

Estudo do desenvolvimento de uma ferramenta de Gerência de Requisitos para apoio aos processos de manutenção de software.

Rafaela Emília Zanetti Teixeira¹, Giuliano Prado de Moraes Giglio²

¹Centro de Ensino Superior de Juiz de Fora (CES-JF)
Juiz de Fora – MG – Brasil

¹rafaelazanetti79@gmail.com, ²giuprado@gmail.com

Abstract. *The objective of this work is to perform a study to identify and evaluate the difficulties in the use of models and processes of the Requirements Engineering in software maintenance and based on the standards found to develop a tool of requirements management to assist the maintenance processes. Software maintenance is now a subject in organizations that develop and maintain software, due to the need to always adjust and improve the software product according to the most diverse needs. Therefore, understanding the meaning, scope and challenges of Requirements Engineering effectively helps organizations and professionals better conduct their efforts in maintaining their products in order to ensure their final quality.*

Resumo. *O objetivo deste trabalho é realizar um estudo para identificar e avaliar as dificuldades no uso de modelos e processos da Engenharia de Requisitos na manutenção de software e com base nos padrões encontrados desenvolver uma ferramenta de Gerência de Requisitos para auxiliar os processos de manutenção. A manutenção de software é hoje um assunto presente em organizações que desenvolvem e mantêm software, devido à necessidade de sempre ajustar e melhorar o produto de software de acordo com as mais diversas necessidades. Portanto, entender o significado, a abrangência e os desafios da Engenharia de Requisitos permite auxiliar de forma efetiva às organizações e profissionais a melhor conduzir seus esforços na manutenção de seus produtos, a fim de garantir sua qualidade final.*

1. Introdução

A manutenção de software pode ser caracterizada como um processo de realizar mudanças em um software após a sua implantação ou até mesmo durante o seu desenvolvimento. Atualmente é um assunto em voga para as empresas que desenvolvem e mantêm software, pois se cria produtos de software com o objetivo de que o mesmo dure por muitos anos. Isso se deve à necessidade de sempre ajustar e melhorar o produto de software de acordo com as mais diversas necessidades. Diante desse fato, entender o significado e abrangência do termo manutenção de software e os desafios da Engenharia de Requisitos na construção de softwares, que permitam a sua modificação ou adaptação, podem auxiliar de forma efetiva as organizações e profissionais a melhor conduzir seus esforços na manutenção de seus produtos.

Existe um legado de software construído com tecnologias ultrapassadas, mal projetados e mal codificados, com documentação ineficiente ou até mesmo inexistente. Isso contribui para o grande esforço em manutenção, visto que muitos desses sistemas são de

extrema importância para as organizações e necessitam que se mantenham em funcionamento. A manutenção pode ser responsável pela maior porcentagem de esforço despendido pelas empresas de desenvolvimento de software. Ela é como um “iceberg”, esperamos que o imediatamente visível seja tudo o que existe, mas sabemos que uma enorme massa de possíveis problemas e custo fica sob a superfície [PRESSMAN, 2002].

Para Sommerville (2005) as mudanças em sistemas de software sempre irão ocorrer, pois as necessidades e os requisitos organizacionais modificam o tempo de vida útil do sistema, requerendo então que mudanças correspondentes sejam feitas no software. Afirma ainda, que o processo de gerenciamento de mudanças pode ser iniciado durante a fase de teste ou até mesmo após a entrega ao cliente, sendo que o processo deve assegurar que seja realizada a análise dos custos e benefícios da mudança, garantindo que esta, seja feita de maneira controlada. Diante da importância de que o sistema contemple de forma correta as necessidades dos clientes e que o atendimento a essas necessidades se mantenha consistente mesmo durante os processos de manutenção e evolução do software, temos o trabalho da Gerência de Requisitos, que é feito visando também à melhoria dos processos de manutenção de software. A aplicação correta da Engenharia de Requisitos na manutenção de software pode contribuir com bastante efetividade na redução de custos e prazos, visto que a manutenção realizada com a utilização de documentos atualizados contribui para a correta implementação das solicitações e com o menor risco de ocorrência de erros ou falhas.

Este trabalho tem como objetivo realizar um estudo para identificar e avaliar as dificuldades no uso de modelos e processos da Engenharia de Requisitos na manutenção de software. Demonstrar como esses modelos podem contribuir de forma efetiva e consistente na manutenção, auxiliando na análise de impacto das mudanças e apoiando a tomada de decisão de se implementar ou não uma solicitação de mudança. Com base nos padrões estudados foi desenvolvida uma ferramenta de Gerência de Requisitos orientada aos processos de manutenção de software, com propriedades que garantem a rastreabilidade e identificação das dependências entre requisitos e contribui para que os requisitos se mantenham sempre atualizados e consistentes ao longo do ciclo de vida do software.

A ferramenta tem a capacidade de coletar, armazenar e manter requisitos, gerenciar as mudanças ocorridas e rastrear os relacionamentos entre os requisitos. Foi desenvolvida com recursos de inclusão e gerência dos requisitos, tratamento da rastreabilidade dos requisitos no processo de mudança onde será possível identificar os requisitos de origem bem como atribuir valores as características de risco, importância, impacto e prioridade. A proposta desta aplicação foi estabelecida a partir do estudo de algumas ferramentas e propostas de gerenciamento de requisitos automatizados, procurando incluir características que as demais ferramentas não possuem e que são importantes para a gerência de mudanças no contexto da manutenção de software. A contribuição também é oferecer à comunidade de Engenharia de Software uma ferramenta de uso livre, de fácil instalação e operação para documentação e gerenciamento dos requisitos, para diferentes projetos, durante todo o ciclo de vida do software.

O artigo estará organizado da seguinte maneira: na unidade subsequente serão apresentados conceitos relacionados à Engenharia de Requisitos e as técnicas existentes de Gerência de Requisitos. Na unidade 3 serão delineados os desafios da Gerência de Requisitos na manutenção de software abordando contextos do desenvolvimento tradicional, metodologias ágeis e software livres. A unidade 4 será apresentada a ferramenta e, finalmente, serão expostas as conclusões e considerações finais sobre o assunto proposto.

2. O Processo de Levantamento e Gerenciamento de Requisitos

Analisar os requisitos de um cliente é identificar o real problema e propor uma solução que atenda os seus desejos e expectativas e que tenha como resultado final um produto de software de melhor qualidade dentro um prazo e orçamento adequados. Esse é o propósito da Engenharia de Requisitos, fornecer mecanismos para entender o que o cliente deseja, documentar os requisitos descobertos e fazer a sua gerência à medida que o software é construído, mantido e evoluído. Entender os requisitos de um problema está entre as tarefas mais difíceis enfrentadas por um Engenheiro de Software [PRESSMAN, 2002]. Entregar ao cliente o que ele deseja e precisa é uma tarefa que exige experiência, atenção e esforço e para se garantir que os requisitos do sistema estejam sempre completos, consistentes, relevantes e atualizados se faz necessário o uso de técnicas sistemáticas e repetíveis.

A Engenharia de Requisitos é um processo que engloba as atividades de descoberta, documentação e manutenção de um conjunto de requisitos de um sistema e são divididos em cinco passos distintos: elicitación de requisitos, análise e negociação de requisitos, especificação de requisitos, modelagem do sistema, validação de requisitos e gestão de requisitos [SOMMERVILLE, 2005]. A gestão de requisitos é de suma importância no ciclo de vida de um sistema, pois é um conjunto de atividades que auxiliam para identificar, controlar, rastrear requisitos e modificações de requisitos em qualquer época [PRESSMAN, 2002].

A elicitación, a análise e a documentação são processos da Engenharia de Requisitos que ocorrem durante a construção do sistema, o nosso ponto de interesse é a gestão de requisitos e mudança, pois são elas que irão auxiliar na realização correta e efetiva da manutenção nos sistemas, pois contribuem para a correta análise das solicitações de mudança, avaliação dos impactos, viabilidade da implementação das mudanças e melhor compreensão e conhecimento do sistema e conseqüentemente diminuição de erros e falhas de implementação e maior agilidade e efetividade para as equipes de manutenção de software. A Gerência de Requisitos inclui a documentação de dependências entre os requisitos, o controle de mudanças sobre as identificações e correções de inconsistências entre o requisito e os artefatos do projeto [MACHADO, 2011]. Não importa o quão cuidadoso seja feito o trabalho de definição dos requisitos, sempre haverá mudanças. Realizar a correta gestão dessas mudanças é o fator que facilita o trabalho da manutenção de software, pois é possível identificar onde a solicitação de mudança deverá ser aplicada e quais requisitos relacionados serão impactados.

Manter uma declaração clara dos requisitos e a rastreabilidade entre eles são os principais itens para um gerenciamento eficiente dos requisitos e desta forma facilitar o trabalho na manutenção, pois a primeira etapa da manutenção deve ser realizar a análise da solicitação de mudança verificando sua viabilidade e impacto nos requisitos já construídos. Entender o sistema e saber identificar onde e como implementar a solicitação de mudança é o segundo passo e para isso é importante que o sistema tenha seus requisitos documentados e atualizado. Após a implementação da mudança é necessário fazer a atualização dos documentos com as alterações realizadas para garantir que a documentação reflita exatamente aquilo que se tem no sistema.

Realizar a Gerência de Requisitos é um trabalho que pode ser complexo. E essa complexidade não vem somente da mudança do requisito, mas também do impacto que essa alteração irá causar em outros requisitos. Por isso é importante fazer uso de vínculos de rastreabilidade das dependências entre requisitos e fazer uma estrutura de requisito que seja adaptável a mudanças. Escrever requisitos que sejam adaptáveis a mudanças é escrever pensando não somente na construção do sistema, mas sim na sua manutenção e evolução ao longo do seu ciclo de vida. Sommerville (2005) diz que é considerada boa prática de Gerência

de Requisitos tentar antecipar as mudanças dos requisitos. Mudanças são inevitáveis e não significam falhas nas práticas de Engenharia de Requisitos, por isso é importante identificar os requisitos mais voláteis e classificá-los tentando se preparar para mudanças que possam surgir. Outro aspecto importante a ser avaliado é que cada proposta de mudança seja avaliada e seja feita a análise de custo benefício da alteração. É preciso garantir também que as mudanças propostas suportem os objetivos de negócio fundamentais do sistema.

Planejar é o primeiro estágio essencial ao processo de gerenciamento de requisitos. Dentre esse planejamento é possível classificar cinco aspectos de decisão durante o gerenciamento dos requisitos: identificação dos requisitos, processo de gerenciamento de mudanças, avaliar impacto e custo das mudanças, políticas de facilidade de rastreamento e suporte de ferramenta CASE [SOMMERVILLE, 2005]. As ferramentas servem para dar apoio e facilitar o trabalho, elas não substituem os processos definidos para gerenciar requisitos, mas auxiliam na análise do impacto, gerenciamento de versões e alterações, na rastreabilidade e reutilização dos requisitos. É sugerido usar uma ferramenta quando se já tem uma abordagem que funciona e requer maior eficiência. Para sistemas pequenos, o uso de ferramentas CASE pode não ser necessário, podendo-se fazer uso de processadores de textos, planilhas, bancos de dados. Mas em sistema mais complexos e maiores o uso destas ferramentas vem de forma a facilitar e agilizar o trabalho da Gerência de Requisitos.

Agilidade é um fator importante na manutenção de software, pois solicitações de mudanças podem surgir em caráter de urgência, como por exemplo, correções de erros que forem encontrados com o software já em produção. Esse é um cenário que necessita de agilidade no atendimento e o uso de processos, modelos ou ferramentas que auxiliem são de fundamental importância para se obter agilidade. Se o processo de gestão dos requisitos for um processo moroso, o que pode acontecer é pular a etapa de análise de impacto dos requisitos e já partir para a implementação da solicitação, o que é um risco, pois pode ocasionar erros e falhas no sistema.

Outro fator relevante é o processo de gerenciamento de mudança, que auxilia no controle das alterações realizadas no sistema, garantindo que os requisitos estejam sempre atualizados e contribuindo para a análise do impacto que as alterações irão causar nos requisitos relacionados e no sistema como todo. Essa análise contribui para estimar o custo e esforço da mudança, bem como verificar a viabilidade de implantação da mudança. Políticas de facilidade de rastreamento auxiliam na agilidade em se identificar quais requisitos relacionados serão impactados pela mudança, por isso é importante ter a unicidade do requisito para garantir a identificação do requisito durante todo o ciclo de vida do software.

Para um bom gerenciamento das mudanças deve-se aplicar o mesmo processo para todas as mudanças propostas nos requisitos. É importante o uso de um processo formal para garantir que as propostas de mudanças sejam tratadas de forma consistente e feitas de maneira controlada [SOMMERVILLE, 2005]. Compreender os requisitos, manter a rastreabilidade bilateral, obter comprometimento com os requisitos, identificar resistências entre os resultados do projeto e dos requisitos e gerenciar as mudanças são práticas executadas para alcançar a meta de Gerência de Requisitos [MACHADO, 2011]. O grande desafio se encontra em manter uma Gerência de Requisitos mesmo após o término do desenvolvimento do software. Após a construção de um sistema não temos mais equipes dedicadas ao projeto, e isso faz com que a gerência dos requisitos não se torne mais uma prioridade. Com isso, mesmo se adotando um modelo bem maduro e estruturado de gerência se não houver comprometimento com a sua manutenção, mesmo após o término da construção do sistema, continuaremos com uma base de requisitos desatualizada.

3. Os Desafios da Gerência de Requisitos na Manutenção de Software

Os requisitos evoluem devido a mudanças no ambiente do sistema e também pela evolução da compreensão e necessidades dos *stakeholders*. Requisitos existentes podem sofrer mudanças e novos requisitos podem surgir em qualquer momento do ciclo de vida de um sistema. Sistemas são feitos para refletir comportamentos do mundo real, por isso é importante que o software acompanhe as mudanças de requisitos que surgem no ambiente ao qual ele está inserido. Uma falha ou dificuldades em acompanhar essas mudanças implicam em perda de qualidade e pode inclusive reduzir a vida útil de um produto [SOMMERVILLE, 2005].

Softwares envelhecem e seu envelhecimento pode ocorrer por conta de mudanças, sejam pela forma como essas mudanças são implementadas, seja pelo objetivo das mudanças, ou até mesmo por tecnologias necessárias ao funcionamento do software se tornarem obsoletas. A manutenção de software atua para retardar esse envelhecimento e tentar contribuir para estender ao máximo o ciclo de vida do software.

A manutenção de software pode ser definida também como uma parte do ciclo de vida de desenvolvimento do software. É o processo geral de modificação de um sistema depois que ele foi colocado em uso [SOMMERVILLE, 2005]. É definida pela norma IEEE 1219 como a modificação de um produto de software depois da entrega para corrigir falhas, para melhorar o desempenho ou outros atributos, ou para adaptar o produto a mudanças do ambiente [IEEE 1219, 1998]. Embora essa definição trate de qualquer produto de software existem diferenças em manutenção de softwares com propósitos distintos. O processo de manutenção de software deve considerar os diferentes tipos de software. Temos, por exemplo, softwares legados, softwares do tipo embarcados e softwares livres onde cada tipo possui sua especificidade e sua manutenção deve ser tratada de forma a atender de forma completa o seu contexto. Desta forma, se faz necessário buscar métodos que auxiliem na manutenção de forma a minimizar os impactos que essas alterações podem causar no funcionamento do sistema. Manutenções realizadas sem o correto entendimento da estrutura do sistema podem ocasionar falhas, que ao decorrer do tempo e a cada nova mudança, se tornará cada vez mais difícil e possivelmente mais cara.

Em um processo ideal de manutenção de software, o estágio de implementação de mudanças deve modificar a especificação, ou seja, os requisitos, o projeto e a implementação do sistema, a fim de refletir a modificação completa no sistema [SOMMERVILLE, 2005]. Mas isso não é o que reflete a realidade. Muitas vezes pressionados pela questão do tempo, alterações são realizadas sem a análise e atualização dos documentos, fazendo com que a documentação existente se torne, com o passar do tempo, inútil. Nesses casos, é tido como mais importante fazer a mudança rapidamente do que assegurar que o processo formal seja seguido. Contudo, o perigo dessa abordagem é que os requisitos, o projeto de software e o código, gradualmente se tornam inconsistentes [SOMMERVILLE, 2005].

Os diferentes ambientes e modelos de desenvolvimento de software também devem ser levados em consideração quando se fala em manutenção de software. Ambientes ágeis de desenvolvimento onde se prioriza maior interação entre as pessoas do que documentação, de software legados onde é possível encontrar desde documentação desatualizada até a inexistência de documentos e também no caso de softwares livres que podem não ter uma equipe exclusiva e dedicada para construção e evolução do software, mas pode acontecer de forma colaborativa entre desenvolvedores espalhados pelo mundo todo.

3.1 Gerência de Requisitos em Metodologias Ágeis

No uso de metodologias ágeis, temos processos diferentes de construção e entrega de software, são modelos com liberações frequentes de funcionalidades construídas e nesses cenários, o tempo de análise, construção e entrega é mais curto. Esse modelo traz a proposta de quebrar o paradigma de se trabalhar com um processo e documentação engessados. Mas isso é interpretado, muitas vezes, de forma errônea. Substitui-se o grande volume de documentação de processos tradicionais para uma documentação quase inexistente. O que pode ocasionar grande dificuldade na manutenção de sistemas em equipes de desenvolvimento ágeis quando não é feita a gerência dos requisitos.

O Scrum, que é um modelo ágil de desenvolvimento suporta alta taxa de mudança de requisitos e por ser um processo adaptado a mudanças ele pode se tornar também um modelo de processo para a manutenção de software. Nele os requisitos são organizados em uma lista chamada de *Product Backlog*, que é uma lista que contém todas as funcionalidades do produto [DESENVOLVIMENTO ÁGIL, 2017]. Essa lista também pode conter solicitações de mudanças e é tarefa do *Product Owner* manter o *backlog* atualizado e priorizado. A cada Sprint, que são ciclos completos de desenvolvimento, os requisitos são revistos, aprimorados e o feedback é constantemente utilizado para construir um melhor produto. O que acaba por se caracterizar com um modelo Gerência dos Requisitos dentro de um ambiente ágil.

O Extreme Programming, XP, é outro modelo de desenvolvimento ágil que tem como objetivo principal levar ao extremo as práticas que são ditas como boas na engenharia de software. Tem como premissas a agilidade, com qualidade e que atenda as necessidades do cliente [OLIVEIRA et al, 2009]. Em contrapartida, ele descarta uma documentação pesada priorizando a comunicação, o que pode ser um fator preocupante quando o assunto é manutenção, pois o conhecimento fundamental do sistema está nas pessoas que participam da sua construção e nem sempre teremos a mesma equipe ou as mesmas pessoas alocadas para realizar a manutenção do sistema após a sua conclusão.

Apesar dos modelos ágeis não trabalharem com uma vasta gama de documentação eles possuem padrões e processos que podem aumentar a produtividade e também a qualidade dos produtos entregues, o que visa facilitar a manutenção dos sistemas. Adotam modelos que contribuem para transmissão do conhecimento entre os membros das equipes, que por um lado contribui para o entendimento do sistema para realizar correções e mudanças com mais agilidade e eficiência, mas em contrapartida necessita que as mesmas pessoas envolvidas no processo de construção estejam sempre envolvidas nos processos de manutenção. Esses modelos exigem um nível de maturidade maior dos membros da equipe e de um maior conhecimento dos padrões e processos, o que pode gerar um custo maior para o projeto, pois exige profissionais com um nível maior de conhecimento e senioridade.

Silva et al (2009) relata em seu artigo como uma das desvantagens dos métodos ágeis, por desenvolver em incrementos é a implementação de um dado processo de negócio sem identificar todas as relações com os demais processos, o que pode desencadear a manutenção e retrabalho. A Gerência de Requisitos não pode impedir que isso ocorra, por não ter o conhecimento do todo, ou seja, de todos os requisitos que compõem o projeto, mas pode auxiliar na análise de impacto nos requisitos já identificados contribuindo para a redução no retrabalho e auxiliando na manutenção do software nesses cenários. Mesmo em um ambiente ágil e com um volume de documentação menor é importante que seja adotado um modelo de Gerência de Requisitos, pois ela pode garantir que mudanças sejam feitas de forma, ágil, completa, correta e que não gere impacto em funcionalidades dependentes e pode contribuir para a disseminação do conhecimento.

3.2 Gestão da documentação em Sistemas Legados

Já a manutenção de sistemas legados é um desafio à parte. Toma-se como sistema legado um sistema antigo, frequentemente mal projetado ou documentado, mas que é crítico para o negócio e deve ser suportado por vários anos [PRESSMAN, 2002]. Normalmente sistemas que oferecem alto risco para o negócio são candidatos a serem mantidos, pois o risco na sua substituição é crítico. Diante disso tem-se que a realização da gerência dos requisitos e das solicitações de mudança, pode contribuir para o menor risco de erros, uma correta implementação das solicitações e diminuição das falhas. Nas situações onde não exista documentação, o ideal é documentar as mudanças realizadas para que ao passar do tempo se consiga gerar uma base de documentação para esses sistemas.

Um estudo realizado traçou uma estratégia para substituição de sistemas legados utilizando técnicas de métodos ágeis [NEIVA et al, 2014]. Foi utilizado um modelo baseado nas práticas do Scrum. O processo foi dividido nas fases de pré-desenvolvimento, desenvolvimento e pós-desenvolvimento. A fase de pré-desenvolvimento foi a fase de entendimento e conhecimento do sistema legado, foram usadas técnicas de reengenharia, pois a equipe de desenvolvimento não foi a mesma equipe que construiu o sistema e o foram observados os cenários onde a documentação era inexistente ou desatualizada. Para esse processo foi necessário o envolvimento dos usuários e clientes, bem como a análise do código-fonte. Um fator importante foi documentar todas as informações levantadas para se criar uma nova base de conhecimento para o sistema. A partir dos elementos produzidos foi estabelecido o Product Backlog para iniciar o desenvolvimento, destacando que o mesmo poderia ser modificado na etapa de desenvolvimento para atender as alterações de requisitos.

Um fator bastante relevante para o andamento do processo foi à gestão do conhecimento. Para otimizar o desenvolvimento, a manutenção e a comunicação da equipe, era relevante que o conhecimento do projeto estivesse bem organizado. Vários processos foram utilizados para a gestão do conhecimento, em destaque o controle de versão, construção da matriz de rastreabilidade dos requisitos, controle das mudanças dos requisitos, que são itens relevantes da Gerência de Requisitos.

O estudo concluiu que a utilização de práticas ágeis que lidam com processos mais simples, mais eficientes e que atendem às mudanças de requisitos, com práticas voltadas para o controle contínuo do conhecimento são essenciais para atender ao caráter evolutivo desse tipo de software. Destacando que a eficiência dessa estratégia depende muito mais das pessoas, pois necessita de uma equipe comprometida e com conhecimento dos processos envolvidos. Mesmo em cenários tão ambíguos que é o modelo de desenvolvimento ágil que adota práticas, processos, padrões e modelos mais atualizados e o de um sistema legado construído muitas vezes com tecnologias e modelos defasados é possível reverter à situação e obter um produto final com qualidade e que se mantenha em acordo com as necessidades dos clientes e aderente ao objetivo do negócio. Isso foi possível com o uso de práticas e modelos de Gerência de Requisitos que contribuíram para a organização, armazenamento e divulgação do conhecimento do sistema, mesmo em softwares tidos como legados.

3.3 Gerência de Requisitos em software Livre

Outro contexto a ser avaliado é quando se trabalha com software livre. Esse modelo de desenvolvimento trabalha com a liberdade de evolução do software e disseminação do conhecimento. Ao distribuir cópias de suas versões modificadas a outros, você pode dar a toda comunidade a chance de beneficiar de suas mudanças. Para tanto, acesso ao código-fonte é um

pré-requisito [GNU, 2017]. Esse é um modelo de disseminação do conhecimento do software e o mesmo pode estar sempre em um processo evolutivo de melhoria.

Nesse cenário temos um software que é construído e evoluído de forma colaborativa por profissionais e voluntários no mundo todo. Esse processo muitas vezes se dá através de comunidades e fóruns o que é uma vantagem, não apenas pela quantidade de colaboradores, mas também pela diversidade de visões e níveis de conhecimento, o que facilita detectar mais facilmente erros de código. O processo de manutenção é realizado de modo colaborativo e com mais conhecimento em seu conjunto do que somente um indivíduo ou uma equipe em uma organização possa contratar.

Nos projetos de software livre, a primeira vista, não se dá muita ênfase à especificação de requisitos. Um modelo formal de documentação de requisitos é algo que parece ser raro entre a comunidade de software livre, apesar de alguns sistemas mais complexos possuam especificações descrevendo partes restritas de seu funcionamento [REIS, 2003]. Apesar de não existir um modelo formal de Gerência de Requisitos ou até mesmo uma documentação dentro de modelos padrões, o processo de desenvolvimento de software de forma livre contribui para a disseminação do conhecimento e o objetivo é a constante evolução do sistema.

Silva e Falbo (2006) desenvolveram uma proposta para a definição de um processo padrão para desenvolvimento de Software Livre com base nos processo da MPS.BR, Melhoria do Processo de Software Brasileiro, e verificaram que o modelo de Gerencia de Requisitos proposto por esse modelo é aplicável ao desenvolvimento e manutenção de Software Livre, onde as atividades de Levantamento e Análise de Requisitos, são de responsabilidade do colaborador e as atividades de Integração, testes e validação são realizadas também por outros membros colaboradores. No desenvolvimento de Softwares Livres são utilizados fóruns e comunidades para troca de experiências e conhecimentos e esses meios acabam por ser tornar uma base documental de todo o sistema, utilizar uma ferramenta que agrupe e organize essas informações é um meio de gerenciar o conhecimento e possivelmente os requisitos do sistema.

4. Ferramenta para Gerência de Requisitos

Ultimamente vem se desenvolvendo na comunidade de engenharia de software um crescente interesse em processos, modelos e normas de qualidade, visando não somente atender as necessidades dos clientes, mas também com o objetivo de gerar produtos com mais qualidade. É possível identificar em normas e modelos propostos por organizações mundiais como o IEEE, Instituto de Engenheiros Eletricistas e Eletrônicos, o PMI, *Project Management Institute*, a ISO, Organização Internacional para Padronização e o SEI, Sistema Eletrônico de Informações, que o processo de Gerência de Requisitos é o primeiro aspecto a ser tratado na busca pela qualidade. Isso só vem a confirmar a importância do trabalho da Gerência de Requisitos no desenvolvimento e manutenção do software.

O uso de uma ferramenta de gerenciamento de requisitos auxilia e facilita o trabalho de gerenciamento dos requisitos, contribuindo para essa busca de qualidade e trazendo grandes benefícios para um projeto de software, entre eles podemos destacar: gerenciar versões e mudanças, armazenar os atributos dos requisitos, facilitar a análise de impacto, controle de acesso, comunicação com as partes interessadas e reuso de requisitos [WIEGERS, 2003].

Existem muitas ferramentas disponíveis no mercado que auxiliam nas atividades relativas ao processo de Gerência de Requisitos, mas a grande maioria delas é orientada aos projetos de desenvolvimento de software e não oferecem suporte adequado aos processos de manutenção de software. Belleza (2009) ao realizar o comparativo entre as ferramentas RequisitePro, Compuware Optimal Trace Enterprise e DOORS constatou que elas são deficientes nos seguintes pontos:

- Não possibilitam informar quais os requisitos foram afetados pelas requisições de mudança e de que forma foram afetados, ou seja, se foram adicionados, removidos ou alterados;
- Não possuem a habilidade de manter requisitos de aplicação através dos projetos de manutenção;
- Não possuem a habilidade de ligar requisitos de aplicação com requisições de mudanças através de relações como Adicionar, Alterar, Remover;
- Não possuem a habilidade de atualizar os requisitos de aplicação de forma automática quando a implementação da requisição de mudança é entregue ao cliente da aplicação.

Dentro desse contexto, foi desenvolvido o objetivo de criação da ferramenta Req Manager que é realizar o gerenciamento dos requisitos não orientados ao projeto, mas sim ao produto de software e permitir o controle e gerencia dos requisitos seja durante o projeto de desenvolvimento do software ou durante os diversos projetos de manutenção que o produto poderá passar.

A ferramenta Req Manager possui funcionalidades que vem atender algumas das deficiências relatadas por Belleza (2009) , que são elas:

- É possível obter a informação de quais requisitos foram afetados pelas solicitações de mudança, ao registrar e finalizar uma solicitação, a informação da alteração é refletida no requisito ou requisitos impactados e é exibida no versionamento juntamente com a identificação da solicitação de mudança.
- Permite associar as solicitações de mudança o requisitos ou requisitos afetados pela solicitação, bem como ligar aos requisitos as relações de adição, alteração e remoção.
- Permite manter requisitos através de diferentes projetos de manutenção.

O diferencial da ferramenta Req Manager é dar o suporte aos diferentes projetos de manutenção de um software e acompanhar a evolução dos requisitos ao longo do seu ciclo de vida.

4.1 Comparativo entre as Ferramentas

Em busca da diferenciação da proposta do desenvolvimento da ferramenta Req Manager, buscou-se realizar um levantamento das ferramentas disponíveis com o mesmo propósito de gerenciamento de requisitos que a Req Manager, a fim de se colher as virtudes e deficiências das mesmas, como forma de se estabelecer uma proposta pautada pela diferenciação qualitativa. Nesta seção são apresentados alguns estudos comparativos entre algumas ferramentas existentes no mercado, as quais foram levantadas no levantamento exploratório desta pesquisa.

Calazans et al. (2014) realizou um estudo comparativo entre as ferramentas TraceM, Caliber e RequisitePro com base nos critérios de pertencerem a grandes fornecedores e de popularidade das ferramentas. Esse estudo analisou as principais funcionalidades das

ferramentas com relação à Gerência de Requisitos e atendimento aos requisitos identificados. O resultado encontrado é exibido na Tabela 1.

Tabela 1. Quadro Comparativo entre as Ferramentas

Itens Analisados	TraceM	CaliberRM	RequisitePro
Registrar funcionalidades	Sim	Sim	Sim
Registrar requisitos funcionais	Sim	Sim	Sim
Registrar regras de negócio	Sim	Sim	Sim
Registrar atributos	Sim	Sim	Sim
Vincular requisitos funcionais às funcionalidades	Sim	Sim	Sim
Vincular funcionalidade aos atributos	Sim	Sim	Sim
Vincular funcionalidades às regras de negócio	Sim	Sim	Sim

Fonte: Adaptado de CALAZANS et al. (2014)

Como pode ser observado, todas as ferramentas analisadas atendem aos atributos analisado, porém não foram analisados itens relevantes como versionamento e controle de alterações que são funcionalidades importantes de Gerência de Requisitos na manutenção de software.

Grings and Sayão (2009) realizaram desenvolvimento de um ferramenta de Gerência de Requisitos, o OpenReq e fizeram o comparativo da ferramenta com outras existentes no mercado analisando os seguintes aspectos: tipos de requisitos e seus atributos, cadastro e edição de requisitos, controle e histórico de alterações, baseline e controle de versões, fórum de discussões, integração com outras ferramentas, acesso concorrente, acesso Web e restrições de ambiente. A tabela 2 exibe os resultados encontrados.

Tabela 2. Quadro Comparativo entre as Ferramentas

Itens Analisados	RequisitePro	CaliberRM	DOORS	OpenReq
Tipos de requisitos e seus atributos	Sim	Sim	Sim	Sim
Cadastro e edição de requisitos	Sim	Sim	Sim	Sim
Controle e histórico de alterações	Sim	Sim	Sim	Sim
Baseline e controle de versões	Sim	Sim	Sim	Não
Fórum de discussões	Sim	Sim	Não	Não
Integração com outras ferramentas	Sim	Sim	Sim	Não
Acesso concorrente	Sim	Sim	Sim	Sim
Acesso Web	Sim	Sim	Sim	Sim
Restrições de ambiente	Sim	Sim	Sim	Sim

Fonte: Adaptado de [Grings and Sayão 2009]

Os estudos não analisaram nas ferramentas características relevantes a manutenção de software como manter requisitos através diferentes projetos de software, manter requisições de mudança e associá-las aos projetos e requisitos, bem como identificar de que forma essas

requisições afetaram os requisitos. Baseado nesta análise pode-se verificar a importância de se desenvolver uma ferramenta para gerenciar os requisitos que permita acompanhar a evolução de uma aplicação ao longo de todo o seu ciclo de vida.

4.2 Desenvolvimento da Ferramenta Req Manager

A ferramenta foi desenvolvida com base em um modelo de gestão de requisitos proposto por Wieggers (2003), conforme mostrado na figura 1, que divide a gerência de requisitos em quatro partes: controle de mudanças, controle de versão, acompanhar o estado dos requisitos e rastrear os requisitos.

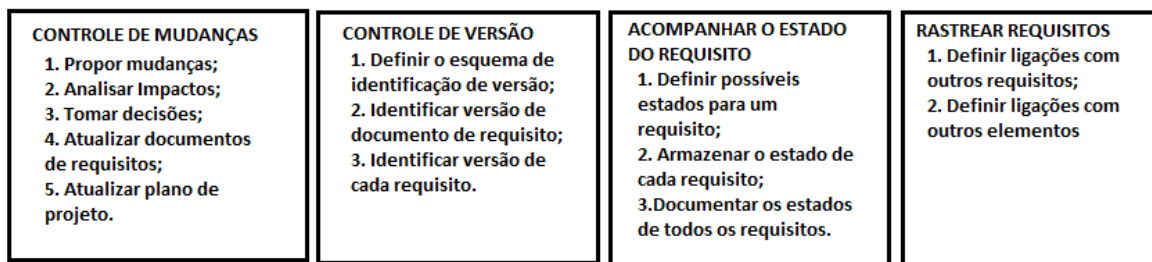


Figura 1. Modelo de Gestão de Requisitos. Fonte: Adaptado de WIEGERS (2003)

No quesito Controle de Mudança, a ferramenta permite inclusão e gerência das solicitações de mudança, realizando a associação com os requisitos afetados e suas dependências. Permite incluir atributos de custo, prioridade e risco nas solicitações de mudança, agrupando informações necessárias a análise de impacto e a tomada de decisões. A ferramenta, com base nesse modelo de gestão, permite o controle das versões dos requisitos geradas a cada alteração, realiza o acompanhamento de todos os requisitos permitindo acompanhar e visualizar seu estado, além de se obter o rastreamento dos requisitos inclusive nas solicitações de mudança. Tem a capacidade para coletar, armazenar e manter requisitos e gerenciar as mudanças ocorridas. Será desenvolvida com recursos de tratamento da rastreabilidade dos requisitos no processo de mudança onde será possível identificar os requisitos de origem bem como atribuir valores as características de volatilidade, importância, e prioridade.

A ferramenta será desenvolvida em PHP, aplicando Orientação a Objetos, será projetada para um ambiente web, utilizando MySQL como Sistema Gerenciador de Banco de Dados. Com o intuito de a ferramenta ser de livre distribuição e uso, foram utilizadas ferramentas OpenSource para o seu desenvolvimento.

Foi utilizada uma plataforma Web para permitir o acesso e gerenciamento em uma base única e centralizada e utilizando uma arquitetura MVC, *Model-view-controller* [BAPTISTELLA, 2009]. A Figura 1 representa a estrutura da aplicação.

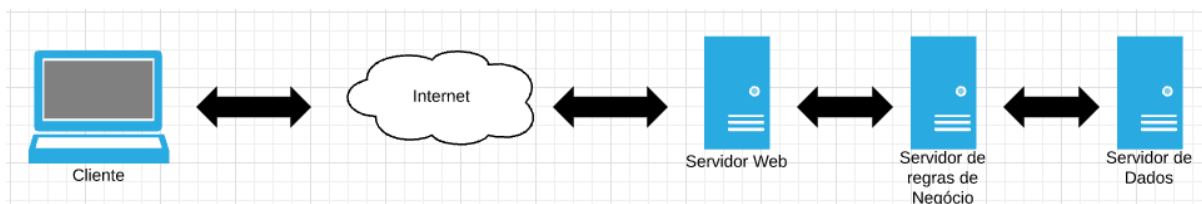


Figura 1 – Req Manager - Ambiente de Operação em Três Camadas. Fonte: Do Autor

As requisições dos clientes, oriundas da Internet são gerenciadas por um Servidor Web, que repassa ao Servidor de Regras de Negócio, onde se concentra a lógica da aplicação. A base de informações do sistema fica no Servidor de Dados.

4.3 Funcionalidades do Req Manager

A ferramenta contém um módulo de administração onde é possível gerenciar os seguintes cadastros: usuários, glossário, volatilidade, risco, importância, impacto, motivo, custo, que são cadastros básicos. Conforme é possível observar na Figura 2. A parametrização e configuração dos cadastros acima são de responsabilidade do administrador do sistema.

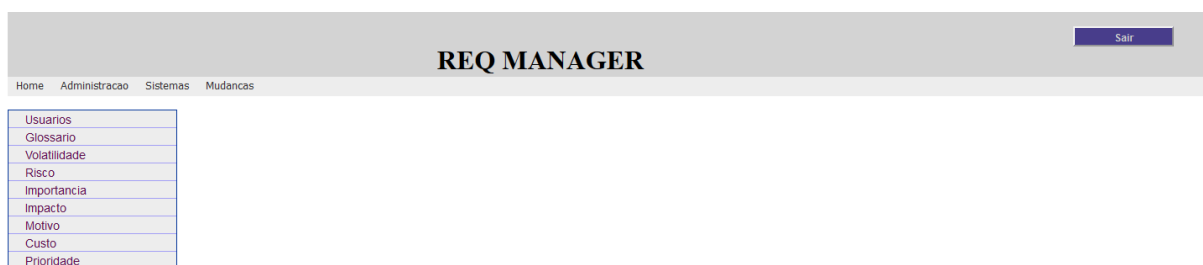


Figura 2 – Req Manager - Módulo Administração. Fonte: Do Autor

A Figura 3 traz o módulo de Sistemas onde são mantidas as aplicações, projetos e requisitos. Permite realizar o cadastro dos sistemas e dos seus projetos, sejam eles de desenvolvimento ou manutenção. Permite cadastrar e alterar os requisitos do sistema, realizar a relação entre requisitos de forma a obter a rastreabilidade entre eles. Nesse módulo também é possível relacionar os documentos produzidos como casos de uso, documentos de especificação, documentos de textos, entre outros.

Nesse módulo são exibidos os dados da aplicação, como nome, data de criação, situação. Possui um menu lateral onde é possível navegar entre as opções de incluir projetos, novos requisitos, incluir ou pesquisar termos no glossário, anexar documentos e gerar a matriz de rastreabilidade dos requisitos. Na parte inferior da tela o sistema exibe os projetos e os requisitos cadastrados. Essas informações são exibidas dentro de uma estrutura hierárquica onde os projetos são classificados em Projeto de Desenvolvimento e Projetos de Manutenção e os requisitos são agrupados em Requisitos Funcionais, Requisitos Não Funcionais e Restrições.

[Sair](#)

REQ MANAGER

Home Administração Sistemas Mudanças

- Projetos
- Requisitos
- Documentos
- Matriz de Rastreabilidade
- Glossário

SISTEMAS - DETALHAR

Código Sistema:

Status:

Nome:

Data de Criação:

Projetos

- Desenvolvimento
- Manutenção

Requisitos

- Funcionais
 - REQ - 1
 - REQ - 2
 - REQ - 3
- Não Funcionais
 - REQ - 4
- Restrições

Dados Gerais

Versões

Solicitações de Mudança

[Alterar Status](#) [Editar](#)

Identificador: Título:

Projeto: Situação:

Prioridade: Volatilidade:

Importancia:

Descrição:

Funcionalidade que permite a gerencia dos registros de funcionarios.

Dependencias

Afeta os seguintes requisitos:

Identificador	Titulo
REQ - 2	Manter clientes
REQ - 3	Manter filmes

É afetado pelos seguintes requisitos:

Identificador	Titulo

Figura 3 – Req Manager – Aba Dados Gerais. Fonte: Do Autor

Na criação de um requisito o sistema irá gerar de forma automática um identificador único do requisito independentemente do projeto, isso permite criar vínculos de rastreabilidade entre requisitos de um projeto ou até mesmo entre projetos diferentes. Contribuindo para manter a rastreabilidade entre requisitos de projetos que se comunicam entre si e possibilitando uma análise de impacto mais completa de uma solicitação de mudança e também permitindo o gerenciamento do requisito através de diferentes projetos de manutenção de software. A ferramenta também permite manter o histórico do requisito e armazenar a informação do ciclo de vida do requisito, desde a sua criação, modificação ou até mesmo sobre sua remoção.

No módulo “Sistemas” é possível manter o rastreamento entre todas as versões de um requisito criado e também que seja feito o vinculo de rastreabilidade do requisito com outros artefatos do projeto, sendo considerados como outros artefatos, outros requisitos, documentos que forem anexados ao projeto ou solicitações de mudança. Isso garante criar vínculos de rastreabilidade horizontal e vertical, onde rastreabilidade horizontal é a rastreabilidade entre diferentes versões de requisitos e a rastreabilidade vertical é realizada entre requisitos e artefatos produzidos pelo processo de desenvolvimento ao longo do ciclo de vida do projeto [GENVIGIR, 2009].

Conforme é possível observar nas Figuras 3,4 e 5, a ferramenta disponibiliza em forma de abas, para melhor navegação, todas as informações do requisito. Na aba Dados Gerais é possível visualizar o requisito, bem como seus atributos e suas relações de dependências. Na aba Versões a ferramenta permite visualizar todas as versões do requisito, exibe a justificativa de alteração do mesmo e também traz a informação de qual solicitação de mudança motivou a criação de uma nova versão do requisito.

Sair

REQ MANAGER

Home Administração Sistemas Mudanças

Projetos

Requisitos

Documentos

Matriz de Rastreabilidade

Glossário

SISTEMAS - DETALHAR

Código Sistema:

Status:

Nome:

Data de Criação:

Projetos

- Desenvolvimento
- Manutenção

Requisitos

- Funcionais
 - REQ - 1
 - REQ - 2
 - REQ - 3
- Não Funcionais
- Restrições

Dados Gerais

Versões

Solicitações de Mudança

Versão	Status	Data da Alteração	Motivo Alteração	Responsável
1	Novo	08/12/2017 10:41:39	Criação	Rafaela Zanetti
2	Alterado	08/12/2017 10:44:56		Rafaela Zanetti
3	Alterado	08/12/2017 10:52:42	Correcao do nome	Rafaela Zanetti

Figura 4 – Req Manager – Aba Versões. Fonte: Do Autor

Na aba Solicitações de Mudança a ferramenta exibe todas as solicitações de mudança associadas ao requisito, que motivaram ou não modificações nos mesmo. São exibidas as informações de impacto da solicitação, custo, risco e motivo e é possível também identificar a qual projeto pertence à solicitação de mudança, conforme é possível observar na Figura 5.

Sair

REQ MANAGER

Home Administração Sistemas Mudanças

Projetos

Requisitos

Documentos

Matriz de Rastreabilidade

Glossário

SISTEMAS - DETALHAR

Código Sistema:

Status:

Nome:

Data de Criação:

Projetos

- Desenvolvimento
- Manutenção

Requisitos

- Funcionais
 - REQ - 1
 - REQ - 2
 - REQ - 3
- Não Funcionais
 - REQ - 4
- Restrições

Dados Gerais

Versões

Solicitações de Mudança

Código	Data da Alteração	Motivo Alteração	Status	Responsável
MUD - 2	08/12/2017 10:59:20	Melhoria	Novo	Rafaela Zanetti

Figura 5 – Req Manager – Aba Solicitações de Mudança. Fonte: Do Autor

A ferramenta centraliza em um único lugar todas as informações necessárias para conhecimento e entendimento da evolução de um requisito, mesmo dentro de um contexto de diferentes projetos ao qual um sistema pode ter passado. É possível verificar facilmente em qual projeto determinada alterações foi realizada em um requisito e qual solicitação motivou essa alteração e qual o estado final do requisito após essa alteração.

A Figura 6 traz o módulo de mudanças, que permite que as solicitações de mudança sejam propostas e com base nelas os impactos são analisados para que sejam tomadas as decisões de se implementar ou não uma mudança e para que se tenha o controle das alterações realizadas. Permite incluir atributos de impacto, custo, risco, prioridade e motivo a uma solicitação de mudança.

[Sair](#)

REQ MANAGER

Home Administração Sistemas Mudanças

Detalhar Solicitação Mudanca

Título:
Alterar o cadastro de cliente

Descrição:
Incluir campo para informar o celular do cliente.

Sistema:
Loca Video Web

Motivo:
Melhoria

Prioridade:
Alta

Risco:
Alto

Impacto:
Alto

Custo:
Alto

Status:
Novo

Horas:
02 : 00

Requisitos Afetados

Identificador	Titulo
REQ - 2	Manter clientes

[Voltar](#)

Figura 6 – Req Manager – Solicitações de Mudança. Fonte: Do Autor

A ferramenta permite associar na própria solicitação de mudança o requisito ou requisitos afetados pela solicitação, e com base nos requisitos associados ela realiza uma busca das relações de dependência destes requisitos com outros, permitindo uma melhor análise de impacto da solicitação, visto que o usuário terá a visão completa de todos os requisitos afetados pela alteração.

4.4 Comparativo de Funcionalidades do Req Manager

A tabela 3 exhibe a relação das funcionalidades analisadas nos estudo comparativos de ferramentas citados anteriormente e quais destas funcionalidades são contempladas pela ferramenta Req Manager. Essa análise foi realizada com intuito de apoiar o processo de identificar as funcionalidades a serem implementadas no Req Manager.

Tabela 3. Quadro de Análise das Funcionalidades do Req Manager

Itens Analisados	Req Manager
Tipos de requisitos e seus atributos	Sim
Cadastro e edição de requisitos	Sim
Controle e histórico de alterações	Sim
Baseline e controle de versões	Sim
Fórum de discussões	Não
Integração com outras ferramentas	Não
Acesso Web	Sim
Restrições de ambiente	Sim
Habilidade de ligar requisitos com requisições de mudanças	Sim
Habilidade de manter requisitos através de diferentes projetos	Sim

Fonte: Do Autor

A conforme é possível verificar na tabela, a ferramenta possui a maioria das funcionalidades das principais ferramentas disponíveis no mercado e também as funcionalidades para manter solicitações de mudanças, ligar essas solicitações aos requisitos e manter requisitos através de diferentes projetos de manutenção, que são funcionalidades de extrema importância no apoio às atividades de manutenção de software.

Nos testes realizados na ferramenta constatou-se o seguinte:

- Todos os cadastros básicos necessários ao uso das funcionalidades estão contemplados na ferramenta.
- Identificou-se a necessidade de uma alteração futura no cadastro de usuário para classificá-los de forma a atribuir responsabilidade específica para que seja possível identificar os papéis dos usuários dentro de cada projeto do sistema.
- Nas gerações de versões identificou-se a necessidade de um tratamento e classificação das alterações, é importante o versionamento dos requisitos a cada alteração, mas nem toda alteração realizada em um requisito é relevante para o acompanhamento da evolução do mesmo, como por exemplo, ajustes no texto do requisito com correções de erros de grafia.
- As inclusões dos requisitos e sua manutenção foram efetuadas sem impedimentos e as dependências foram incluídas com sucesso.
- Foi identificada também durante os testes a necessidade de se gerar relatórios, como por exemplo, relatórios de requisitos por situação, relatórios de dependências, quantidade de horas em solicitações de mudança, entre outros.

Com base nisso, foi possível verificar que a ferramenta atende aos requisitos inicialmente propostos, apesar de necessitar de algumas correções e melhorias. A ferramenta apresenta grande potencial de uso na comunidade de Engenharia de Software.

5. Considerações e Trabalhos Futuros

Este trabalho apresentou o apoio da Gerência de Requisitos nas atividades de manutenção de software, demonstrando a vantagem em se adotar modelos e padrões de gerência para facilitar a análise e impacto de solicitações de mudanças em um produto de software, obtendo maior agilidade e eficiência no atendimento das solicitações de mudança. Demonstrou que o processo de gerenciamento de requisitos auxilia na identificação das atividades, controle e rastreamento dos requisitos e apoia no gerenciamento das mudanças durante todo o ciclo de vida do software, contribuindo para a manutenção da qualidade.

A dificuldade atual da gerência de requisitos na manutenção encontra-se no fato que o foco das empresas é voltado principalmente para a etapa de desenvolvimento do software. Mudanças são realizadas sem a devida análise de impacto e sem a atualização dos requisitos, gerando falhas, erros e reduzindo a qualidade do software. Se na manutenção os requisitos não forem gerenciados de forma eficiente, muito tempo pode ser demandado para a manutenção e um alto custo pode ser gerado. É preciso uma mudança de cultura no sentido de se preocupar com a manutenção buscando tecnologias que proporcionem um novo método ou técnica para manutenção e evolução de software, que gere reduções de custo e agilidade nos processos.

O uso de ferramentas de gerência de requisitos possibilita reduzir essas dificuldades, pois, reduzem o esforço e tempo, agilizam o processo com a diminuição da documentação, automatizam atividades do processo, impactando diretamente na melhoria do processo de manutenção de software. Possuem benefícios que colaboram no acompanhamento do projeto, possibilitando um fluxo de desenvolvimento e manutenção mais previsível, possibilitando a redução de erros, falhas e retrabalhos, contribuindo com a diminuição dos atrasos nas entregas, e conseqüentemente, reduzindo o custo com a manutenção.

A ferramenta ReqManager foi desenvolvida com o intuito de apoiar os processo de manutenção de software. Possui funcionalidades que auxiliam a manutenção, pois permite gerenciar dentro da própria ferramenta as solicitações de mudança, associa-las aos requisitos, rastrear dependências e acompanhar os diferentes estados que um requisito poderá ter durante todo o ciclo de vida do software dentre os diferentes projetos que o mesmo poderá passar. É importante a realização novos estudos para buscar outros modelos de Gerencia de Requisitos com intuito de aprimorar o modelo desenvolvido na ferramenta, com a inclusão de novas funcionalidades ou evolução das funcionalidades já construídas, obtendo assim uma ferramenta mais completa no auxilio das atividades de manutenção de software.

Como trabalhos futuros pretende-se testar a ferramenta em um ambiente corporativo, com projetos reais, a fim de validar a efetividade da ferramenta, realizando uma comparação entre os testes realizados no modelo experimental e acadêmico e os testes aplicados em um ambiente real, permitindo identificar e corrigir possíveis erros e deficiências da ferramenta e verificar a sua contribuição efetiva nas atividades de manutenção. É preciso realizar a sua evolução com a inclusão das funcionalidades de notificação de usuários e fórum de discussão, que são funcionalidades úteis quando se trata de Software Livre. Permitir a classificação dos usuários de forma a identificar os papeis e responsabilidades de cada usuário nos projetos de desenvolvimento e manutenção do software. Aperfeiçoar a visualização do versionamento, a fim de permitir o melhor acompanhamento das alterações mais relevantes realizadas nos requisitos.

Referências

- BAPTISTELLA, Adriano José (2009) “Abordando a arquitetura MVC, e Design Patterns: Observer, Composite, Strategy”. Academico de Sistemas de Informação, UNIPAR. Disponível em: [/www.linhadecodigo.com.br/artigo/2367/abordando-a-arquitetura-mvc-e-design-patterns-observer-composite-strategy.aspx](http://www.linhadecodigo.com.br/artigo/2367/abordando-a-arquitetura-mvc-e-design-patterns-observer-composite-strategy.aspx). Acesso em: 31 de Outubro 2017.
- BELLEZA, Luciana Mesquita (2009) “Modelo para suportar a atualização e consistência de requisitos em processos de manutenção de software”. Dissertação de Mestrado da Fac. De Informática, PUCRS, Porto Alegre. Disponível em: <http://repositorio.pucrs.br/dspace/handle/10923/1681>
- CALAZANS, A. T. S., PALDÊS, R. A., OLIVEIRA, D. C., SILVA, F. G. C. M., GAMA, T. C. B., LOUZADA, W. de A., GARCIA, Y. C (2014) “O processo de identificação dos requisitos funcionais para uma ferramenta de rastreabilidade de requisitos de software”. Programa de Pós-Graduação em Engenharia de Requisitos Centro Universitário de Brasília, Brasília.
- Disponível em: <https://www.publicacoesacademicas.uniceub.br/gti/article/view/2816>
- GENVIGIR, Elias Canhadas (2009) “Um Modelo para Rastreabilidade de Requisitos de Software Baseado em Generalização de Elos e Atributos”. Tese de Doutorado do Curso de Pós-Graduação em Computação Aplicada, INPE, São Jose dos Campos.
- GRINGS, Claiton Luiz, SAIÃO, Mirian (2009) “OpenReq: uma Ferramenta para Auxílio à Gerência de Requisitos”. Pontifícia Universidade Católica do Rio Grande Do Sul, Porto Alegre.
- IEEE (1998) “Std 1219 – IEEE Standart For Software Maintenance”, Institute Of Electrical And Engineers, New York.
- MACHADO, Felipe Nery Rodrigues (2011) “Análise e Gestão de Requisitos: onde Nascem os Sistemas”. 1ª Edição. Érica, São Paulo.
- DESENVOLVIMENTO ÁGIL, “Manifesto para o Desenvolvimento Ágil de Software” (2001). Disponível em: <http://www.manifestoagil.com.br/>. Acesso em: 01 de Maio 2017.
- NEIVA, Denise Ferreira S., MATOS, Ecivaldo de Souza. (2014) “Estratégia para Substituição de Sistemas Legados”. Universidade Federal da Bahia, Salvador. Universidade Federal de Campina Grande, Campina Grande.
- GNU, “O que é Software Livre?” (2017). Disponível em: <https://www.gnu.org>. Acesso em: 01 de Maio 2017.
- OLIVEIRA, Eduardo Junior de, GUIMARÃES, Cayley. (2009) “Motivação para utilização da Extreme Programming”. Centro Universitário de Belo Horizonte, UNI-BH, Belo Horizonte.
- PRESSMAN, Roger S. (2002) “Engenharia de Software”. 5ª Edição. MacGraw-Hill.
- REIS, Christian Robottom. (2003) “Caracterização de um Processo de Software para projetos de Software Livre”. Dissertação de Mestrado, ICMC/USP, São Carlos.
- SILVA, Bruno carreira Coutinho, FALBO, Ricardo de Almeida. (2006) “Definição de um Processo Padrão para Software Livre”. Mestrado em Informática da Universidade Federal do Espírito Santo, UFES, Vitória.

- SILVA, Luiz Fernando de Carvalho, SALES, Diego da Silva, CARVALHO, Rogerio Atem. (2009) “Desenvolvimento e Manutenção de Software Ágil: Um Novo Processo”. XVI Simpósio de Engenharia de Produção, Bauru.
- SOMMERVILLE, Ian. (2005) “Engenharia de Software”. 6ª Edição. Person Addison Wesley
- WIEGERS, Karl (2003) “Software Requirements: Practical techniques for gathering and managing requirements throughout the product development cycle”. 2ª Edition. Microsoft Press, Redmond, Washington.