunicação e entendimento pode gerar retrabalho e comprometimento de todo o processo nas atividades posteriores.

Esta atividade é representada no *workflow* do Redmine como “Em elaboração” - onde os analistas têm total liberdade para editar os casos de uso baseado nos requisitos propostos - e “Definido” – quando os analistas indicam que terminaram o trabalho de edição e criação dos casos de uso.

3.2.3 Avaliar casos de uso pela equipe do projeto

Inicialmente devem-se definir os papéis da equipe do projeto. No processo que foi estipulado, é nesta fase que ocorre a avaliação dos casos de uso pela equipe do projeto, que são os analistas, codificadores e outros projetistas. A equipe deve

discutir e analisar os documentos, além de contribuir com sugestões e críticas. O mecanismo a ser utilizado é o de revisões de software.

Depois de terminada a edição do caso de uso e, finalmente, definida sua construção, no estado seguinte “Em avaliação pela equipe”, o analista deverá aprovar ou reprovar o caso de uso em questão. Consequentemente, deste estado pode-se seguir para o estado “Reprovado pela equipe” ou “Aprovado pela equipe”. Além disso, neste estado, deverá ser indicada a apreciação dos critérios de avaliação do caso de uso. Os analistas deverão preencher os critérios avaliando o caso de uso e, então, responder de acordo com o estado seguinte (“Aprovado” ou “Reprovado”). Caso tenha sido reprovado, ele retorna para o estado “em elaboração” para que os analistas possam voltar a editar a tarefa, baseando-se nas respostas do critério de avaliação e nos comentários que a equipe de projeto disponibilizou na tarefa, justificando sua rejeição. Os critérios de avaliação de casos de uso a serem utilizados pela equipe de projeto foram baseados no Guia de Implementação do MR-MPS-SW (SOFTEX, 2011) e são apresentados na Tabela 3.3.

Clareza		
CLA.01 - O objetivo do caso de uso é compreensível?	Sim <input type="checkbox"/>	Não <input type="checkbox"/>
CLA.02 - O caso de uso está descrito com um nível de detalhes suficiente para o entendimento?	Sim <input type="checkbox"/>	Não <input type="checkbox"/>
Completeza		
COM.01 - O caso de uso possui fluxo principal especificado	Sim <input type="checkbox"/>	Não <input type="checkbox"/>
COM.02 - O caso de uso possui fluxo alternativo especificado	Sim <input type="checkbox"/>	Não <input type="checkbox"/>
Consistência		
CON.01 - Os passos dos fluxos do caso de uso estão consistentes com o desejado?	Sim <input type="checkbox"/>	Não <input type="checkbox"/>
Ambiguidade		
AMB.01 - Os requisitos estão escritos de forma não ambígua?	Sim <input type="checkbox"/>	Não <input type="checkbox"/>
Testabilidade		
TES.01 - O caso de uso é testável?	Sim <input type="checkbox"/>	Não <input type="checkbox"/>

Tabela 3-3: Critérios de avaliação dos casos de uso pela equipe de projeto

3.2.4 Avaliar casos de uso pelo cliente

É nesta atividade que será realizada a avaliação do cliente. Após ter os requisitos e os casos de uso levantados e consolidados é necessária a aprovação dos

patrocinadores (*stakeholders*) e dos clientes. Todas as partes interessadas no processo devem aprovar e entender os requisitos. Caso seja necessária alguma alteração, deverá ser realizada nesta etapa. Com isso é possível evitar as mudanças posteriores, evitando, assim, a volatilidade de requisitos.

Após a aprovação da equipe de projeto, o novo estado do caso de uso é “Em avaliação pelo cliente”. Assim como no estado “Em avaliação pela equipe”, deverão ser respondidos os critérios de avaliação propostos. De forma semelhante ao estado anterior, os possíveis estados sucessores são “Aprovado pelo cliente” e “Reprovado pelo cliente”. Caso tenha sido reprovado pelo cliente, o estado volta a ser “Em elaboração” e deve conter as respostas dos critérios justificando a rejeição do cliente. Se for aprovado, segue para o estado “Em construção”, onde o caso de uso será codificado.

Os critérios de avaliação também foram adaptados do Guia de Implementação do MR-MPS-SW (SOFTEX, 2011). O cliente deve responder se o caso de uso está aprovado quanto à adequabilidade, ou seja, se está adequado às suas necessidades. Enquanto que a equipe de projeto deverá responder se o caso de uso é testável, avaliar a sua testabilidade. Os critérios para avaliação do cliente são descritos na Tabela 3.4 a seguir.

Clareza		
CLA.01 - O objetivo do caso de uso é compreensível?	Sim <input type="checkbox"/>	Não <input type="checkbox"/>
CLA.02 - O caso de uso está descrito com um nível de detalhes suficiente para o entendimento?	Sim <input type="checkbox"/>	Não <input type="checkbox"/>
Adequabilidade		
ADQ.01 - O caso de uso está adequado às necessidades do cliente?	Sim <input type="checkbox"/>	Não <input type="checkbox"/>
Completeza		
COM.01 - O caso de uso possui fluxo principal especificado	Sim <input type="checkbox"/>	Não <input type="checkbox"/>
COM.02 - O caso de uso possui fluxo alternativo especificado	Sim <input type="checkbox"/>	Não <input type="checkbox"/>
Consistência		
CON.01 - Os passos dos fluxos do caso de uso estão consistentes com o desejado?	Sim <input type="checkbox"/>	Não <input type="checkbox"/>
Ambiguidade		
AMB.01 - Os requisitos estão escritos de forma não ambígua?	Sim <input type="checkbox"/>	Não <input type="checkbox"/>

Tabela 3-4: Critérios de avaliação dos casos de uso pelo cliente

3.2.5 Construir caso de uso

Nesta atividade os casos de uso devem ser codificados.

Com o caso de uso codificado, o processo segue, do estado “Em construção”, para dois possíveis estados: “Em avaliação da construção”, onde os desenvolvedores testarão o código e, assim, podem aprová-lo ou reprová-lo, e o estado “Em alteração”, onde o cliente pode solicitar alguma possível alteração ou algum novo requisito. Esta alteração deve ser aprovada pela equipe, que analisará se haverá viabilidade de realizá-la ou não. Como resultado, há dois estados seguintes possíveis: “Alteração aprovada” ou “Alteração reprovada”. Caso a alteração tenha sido reprovada, o fluxo volta para o estado “Em construção”. Já se tiver sido aprovada, o fluxo segue para um novo estado “Alterado”, indicando que a alteração foi realizada. Realizada a alteração, o fluxo retorna para o estado “Em avaliação pela equipe” onde o analista analisará a modificação e, assim, seguirá para os próximos estados do processo regularmente.

Como não é objetivo da monografia tratar de assuntos sobre projeto (*design*) de software e codificação, não é necessário o aprofundamento deste assunto nesta atividade. Deve-se deixar claro, no entanto, que esta é a fase de transição da documentação dos casos de uso para a codificação executável, devendo ser realizados testes de verificação dos códigos, conforme pertinente.

3.2.6 Aprovar construção

O processo termina com a aprovação da codificação do caso de uso, representado pelo estado “Construção aprovada”. Caso tenha sido reprovado, o caso de uso com o estado “Construção reprovada” retorna para “Em construção” e os desenvolvedores corrigem os possíveis erros até chegar ao estado em que ela seja finalmente aprovada. Após a atividade de testes e a aprovação do código, os diversos módulos do sistema são integrados, resultando finalmente no produto do software.

3.2.7 Alterar casos de uso

Há ainda a atividade “Em alteração”, onde o cliente solicita uma mudança de requisito durante o processo. Nesta atividade, essa mudança é avaliada pela equipe

de projeto e decidida se será viabilizada ou não. No *workflow*, caso a mudança seja aprovada, o estado que se sucede é “Alteração aprovada”, seguindo para o estado “Alterado” indicando que a alteração foi realizada e, posteriormente, o fluxo alternativo volta para o fluxo principal, partindo do estado “Alterado” para o estado “Em avaliação pela equipe”. De volta ao fluxo principal, os estados seguem regularmente, dando prosseguimento à alteração realizada.

Caso a alteração seja reprovada, o estado passa para “Alteração reprovada” e volta para o estado “Em construção”, onde o desenvolvedor retomará a construção do caso de uso, descartando a mudança solicitada pelo cliente.

3.3 Configuração do Redmine

O objetivo desta seção é descrever como o Redmine foi configurado para apoiar o processo apresentado na Seção 3.2.

Para a elaboração do processo da Seção 3.2, foram criados novos estados das tarefas. A Figura 4 exibe a criação pelo Redmine dos novos estados da tarefa que serão utilizados no fluxo em questão.

Estados da tarefa + Novo estado

Estado	Valor por omissão	Tarefa fechada	Ordenar	
Nova			▲ ▲ ▼ ▽	🗑 Apagar
Em andamento			▲ ▲ ▼ ▽	🗑 Apagar
Resolvida			▲ ▲ ▼ ▽	🗑 Apagar
Feedback			▲ ▲ ▼ ▽	🗑 Apagar
Fechada		✓	▲ ▲ ▼ ▽	🗑 Apagar
Reprovado pela equipe			▲ ▲ ▼ ▽	🗑 Apagar
Pronto para Iniciar	✓		▲ ▲ ▼ ▽	🗑 Apagar
Em elaboração			▲ ▲ ▼ ▽	🗑 Apagar
Definido			▲ ▲ ▼ ▽	🗑 Apagar
Em avaliação pela equipe			▲ ▲ ▼ ▽	🗑 Apagar
Aprovado pela equipe			▲ ▲ ▼ ▽	🗑 Apagar
Em avaliação pelo cliente			▲ ▲ ▼ ▽	🗑 Apagar
Aprovado pelo cliente			▲ ▲ ▼ ▽	🗑 Apagar
Reprovado pelo cliente			▲ ▲ ▼ ▽	🗑 Apagar
Em construção			▲ ▲ ▼ ▽	🗑 Apagar
Em avaliação da construção			▲ ▲ ▼ ▽	🗑 Apagar
Em alteração			▲ ▲ ▼ ▽	🗑 Apagar
Construção aprovada		✓	▲ ▲ ▼ ▽	🗑 Apagar
Construção reprovada			▲ ▲ ▼ ▽	🗑 Apagar
Alteração Aprovada			▲ ▲ ▼ ▽	🗑 Apagar
Alteração Reprovada			▲ ▲ ▼ ▽	🗑 Apagar
Alterado			▲ ▲ ▼ ▽	🗑 Apagar

Figura 4: Tela da criação de novos estados

Para elaborar o caminho do fluxo de estado das atividades, foi necessária a criação dos novos estados, como visualizado na Figura 4, e a definição de um fluxo de trabalho para os novos estados da tarefa. Este fluxo de trabalho mostra quais os possíveis novos estados disponíveis a partir de um determinado estado atual. A criação do fluxo de trabalho na ferramenta Redmine pode ser visualizada na Figura 5.

3.4 Exemplo do uso do workflow na ferramenta Redmine

Para demonstrar o fluxo de trabalho citado neste capítulo, foi utilizado como exemplo um caso de uso de cadastro de cliente de um projeto de uma oficina mecânica, aprovado por todas as partes envolvidas, seguindo os seus estados associados à sua condição. Também foi utilizado um exemplo em que explicita a gerência de configuração, onde o caso de uso é reprovado por uma das partes e é alterado posteriormente.

3.4.1 Fluxo de sucesso

Neste exemplo, o caso de uso é aprovado por todas as partes até a aprovação da sua codificação. A seguir é apresentado o fluxo principal sem nenhuma interferência ao longo do processo.

A Figura 6 apresenta a criação de uma tarefa com a descrição do caso de uso de exemplo, conforme descrita no Redmine. A transição de estados do Redmine guiará o processo.

Função: Tipo: Só exibir estados empregues por este tipo de tarefa

Estado actual	Novos estados permitidos																	
	Fechada	Reprovado pela equipe	Pronto para Iniciar	Em elaboração	Definido	Em avaliação pela equipe	Aprovado pela equipe	Em avaliação pelo cliente	Aprovado pelo cliente	Reprovado pelo cliente	Em construção	Em avaliação da construção	Em alteração	Construção aprovada	Construção reprovada	Alteração Aprovada	Alteração Reprovada	Alterado
✓ Fechada	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
✓ Reprovado pela equipe	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
✓ Pronto para Iniciar	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
✓ Em elaboração	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
✓ Definido	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
✓ Em avaliação pela equipe	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
✓ Aprovado pela equipe	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
✓ Em avaliação pelo cliente	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
✓ Aprovado pelo cliente	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
✓ Reprovado pelo cliente	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
✓ Em construção	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
✓ Em avaliação da construção	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
✓ Em alteração	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
✓ Construção aprovada	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
✓ Construção reprovada	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
✓ Alteração Aprovada	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
✓ Alteração Reprovada	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
✓ Alterado	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

▶ Additional transitions allowed when the user is the author
 ▶ Additional transitions allowed when the user is the assignee

Figura 5: Criação do workflow no Redmine

Caso de Uso #5

Cadastrar cliente			
Adicionado por Diogo Admin 16 dias atrás.			
Situação:	Pronto para Iniciar	Início:	23/10/2012
Prioridade:	Normal	Data prevista:	
Atribuído para:	Diogo Admin	% Terminado:	<input type="text"/> 0%
Categoria:	-	Tempo gasto:	-
Versão:	-		
Descrição			
Atendente registra dados do cliente			
Básico:			
1) O caso de uso inicia quando atendente registra entrada de um veículo de um cliente não cadastrado.			
2) Sistema exibe tela de cadastro de cliente.			
3) Atendente entra com dados do cliente.			
4) Sistema registra dados.			
5) Caso de uso termina.			

Figura 6: Tela de criação de caso de uso, estado “Pronto para iniciar”

Após o estado inicial, o caso de uso é atualizado para “Em elaboração”, onde o usuário editará o caso de uso, demonstrado na Figura 7:

Caso de Uso #5

Cadastrar cliente			
Adicionado por Diogo Admin 16 dias atrás. Atualizado 1 minuto atrás.			
Situação:	Em elaboração	Início:	23/10/2012
Prioridade:	Normal	Data prevista:	
Atribuído para:	Diogo Admin	% Terminado:	<input type="text"/> 0%
Categoria:	-	Tempo gasto:	-
Versão:	-		
Descrição			
Atendente registra dados do cliente			
Básico:			
1) O caso de uso inicia quando atendente registra entrada de um veículo de um cliente não cadastrado.			
2) Sistema exibe tela de cadastro de cliente.			
3) Atendente entra com dados do cliente.			
4) Sistema registra dados.			
5) Caso de uso termina.			
Histórico			
Atualizado por Diogo Admin há 1 minuto #1			
• Situação alterado de Pronto para Iniciar para Em elaboração			

Figura 7: Tela do estado “Em elaboração”

Seguindo o fluxo, após a edição do caso de uso o usuário indica que a edição foi concluída. A Figura 8 apresenta o estado seguinte “Definido”:

Caso de Uso #5

Cadastrar cliente

Adicionado por Diogo Admin 16 dias atrás. Atualizado 2 minutos atrás.

Situação:	Definido	Início:	23/10/2012
Prioridade:	Normal	Data prevista:	
Atribuído para:	Diogo Admin	% Terminado:	<input type="text"/> 0%
Categoria:	-	Tempo gasto:	-
Versão:	-		

Descrição

Atendente registra dados do cliente

Básico:

- 1) O caso de uso inicia quando atendente registra entrada de um veículo de um cliente não cadastrado.
- 2) Sistema exibe tela de cadastro de cliente.
- 3) Atendente entra com dados do cliente.
- 4) Sistema registra dados.
- 5) Caso de uso termina.

Histórico

Atualizado por Diogo Admin há 18 minutos #1

- Situação alterado de Pronto para Iniciar para Em elaboração

Atualizado por Diogo Admin há 2 minutos #2

- Situação alterado de Em elaboração para Definido

Figura 8: Tela do estado “Definido”

Após o estado definido, há a indicação de que o caso de uso está pronto, sua edição foi concluída e todos seus requisitos foram levantados e disponibilizados. Após o término desta etapa, o caso de uso produzido é disponibilizado manualmente pelo usuário no *wiki*, como pode ser observado na Figura 9.

Visão geral | Atividade | Tarefas | Nova tarefa | Gantt | Calendário | Notícias | Documentos | Wiki | Ficheiros | Configurações

[Editar](#) [Observar](#) [Bloquear](#) [Renomear](#) [Apagar](#) [Histórico](#)

Cadastrar cliente

Descrição: atendente registra dados do cliente
Ato: Atendente
Pré-requisito: só será necessário o cadastro do cliente se o mesmo não estiver cadastrado no sistema.

Básico:

- 1) o caso de uso inicia quando atendente registra entrada de um veículo de um cliente não cadastrado.
- 2) Sistema exibe tela de cadastro de cliente
- 3) Atendente entra com dados do cliente
- 4) Sistema registra dados
- 5) Caso de uso termina

Regra de negócios:
[RN.1]- [RN1] Dados do cliente: nome completo, endereço, telefone, cpf, rg
[RN2] Dados do carro: placa, cor, modelo, marca, ano

Pós condições: Cliente foi cadastrado no sistema.

[Novo ficheiro](#)

Também disponível em: PDF | HTML | TXT

Figura 9: Caso de uso inserido na wiki

A seguir, a Figura 10, o estado “Em avaliação pela equipe” contendo os critérios específicos para o preenchimento da equipe e sua avaliação:

Caso de Uso #5

Cadastrar cliente

Adicionado por Diogo Admin 16 dias atrás. Atualizado 3 minutos atrás.

Estado:	Em avaliação pela equipe	Data início:	24/02/2013
Prioridade:	Normal	Data fim:	
Atribuído a:	Diogo Admin	% Completo:	0%
Categoria:	-	Tempo gasto:	-
Versão:	-	Ambiguidade?:	Sim
Completeza?:	Sim	Consistência?:	Sim
Testabilidade?:	Sim		
Clareza?:	Sim		

Descrição

Atendente registra dados do cliente

Básico:
1) O caso de uso inicia quando atendente registra entrada de um veículo de um cliente não cadastrado.
2) Sistema exibe tela de cadastro de cliente.
3) Atendente entra com dados do cliente.
4) Sistema registra dados.
5) Caso de uso termina.

Histórico

Atualizado por Diogo Admin há 31 minutos #1

- Situação alterado de Pronto para Iniciar para Em elaboração

Figura 10: Tela do estado “Em avaliação pela equipe”

Na etapa seguinte, a equipe responde os critérios preenchendo se o caso de uso é testável: estado “Aprovado pela equipe”:

Caso de Uso #5

Cadastrar cliente

Adicionado por Diogo Admin 16 dias atrás. Atualizado 3 minutos atrás.

Estado:	Aprovado pela equipe	Data início:	24/02/2013
Prioridade:	Normal	Data fim:	
Atribuído a:	Diogo Admin	% Completo:	0%
Categoria:	-	Tempo gasto:	-
Versão:	-	Ambiguidade?:	Sim
Completeza?:	Sim	Consistência?:	Sim
Testabilidade?:	Sim		
Clareza?:	Sim		

Descrição

Atendente registra dados do cliente

Básico:
1) O caso de uso inicia quando atendente registra entrada de um veículo de um cliente não cadastrado.
2) Sistema exibe tela de cadastro de cliente.
3) Atendente entra com dados do cliente.
4) Sistema registra dados.
5) Caso de uso termina.

Histórico

Atualizado por Diogo Admin há 31 minutos #1

- Situação alterado de Pronto para Iniciar para Em elaboração

Figura 11: Tela do estado “Aprovado pela equipe”

O fluxo segue para o estado sucessor, “Em aprovação pelo cliente”. Neste caso também é respondido os critérios de avaliação do caso de uso específico para o

cliente. Feito isso, o cliente confirma o campo adequabilidade, afirmando que o caso de uso atende os requisitos do cliente, como demonstrado na Figura 12.

Semelhante ao estado “Aprovado pela equipe”, o estado seguinte “Aprovado pelo cliente” na Figura 13, também contém os critérios preenchidos pelo cliente.

Após o cliente aprovar o caso de uso, ele indica que o mesmo poderá começar a ser construído. O estado seguinte da aprovação do cliente é o estado “Em construção”, Figura 14:

Caso de Uso #5

Cadastrar cliente

Adicionado por Diogo Admin 16 dias atrás. Atualizado 3 minutos atrás.

Estado:	Em avaliação pelo cliente	Data início:	24/02/2013
Prioridade:	Normal	Data fim:	
Atribuído a:	Diogo Admin	% Completo:	0%
Categoria:	-	Tempo gasto:	-
Versão:	-	Ambiguidade?:	Sim
Completeza?:	Sim	Consistência?:	Sim
Testabilidade?:	Sim	Adequabilidade?:	Sim
Clareza?:	Sim		

Descrição

Atendente registra dados do cliente

Básico:
1) O caso de uso inicia quando atendente registra entrada de um veículo de um cliente não cadastrado.
2) Sistema exibe tela de cadastro de cliente.
3) Atendente entra com dados do cliente.
4) Sistema registra dados.
5) Caso de uso termina.

Histórico

Atualizado por Diogo Admin há 31 minutos #1

- Situação alterado de Pronto para Iniciar para Em elaboração

Figura 12: Tela do estado “Em avaliação pelo cliente”

Caso de Uso #5

Cadastrar cliente

Adicionado por Diogo Admin 16 dias atrás. Atualizado 3 minutos atrás.

Estado:	Aprovado pelo cliente	Data início:	24/02/2013
Prioridade:	Normal	Data fim:	
Atribuído a:	Diogo Admin	% Completo:	0%
Categoria:	-	Tempo gasto:	-
Versão:	-	Ambiguidade?:	Sim
Completeza?:	Sim	Consistência?:	Sim
Testabilidade?:	Sim	Adequabilidade?:	Sim
Clareza?:	Sim		

Descrição

Atendente registra dados do cliente

Básico:
1) O caso de uso inicia quando atendente registra entrada de um veículo de um cliente não cadastrado.
2) Sistema exibe tela de cadastro de cliente.
3) Atendente entra com dados do cliente.
4) Sistema registra dados.
5) Caso de uso termina.

Histórico

Atualizado por Diogo Admin há 31 minutos #1

- Situação alterado de Pronto para Iniciar para Em elaboração

Figura 13: Tela do estado “Aprovado pelo cliente”

Caso de Uso #5

Cadastrar cliente

Adicionado por Diogo Admin 16 dias atrás. Atualizado 3 minutos atrás.

Estado:	Em construção	Data início:	24/02/2013
Prioridade:	Normal	Data fim:	
Atribuído a:	Diogo Admin	% Completo:	0%
Categoria:	-	Tempo gasto:	-
Versão:	-	Ambiguidade?:	Sim
Completeza?:	Sim	Consistência?:	Sim
Testabilidade?:	Sim	Adequabilidade?:	Sim
Clareza?:	Sim		

Descrição

Atendente registra dados do cliente

Básico:
1) O caso de uso inicia quando atendente registra entrada de um veículo de um cliente não cadastrado.
2) Sistema exibe tela de cadastro de cliente.
3) Atendente entra com dados do cliente.
4) Sistema registra dados.
5) Caso de uso termina.

Histórico

Atualizado por Diogo Admin há 31 minutos #1

- Situação alterado de Pronto para Iniciar para Em elaboração

Figura 14: Tela do estado “Em construção”

O fluxo segue para o estado de “Em avaliação da construção”, a fim de realizar os testes necessários:

Caso de Uso #5

Cadastrar cliente

Adicionado por Diogo Admin 16 dias atrás. Atualizado 3 minutos atrás.

Estado:	Em avaliação da construção	Data início:	24/02/2013
Prioridade:	Normal	Data fim:	
Atribuído a:	Diogo Admin	% Completo:	0%
Categoria:	-	Tempo gasto:	-
Versão:	-	Ambiguidade?:	Sim
Completeza?:	Sim	Consistência?:	Sim
Testabilidade?:	Sim	Adequabilidade?:	Sim
Clareza?:	Sim		

Descrição

Atendente registra dados do cliente

Básico:
1) O caso de uso inicia quando atendente registra entrada de um veículo de um cliente não cadastrado.
2) Sistema exibe tela de cadastro de cliente.
3) Atendente entra com dados do cliente.
4) Sistema registra dados.
5) Caso de uso termina.

Histórico

Atualizado por Diogo Admin há 31 minutos #1

- Situação alterado de Pronto para Iniciar para Em elaboração

Figura 15: Tela do estado “Em avaliação da construção”

No estado “Construção aprovada”, a tarefa pode finalmente ser fechada e, assim, terminar o fluxo de trabalho de um caso de uso aprovado por todas as partes envolvidas.

Caso de Uso #5

Cadastrar cliente

Adicionado por Diogo Admin 16 dias atrás. Atualizado 3 minutos atrás.

Estado:	Construção aprovada	Data início:	24/02/2013
Prioridade:	Normal	Data fim:	
Atribuído a:	Diogo Admin	% Completo:	0%
Categoria:	-	Tempo gasto:	-
Versão:	-	Ambiguidade?:	Sim
Completeza?:	Sim	Consistência?:	Sim
Testabilidade?:	Sim	Adequabilidade?:	Sim
Clareza?:	Sim		

Descrição

Atendente registra dados do cliente

Básico:
1) O caso de uso inicia quando atendente registra entrada de um veículo de um cliente não cadastrado.
2) Sistema exibe tela de cadastro de cliente.
3) Atendente entra com dados do cliente.
4) Sistema registra dados.
5) Caso de uso termina.

Histórico

Atualizado por Diogo Admin há 31 minutos #1

- Situação alterado de Pronto para Iniciar para Em elaboração

Figura 16: Tela do estado “Construção aprovada”

Este foi um exemplo de um fluxo de sucesso de utilização da ferramenta. Existem diversos caminhos que o fluxo pode seguir dependendo da ocorrência dos fatos relacionados ao caso de uso. É necessária a compreensão e o apoio de todas as partes envolvidas para documentar, avaliar e atualizar os casos de uso que estiverem sendo tratados. Desta forma, a ferramenta e o fluxo de trabalho sugerido, são uma alternativa para organizações e seus respectivos processos.

3.4.2 Fluxo alternativo

Neste exemplo, o caso de uso não é testável e portanto é reprovado pela equipe de projeto. O exemplo a ser utilizado será o mesmo do caso de uso de sucesso: cadastro de cliente em um sistema para uma oficina mecânica. Neste exemplo será possível verificar a presença da gerência de configuração no momento

em que o caso de uso é reprovado pela equipe de projeto. O fluxo se mantém idêntico até a aprovação da equipe. A Figura 17 cria uma tarefa idêntica ao fluxo de sucesso do exemplo anterior:

The screenshot shows a task creation interface for 'Cadastrar cliente'. The task was added by 'desenvolvedor' 1 minute ago. The status is 'Pronto para Iniciar', which is circled in red. Other details include a priority of 'Normal', a start date of '06/02/2013', and a progress bar at 0%. The description section contains a list of steps: 1) o caso de uso inicia quando atendente registra entrada de um veículo de um cliente não cadastrado, 2) Sistema exibe tela de cadastro de cliente, 3) Atendente entra com dados do cliente, 4) Sistema registra dados, 5) Caso de uso termina.

Situação:	Pronto para Iniciar	Início:	06/02/2013
Prioridade:	Normal	Data prevista:	
Atribuído para:	-	% Terminado:	0%
Categoria:	-	Tempo gasto:	-
Versão:	-		

Descrição

Atendente registra dados do cliente.

Básico:

- 1) o caso de uso inicia quando atendente registra entrada de um veículo de um cliente não cadastrado.
- 2) Sistema exibe tela de cadastro de cliente.
- 3) Atendente entra com dados do cliente.
- 4) Sistema registra dados.
- 5) Caso de uso termina.

Subtarefas

Tarefas relacionadas

Figura 17: Tela de criação de caso de uso no fluxo alternativo, estado “Pronto para iniciar”.

Idêntico ao exemplo do caso de uso de sucesso, o estado seguinte “Em elaboração” é criado, conforme pode ser visto na Figura 18.

This screenshot is identical to Figure 17, but the status is 'Em elaboração', which is circled in red. All other details, including the description and progress bar, remain the same.

Situação:	Em elaboração	Início:	06/02/2013
Prioridade:	Normal	Data prevista:	
Atribuído para:	-	% Terminado:	0%
Categoria:	-	Tempo gasto:	-
Versão:	-		

Descrição

Atendente registra dados do cliente.

Básico:

- 1) o caso de uso inicia quando atendente registra entrada de um veículo de um cliente não cadastrado.
- 2) Sistema exibe tela de cadastro de cliente.
- 3) Atendente entra com dados do cliente.
- 4) Sistema registra dados.
- 5) Caso de uso termina.

Subtarefas

Tarefas relacionadas

Figura 18: Tela do estado “Em elaboração” no fluxo alternativo

Seguindo o fluxo alternativo, o estado “Definido” pode ser visto na Figura 19.

Cadastrar cliente
Adicionado por desenvolvedor desenvolvedor 3 minutos atrás. Atualizado menos de um minuto atrás.

Situação:	Definido	Início:	06/02/2013
Prioridade:	Normal	Data prevista:	
Atribuído para:	-	% Terminado:	0%
Categoria:	-	Tempo gasto:	-
Versão:	-		

Descrição

Atendente registra dados do cliente.

Básico:
1) o caso de uso inicia quando atendente registra entrada de um veículo de um cliente não cadastrado.
2) Sistema exibe tela de cadastro de cliente.
3) Atendente entra com dados do cliente.
4) Sistema registra dados.
5) Caso de uso termina.

Subtarefas

Tarefas relacionadas

Histórico

Atualizado por desenvolvedor desenvolvedor há 1 minuto

- Situação alterado de Pronto para Iniciar para Em elaboração

Atualizado por desenvolvedor desenvolvedor há menos de um minuto

- Situação alterado de Em elaboração para Definido

Figura 19: Tela do estado “Definido” no fluxo alternativo

Nota-se a gerência de configuração na aba Histórico, mostrando o autor da atualização da troca de estado e, em seguida, para qual estado foi atualizado. A seguir, na Figura 20, o estado “Em avaliação pela equipe” que reprovará o caso de uso.

Cadastrar cliente
Adicionado por desenvolvedor desenvolvedor 4 minutos atrás. Atualizado menos de um minuto atrás.

Estado:	Em avaliação pela equipe	Data início:	06/02/2013
Prioridade:	Normal	Data fim:	
Atribuído a:	-	% Completo:	0%
Categoria:	-	Tempo gasto:	-
Versão:	-	Ambiguidade?:	Sim
Completeness?:	Sim	Consistência?:	Sim
Testabilidade?:	Não		
Clareza?:	Sim		

Descrição

Atendente registra dados do cliente.

Básico:
1) o caso de uso inicia quando atendente registra entrada de um veículo de um cliente não cadastrado.
2) Sistema exibe tela de cadastro de cliente.
3) Atendente entra com dados do cliente.
4) Sistema registra dados.
5) Caso de uso termina.

Subtarefas

Tarefas relacionadas

Histórico

Atualizado por desenvolvedor desenvolvedor há 3 minutos

- Situação alterado de Pronto para Iniciar para Em elaboração

Atualizado por desenvolvedor desenvolvedor há 2 minutos

- Situação alterado de Em elaboração para Definido

Atualizado por desenvolvedor desenvolvedor há menos de um minuto

- Situação alterado de Definido para Em avaliação pela equipe

Figura 20: Tela do estado “Em avaliação pela equipe” no fluxo alternativo

Este foi um exemplo de um fluxo alternativo que se inicia idêntico ao fluxo de sucesso e é reprovado pela equipe de projeto quanto à testabilidade. Com o apoio da gerência de configuração é possível identificar o responsável pela reprovação e a situação atual da tarefa. Desta forma aumenta-se a comunicação entre as partes envolvidas no processo. Este exemplo demonstra como a gerência de configuração pode auxiliar a organização da equipe no que tange a manipulação dos casos de uso dentro do processo.

3.5 Considerações finais

Este capítulo abordou o uso da ferramenta utilizada para o desenvolvimento do trabalho e também a motivação de sua escolha – o Redmine. Além disso, este capítulo tratou também da criação do processo que se baseia o workflow sugerido para a gerência dos casos de uso, além da criação do mesmo.

O Redmine possui um repositório *wiki* para disponibilizar os casos de usos criados. Neste trabalho, após o estado Definido, que indica a definição e o término da edição, os casos de uso são inseridos no *wiki*. Este componente possui um controle de versões onde, conforme novas alterações forem inseridas, uma nova versão é criada e enumerada. Sendo assim possível visualizar as versões anteriores, as mudanças realizadas, data e hora da mudança, etc.

O Redmine também permite a funcionalidade controle de alterações. Neste trabalho ela pode ser visualizada no estado “Em aprovação pela equipe” no fluxo alternativo, onde o caso de uso é reprovado. Uma aba Histórico é disponibilizada abaixo do diagrama de cada tarefa e nesta aba é possível identificar o responsável pela alteração, estado em que foi alterado, data e hora da alteração, além de possuir campo para edição de texto caso o usuário queira justificar a alteração.

Estas funcionalidades caracterizam a gerência de configuração no trabalho pela ferramenta Redmine.

O Capítulo 4 tratará das considerações finais do trabalho realizado, resumando todas as atividades realizadas neste projeto final.

4 Conclusão

A análise de requisitos e a construção de casos de uso são processos fundamentais em um projeto de software. É essencial para as empresas terem esses processos documentados e possuir o controle sobre tudo o que está sendo feito.

Neste trabalho, foi apresentado o conceito de gerência de configuração, e sua importância como um conjunto de atividades desenvolvidas para administrar modificações ao longo do ciclo de vida do software de computador. Também foi mostrado o conceito de wiki e sua utilização como compartilhador de conhecimento em organizações.

As ferramentas disponíveis no mercado para gerência de configuração de forma geral são completas, mas complexas, o que deixam acadêmicos e desenvolvedores de pequenas empresas sem um acesso facilitado a esta tecnologia, além de possuir alto custo para aquisição da licença. O Redmine, uma ferramenta grátis de apoio ao gerenciamento de projetos, se mostrou melhor no aspecto de controle de modificações e documentação por ser de fácil instalação, configuração e uso.

O produto final deste trabalho inclui a definição de um workflow, utilizando o Redmine, que apoia a construção de um caso de uso do início ao fim, ou seja, desde o levantamento dos requisitos até a sua construção final aprovada.

4.1 – Principais Contribuições

Dentre as principais contribuições deste trabalho inclui-se:

- Definição do processo de construção de casos de uso;
- Configuração do Redmine para apoiar o processo;
- Definição de uma wiki para compartilhar os casos de uso construídos.

4.2 – Trabalhos Futuros

Dentre os trabalhos futuros possíveis, podem-se citar:

- Uso da ferramenta customizada em um projeto real;
- Uso do processo de apoio em um projeto real;
- Configuração da ferramenta customizada para disponibilidade de documentos, sem restringir apenas aos casos de uso;
- Prover maior colaboração dos envolvidos em um projeto, através do processo de apoio e *workflow* criados;
- Configuração do workflow criado baseado nas necessidades do projeto. Alteração e/ou criação de novos estados e definição de novos critérios para clientes e equipe de projeto.

Referências Bibliográficas

BACH, J. **The Highs and Lows of Change Control**, Computer, 1998.

BARBARESCO, Eduardo Alexandre. **Software de apoio ao processo de gerência da configuração segundo normas e modelos da qualidade**. 2000. 64 f. Trabalho de Conclusão de Curso (Bacharelado em Ciências da Computação) – Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau.

BEZERRA, Eduardo. **Princípios de análise e projeto de sistemas UML: Um guia prático para modelagem de sistemas**, Campus, 2006.

BLOG DO BOHM (2010). **Análise de requisito como ferramenta de planejamento**. Disponível em <<http://blogbohm.com/2010/10/18/analise-de-requisitos-como-ferramenta-de-planejamento/>>. Acessado em 14/02/2013.

CASTRO, M.H et al.; (2009). **A volatilidade de requisitos**. Disponível em <<https://repositorio.ufba.br/ri/bitstream/ri/3876/3/014.%20pdf>>. Acessado em 26/02/2013.

DART, S. **Spectrum of Functionality in Configuration Management Systems**, Software Engineering Institute, 2001.

FIGUEIREDO, S., SANTOS, G., ROCHA, A. R. C. **Gerência de Configuração em Ambientes de Desenvolvimento de Software Orientados a Organização**, Simpósio Brasileiro de Qualidade de Software, Brasília, 2004.

FORTE, G. **Rally Round the Repository**, CASE Outlook, 1989.

MOLINARI, Leonardo. **Gerência de Configuração**. Visual Books, 2007.

OLIVEIRA, Júlio (2009). **Requisitos, Casos de Uso e Desenvolvimento**. Disponível em

<<http://sistemasecia.freehostia.com/component/jccmultilanguagecontent/article/34-engenhariasoft/73-reqs-casosuso-desenv.html> Acessado em 20/01/2013.

PRESSMAN, Roger. **Engenharia de Software**. 6ª edição. São Paulo: McGraw-Hill, 2006.

ROCHA, A.R.C.; MALDONADO, J.C.; WEBER, K.C. **Qualidade de software: teoria e prática**. São Paulo: Prentice Hall, 2001.

SILVA, Márcio. **A importância do levantamento de requisitos no sucesso dos projetos de software**. Disponível em <http://www.linhadecodigo.com.br/artigo/1685/a-importancia-do-levantamento-de-requisitos-no-sucesso-dos-projetos-de-software.aspx#ixzz2NeE6RbcY>. Acessado em 15/03/2013

SOFTEX, 2011. Associação para Promoção da Excelência do Software Brasileiro , Guia de Implementação – Parte 4: Fundamentação para Implementação do Nível D do MR-MPS, junho 2011. Disponível em www.softex.br/mpsbr.