

Uma Análise Crítica dos Desafios para Engenharia de Requisitos em Manutenção de Software

Rodrigo Santos de Espindola, Azriel Majdenbaum, Jorge Luiz Nicolas Audy
Programa de Pós-Graduação em Ciência da Computação
Faculdade de Informática
Pontifícia Universidade Católica do Rio Grande do Sul
{respindola, audy}@inf.pucrs.br , azm57@hotmail.com

Resumo: Diversos são os desafios encontrados na manutenção de sistemas legados. Dentre estes desafios, a realização da Engenharia de Requisitos destaca-se como sendo uma área particularmente afetada pelas dificuldades envolvidas em projetos de manutenção. O objetivo deste artigo é apresentar uma análise crítica da influência das principais dificuldades encontradas na manutenção de sistemas legados sobre os processos da Engenharia de Requisitos. Como resultado demonstra-se a criticidade de algumas das atividades da Engenharia de Requisitos, quando realizadas no contexto da manutenção de software. Identificou-se também uma certa dificuldade no uso dos modelos e processos de ER quando analisados sob a ótica do mantenedor de software e de suas necessidades, principalmente no que se refere aos aspectos de rastreamento de requisitos.

Keywords: Engenharia de Requisitos, Manutenção de Software, Sistemas Legados

1. Introdução

Estudos demonstram que uma grande quantidade de projetos de software são cancelados ou fracassam por não atenderem completamente as necessidades dos clientes e excederem o prazo e o orçamento estimados. Não há uma explicação simples para este fenômeno, mas diversos trabalhos apontam deficiências nos requisitos dos sistemas como uma das principais causas de fracassos em projetos de software [1][2][3][4]. Tais constatações têm levado alguns autores a considerar a Engenharia de Requisitos (ER) como uma das mais importantes disciplinas da Engenharia de Software (ES) [5].

Por outro lado, a ES já tem cerca de trinta anos, mas muitos dos seus primeiros produtos, sistemas de software desenvolvidos nas décadas de 60 e 70, continuam sendo utilizados até hoje. Segundo [6] e [7], a manutenção e a operação de sistemas legados pode consumir a maior parte de todo o esforço despendido e do orçamento destinado aos sistemas de informação nas organizações.

Tais características da manutenção de sistemas legados tornam relevantes as pesquisas em processos de manutenção de software, especialmente na área de ER, dada sua estreita relação com a garantia ao atendimento das necessidades dos clientes e o cumprimento de prazos e orçamentos estimados, bem como as dificuldades que surgem quando é aplicada no contexto de manutenção de software. Portanto, este artigo tem como objetivo apresentar uma análise dos processos de ER quando aplicados no contexto de manutenção de sistemas legados. Com este propósito foi realizada uma profunda revisão teórica sobre ER e manutenção de software. Com base nesta revisão foi possível elaborar uma análise crítica da influência das principais dificuldades encontradas na manutenção de sistemas legados sobre os processos de ER.

Este artigo está organizado da seguinte forma: os capítulos 2 e 3 correspondem à apresentação da base teórica. O capítulo 4 apresenta a caracterização do problema. O capítulo 5 apresenta a análise crítica. Por fim, o capítulo 6 apresenta as considerações finais.

2. Engenharia de requisitos

De acordo com [2] e [3], ER é um termo que engloba todas as atividades envolvidas na descoberta, documentação e manutenção de um conjunto de requisitos para um sistema computacional. O uso do termo “engenharia” implica que técnicas sistemáticas e repetíveis devem ser utilizadas para garantir que os requisitos do sistema sejam completos, consistentes e relevantes.

O principal artefato trabalhado em ER é o documento de requisitos. De acordo com [3], o documento de requisitos é uma declaração formal dos requisitos para os *stakeholders*, que podem ser clientes, usuários finais ou a equipe de desenvolvimento do software. Dependendo da organização, o documento de requisitos pode receber diversos nomes, tais como “especificação funcional”, “definição de requisitos” ou ainda “especificação dos requisitos do software”. Algumas organizações como o Departamento de Defesa dos EUA e o IEEE definiram seus próprios padrões para o documento de requisitos. Provavelmente, o mais conhecido destes padrões seja o IEEE Std 830-1998 [8].

2.1 Processo de engenharia de requisitos

Um processo de ER é um conjunto estruturado de atividades a serem seguidas para criar, validar e manter um documento de requisitos. Poucas organizações têm um processo de ER explicitamente definido e padronizado. Entretanto, [3] sugere que cada organização deve desenvolver um processo adequado à sua realidade. Em [2] é sugerido um modelo genérico de atividades em alta granularidade que pode descrever a maioria dos processos de ER. Este modelo é apresentado na Figura 1.

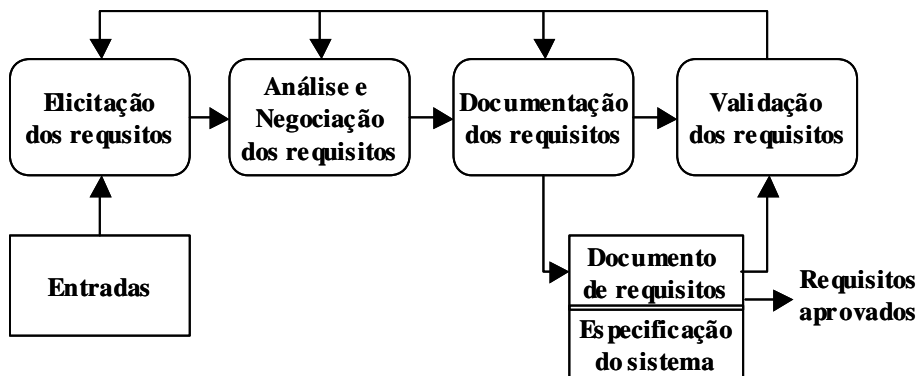


Figura 1. Modelo de atividades genérico para processos de ER [2]

As atividades apresentadas no modelo são descritas a seguir:

Elicitação dos Requisitos: nesta fase são identificadas as expectativas e necessidades dos *stakeholders* com relação ao software a ser desenvolvido;

Análise e Negociação dos Requisitos: depois de obtidos os requisitos iniciais, estes são utilizados como base para análise dos requisitos. A análise distribui os requisitos em categorias, explora as relações entre eles, e classifica a importância de cada um dos requisitos de acordo com as necessidades dos *stakeholders*. Os requisitos são negociados para decidir quais devem ser aceitos, de forma a obter consenso;

Documentação dos Requisitos: os requisitos são documentados em um nível apropriado de detalhe. Em geral é produzido um documento de especificação de requisitos, de forma que todos os *stakeholders* possam entendê-lo;

Validação dos requisitos: esta etapa examina a especificação do software, de forma a assegurar que todos os requisitos foram definidos sem ambigüidades, inconsistências ou omissões, e que todos os erros foram detectados e corrigidos.

Apesar do aparente fluxo entre as atividades, não existe uma fronteira explícita entre elas. Na prática existe muita sobreposição e interação entre uma atividade e outra. Como entradas para o processo são consideradas:

- Descrições do que os *stakeholders* necessitam para suportar suas atividades;
- Informações a respeito do sistema que será substituído ou de qualquer sistema com o qual o sistema sendo definido terá que interagir;
- Padrões vigentes na organização a respeito de práticas de desenvolvimento de sistemas, gerência de qualidade, etc.;
- Regulamentos externos, tais como leis, regulamentos de segurança ou saúde;
- Informações gerais sobre o domínio de aplicação.

Em paralelo com estas atividades está o processo de gerência dos requisitos, voltado ao endereçamento de modificações nos requisitos.

2.2 Processo de gerência de requisitos

De forma geral, a gerência de requisitos envolve a utilização de técnicas e ferramentas para gerenciamento de configuração e controle de versão, além de verificar

inconsistências nas especificações, conforme estas evoluem [9]. Para fins da análise que é apresentada neste trabalho, três aspectos da gerência de requisitos merecem atenção especial. Estes aspectos são destacados na Figura 2.

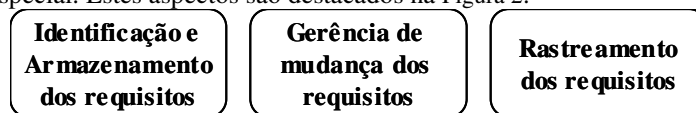


Figura 2. Aspectos importantes na gerência de requisitos

Mudanças dos requisitos são inevitáveis e não implicam necessariamente em falhas nas práticas de ER. Entretanto, segundo [2], é considerada boa prática de gerência de requisitos tentar antecipar as mudanças dos requisitos. Isto normalmente envolve classificar os requisitos para identificar os mais voláteis e analisar as possíveis mudanças que podem surgir.

Um pré-requisito essencial para permitir qualquer tipo de classificação e análise é de que todos os requisitos devem possuir algum tipo de identificador único. Segundo [2], a ausência de identificação única dos requisitos pode tornar a gerência de requisitos impraticável.

As vantagens de armazenar os requisitos em documentos de requisitos é que os requisitos estão reunidos em um único documento. Isto torna fácil o acesso aos mesmos e a confecção de novas versões do documento de requisitos. Entretanto, de uma perspectiva de ER, existem desvantagens nesta abordagem [2], pois ela dificulta a manutenção das informações relativas às dependências entre os requisitos. Para eliminar estas desvantagens os requisitos precisam ser armazenados em um banco de dados, com cada requisito representado por uma ou mais entidades de banco de dados.

Outro aspecto importante diz respeito à gerência de mudanças dos requisitos. Deve-se garantir que informações similares sejam coletadas para cada proposta de mudança de requisitos e que o julgamento seja feito com base em uma análise de custo e benefício de cada proposta. Sem uma gerência de mudanças formal não é possível garantir que as mudanças propostas suportam os objetivos de negócio fundamentais do sistema.

Além disto, de acordo com [2], os requisitos não podem ser efetivamente gerenciados sem rastreamento. Diz-se que um requisito pode ser rastreado se é possível determinar quem sugeriu o requisito, porque o requisito existe, a quais outros requisitos ele está relacionado e como ele está relacionado com outras informações como artefatos de projeto, implementação e documentação de usuário. O rastreamento também permite encontrar outros requisitos que podem ser afetados quando uma mudança é solicitada.

3. Manutenção de software

A manutenção de software é reconhecida por vários autores [5][7][18] como a atividade que demanda o maior volume de esforço dentre todas as atividades de engenharia de software. Muitos dos sistemas dos quais as organizações dependem atualmente foram desenvolvidos há décadas. Desde então, estes sistemas foram

migrados para novas plataformas, ajustados devido a mudanças nos equipamentos e nos sistemas operacionais e melhorados para atender novos requisitos funcionais. O resultado são aplicações mal estruturadas, mal codificadas e fracamente documentadas [5].

A manutenção de software é definida por [8] como a modificação de um produto de software depois de sua entrega (ao cliente) para corrigir erros, melhorar sua performance ou qualquer outro atributo, ou para adaptar o produto a um ambiente modificado. Este processo normalmente é desencadeado por uma solicitação do cliente ou por algum relatório de problemas gerado pelo usuário, sendo que este tipo de solicitação é identificada pelo termo genérico de Requisição de Modificação [8][11].

3.1 Sistemas Legados

A manutenção de software pode ser necessária mesmo em sistemas que acabaram de ser entregues ao seu usuário. Entretanto, é a atividade de manutenção em sistemas legados que apresenta os maiores desafios.

É possível observar na literatura uma convergência quanto às características dos sistemas legados. Para [12] um sistema legado é um sistema de missão crítica, desenvolvido em algum momento do passado, que ainda é usado e vem sendo modificado ao longo do tempo sem submeter-se a ações sistemáticas de melhoria. Para [13] são aqueles sistemas que possuem valor para a organização e ainda resistem às modificações e a evolução para adequar-se a requisitos de negócio em constante mudança. Neste artigo será adotada a definição de [5], que considera um sistema legado como um sistema antigo, frequentemente mal projetado ou documentado, mas que é crítico para o negócio e deve ser suportado por vários anos.

É importante também destacar que [6] define um sistema crítico como sendo essencial para a continuidade dos negócios da organização e, portanto, o sistema deve estar permanentemente operacional. O risco envolvido na substituição de um sistema crítico para o negócio é alto, o que leva a organização a optar por sua manutenção. Isto explica porque sistemas voltados a atividades meio não se tornam legados, enquanto que sistemas voltados a atividades fim tendem a se tornarem sistemas legados.

4. Caracterização do Problema

De acordo com [14], à medida que os negócios expandem-se e os processos de negócio mudam, muitos sistemas de informação tornam-se inadequados em termos de capacidade e funcionalidades. Porém, muitas organizações necessitam manter estes sistemas, conhecidos como sistemas legados, para fornecer informações críticas e manter suas operações. Portanto, estes sistemas não podem ser simplesmente descartados, mas necessitam ser melhorados e integrados à infraestrutura organizacional de informação.

Segundo [14], compreender os requisitos de sistemas legados torna-se importante para melhorar o sistema legado, integrá-lo com o restante dos sistemas de informação

ou realizar a reengenharia do sistema. Entretanto, a falta de precisão da documentação do sistema ou a inexistência desta documentação e a indisponibilidade dos *stakeholders* originais tornam o trabalho caro e difícil. Este cenário é muito comum na prática, conforme apresentado por diversos autores [15][14][5][16][10]. Em muitas circunstâncias o conhecimento preciso sobre o sistema foi há muito perdido. Nestes casos o sistema em si, seus códigos-fonte e aqueles que o usam são as únicas fontes de informação sobre o que e porque o sistema faz.

Um outro fator importante a ser observado é que a ER é geralmente discutida como uma fase inicial no desenvolvimento de software [10]. Para [2], por exemplo, requisitos são definidos nos estágios iniciais do desenvolvimento de software como uma especificação do que precisa ser implementado. Esta definição em parte está correta, mas vincula os requisitos do software ao escopo do projeto de desenvolvimento do software, não contemplando os projetos de manutenção de software.

Entretanto, como destaca [10], o conhecimento dos requisitos requer um esforço contínuo no refinamento progressivo das exigências contidas nas regras de negócio das organizações e em atendimento às necessidades dos *stakeholders*, durante todo o ciclo de vida do software. Uma definição mais adequada é fornecida por [17], onde requisitos são características do software necessárias para o usuário solucionar um problema de forma a atingir um objetivo ou ainda uma característica de software que deve ser realizada ou possuída por um sistema ou componente de sistema para satisfazer um contrato, padrão, especificação ou outra documentação formalmente imposta. Esta definição é mais consistente com as necessidades de projetos de manutenção de software, pois vincula os requisitos ao escopo do produto em si, independentemente da fase do ciclo de vida em que o software se encontra.

Em vista disto, este trabalho destaca como as principais dificuldades encontradas em ER em projetos de manutenção de software, conforme ilustrado na Figura 3: a ausência de documentação, a indisponibilidade dos *stakeholders* originais e os requisitos vinculados aos projetos ao invés do produto.

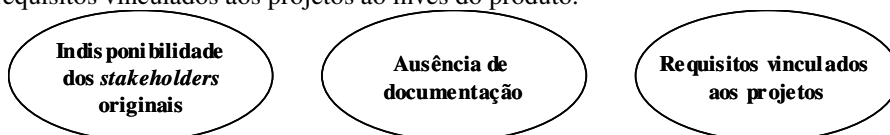


Figura 3. Principais dificuldades encontradas em ER em projetos de manutenção de software

Portanto, o problema em questão é identificar qual o impacto destes fatores sobre a ER. Este problema será tratado, no próximo capítulo, através da análise crítica da influência de cada uma das dificuldades destacadas sobre os processos de ER.

5. Análise Crítica

Os fatores destacados na caracterização do problema tornam particularmente difícil, em projetos de manutenção de software, o trabalho de ER, dificultando várias atividades relacionadas aos processos descritos no capítulo 2. Estas dificuldades têm

impacto direto no sucesso de tais projetos, bem como na qualidade dos produtos resultantes do processo de manutenção.

Com base nesta realidade, a seguir serão apresentados os principais resultados desta pesquisa, visando analisar como as dificuldades apresentadas na Figura 3 afetam a manutenção de sistemas legados nas organizações. Foram analisadas as atividades do processo de ER, tal como apresentadas na seção 2.1, e as características do processo de gerência de requisitos, apresentadas na seção 2.2. Esta análise foi realizada à luz do conhecimento acumulado sobre manutenção de software, presente na literatura.

Este estudo é de natureza qualitativa, sendo o principal método de pesquisa o estudo de caso. A pesquisa encontra-se no final da primeira fase, teórico-reflexiva, buscando aliar uma profunda pesquisa teórica em ER na manutenção de sistemas legados ao conhecimento prático dos pesquisadores nesta área. Na segunda fase, de base empírica, se buscará testar e validar os resultados obtidos nesta primeira fase, por meio do desenvolvimento de estudos de caso múltiplos visando ampliar a validade externa do estudo.

5.1 Dificuldades no Processo de Engenharia de Requisitos

Esta seção apresenta a análise de impacto no processo de ER das dificuldades descritas no capítulo 4. A Figura 4 ilustra a relação entre as dificuldades encontradas na manutenção de software e as atividades do processo de ER. As próximas seções irão descrever o impacto destas dificuldades sobre cada uma das atividades do processo de ER.

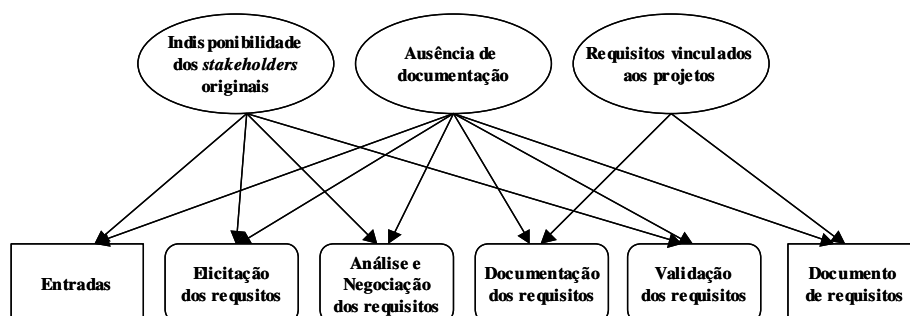


Figura 4. Impacto das dificuldades da manutenção sobre o processo de ER

5.1.1 Impacto na Atividade de Elicitação dos Requisitos

Na seção 2.1 foram descritas algumas informações consideradas como entradas para a elicitação de requisitos. No contexto de projetos de manutenção de software, alguns trabalhos [11][14][16][10] destacam a importância dos requisitos do sistema legado como ponto de partida para a compreensão do sistema atual e o correto levantamento de novos requisitos. Entretanto, a imprecisão ou ausência de documentação adequada, bem como a indisponibilidade das pessoas que desenvolveram o sistema, podem

impedir a recuperação dos requisitos originais do sistema e dificultar a tarefa de elicitação dos novos requisitos.

Muitas vezes a única alternativa para a recuperação dos requisitos do sistema é a pesquisa no código-fonte. Entretanto, recuperar os requisitos de um sistema a partir do código-fonte é difícil, visto que é preciso reconhecer decisões tomadas pelos projetistas e distinguir estas decisões dos requisitos reais.

5.1.2 Impacto na Análise e Negociação dos Requisitos

Durante a atividade de análise busca-se identificar requisitos esquecidos, conflitantes, ambíguos, sobrepostos e irreais. Além disto, em projetos de manutenção de software, é necessário que esta análise contemple a descoberta de conflitos, sobreposições e inconsistências tanto entre os requisitos do projeto de manutenção quanto entre estes e os requisitos originais do sistema, caso existam ou possam ser recuperados. Entretanto, a atividade de negociação com os *stakeholders* pode ser prejudicada, pois alguns ou todos os *stakeholders* responsáveis pelos requisitos originais podem não estar mais disponíveis para negociação. Neste caso pode ser difícil esclarecer as motivações por trás destes requisitos, tornando a tarefa de resolução de eventuais conflitos particularmente delicada e sujeita a erros. Este risco também existe em projetos de desenvolvimento de software, pois qualquer um dos *stakeholders* pode deixar o projeto ou a organização a qualquer momento. Mas este risco é maximizado em projetos de manutenção, tendo em vista que o período de uso de um sistema e, portanto, de manutenção do mesmo, tende a ser significativamente maior que o seu tempo de desenvolvimento.

A análise e negociação dos requisitos podem tornar-se ainda mais difíceis caso os requisitos originais não estejam disponíveis ou não sejam recuperados durante a elicitação dos requisitos. Neste caso não é possível verificar possíveis inconsistências entre a modificação proposta e os objetivos originais do sistema. Um dos Fatores Críticos de Sucesso (FCS) em manutenção de software propostos por [18] estabelece que a operação de manutenção deve pelo menos preservar, senão melhorar, a funcionalidade do sistema em manutenção. O usuário não deve receber, após a manutenção, menos funcionalidades do que tinha antes dela. O que significa que cada novo release deve ter, pelo menos, as mesmas funcionalidades e dados presentes nos *releases* anteriores, enquanto elas ainda forem necessárias. Entretanto, é difícil garantir durante a análise de requisitos que a modificação proposta não viola este FCS sem conhecer os requisitos funcionais originais do sistema. Uma alternativa para realizar esta verificação, na ausência dos requisitos, seria realizar testes comparativos de todas as funcionalidades apresentadas pelo sistema antes e depois da modificação, o que pode ser um procedimento caro e ineficiente.

5.1.3 Impacto na Documentação dos Requisitos

O principal impacto na documentação dos requisitos está relacionado à forma como são documentadas as dependências entre os requisitos definidos no projeto de

manutenção e os requisitos originais do sistema. Dependendo da abordagem utilizada para lidar com esta questão, alguns problemas podem surgir:

- Caso estas dependências não sejam documentadas os novos requisitos, bem como as modificações solicitadas, podem ficar fora de contexto. O que pode dificultar sua interpretação pelos responsáveis pela análise, projeto, implementação e testes das modificações no software;
- Caso o contexto seja documentado diretamente no documento de requisitos, sem o devido cuidado para diferenciar o que deve ser modificado do que já está implementado, pode-se dificultar as estimativas do projeto, além de confundir os responsáveis pela análise, projeto, implementação e testes das modificações no software.

É necessária uma documentação adequada das dependências e do rastreamento entre os requisitos para garantir que o projeto seja corretamente dimensionado e que as modificações propostas fiquem claramente contextualizadas como parte de um sistema já implementado.

Além disto, a atividade de documentação torna-se crítica para a ER se considerarmos os requisitos como propriedades do produto e não de um dos projetos, sejam eles de desenvolvimento ou de manutenção. Neste caso os requisitos precisam ser documentados de forma que possam facilmente referenciar e ser referenciados por outros artefatos, independentemente do projeto que deu origem ao artefato ou ao requisito. Desta forma será possível analisar o impacto na modificação de qualquer artefato ou componente do sistema até sua origem nos requisitos do sistema e vice-versa.

Estas características estão fortemente relacionadas às questões de rastreamento de requisitos, que são abordadas em mais detalhes na seção 5.2, que analisa o impacto dos problemas decorrentes da manutenção de software na gerência de requisitos.

5.1.4 Impacto na Validação dos Requisitos

Devido à semelhança com a atividade de análise, a validação de requisitos sofre o mesmo impacto descrito na seção 5.1.2, quando realizada no contexto de manutenção de software. O sucesso desta atividade está intimamente ligado com o sucesso da atividade de análise, pois os problemas decorrentes de documentação inadequada ou ausente e de indisponibilidade dos *stakeholders* originais que não são resolvidos durante a análise também não são resolvidos durante a validação.

Além disto, as abordagens baseadas na criação de casos de teste podem ser prejudicadas. Um dos FCS propostos por [18] estabelece que a operação de manutenção deve preservar, senão melhorar, a qualidade do sistema em manutenção. Entretanto, é difícil verificar se a qualidade do sistema foi preservada após a modificação, pois sem conhecer os requisitos originais do sistema não é possível planejar testes adequados, principalmente testes de regressão. Novamente, uma alternativa seria realizar testes comparativos de todas as funcionalidades apresentadas pelo sistema antes e depois da modificação, o que neste caso também pode ser um procedimento caro e ineficiente.

5.2 Dificuldades na Gerência de Requisitos

Esta seção apresenta a análise de impacto das dificuldades descritas no capítulo 4 sobre a gerência de requisitos, com exceção da indisponibilidade dos *stakeholders* originais, pois esta dificuldade não parece ter efeito direto sobre as características da gerência de requisitos.

Esta análise será descrita para cada uma das características apresentadas na seção 2.2. A Figura 5 ilustra o impacto das dificuldades na manutenção de software sobre a gerência de requisitos.

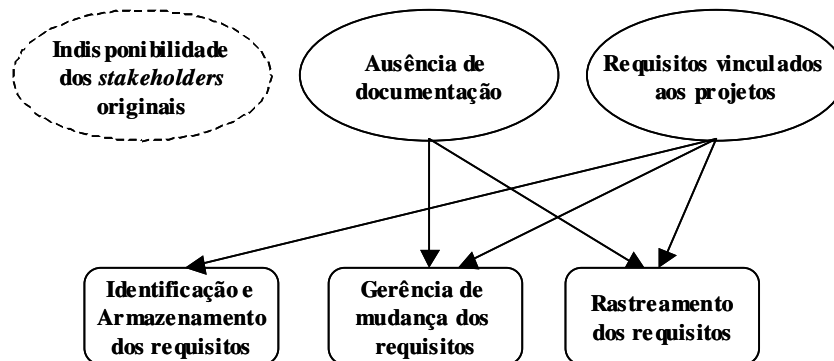


Figura 5. Impacto das dificuldades na manutenção de software sobre a gerência de requisitos

5.2.1 Impacto na Identificação e Armazenamento dos Requisitos

A identificação e o armazenamento dos requisitos podem ser prejudicados em abordagens onde os requisitos são vinculados ao escopo do projeto, ao invés do produto. Um exemplo dos prejuízos desta abordagem é a prática, muito comum segundo [2], de numerar os requisitos de acordo com o capítulo ou seção do documento de requisitos onde eles se encontram.

Como foi visto na seção 2.2, um pré-requisito essencial para a gerência de requisitos é de que todos os requisitos devem possuir algum tipo de identificador único. Mas o emprego da prática citada garante apenas a unicidade de identificação dentro de um mesmo documento de requisitos. Não garante a unicidade de identificação ao longo do ciclo de vida completo do sistema, pois cada projeto pelo qual o produto passa, seja de desenvolvimento ou de manutenção, pode gerar seu próprio documento de requisitos. Tornando assim impraticável o gerenciamento dos requisitos ao longo de diversos projetos.

Duas alternativas que possibilitam a identificação única são o uso de identificação baseada em banco de dados e de identificação simbólica baseada no tipo de requisito, pois nestes casos as identificações não estão vinculadas ao escopo do documento ou do projeto. Assim é possível criar relações de dependência e referenciar requisitos tanto dentro de um mesmo documento de requisitos quanto entre os documentos de diversos projetos de manutenção do mesmo produto.

5.2.2 Impacto na Gerência de Mudança dos Requisitos

A gerência de mudanças dos requisitos pode ser prejudicada em circunstâncias onde não está disponível uma documentação adequada dos requisitos do sistema, pois não é viável tentar gerenciar mudanças em algo que é desconhecido. A seguir serão descritas as dificuldades que podem surgir nestes casos em cada uma das etapas da gerência de mudança dos requisitos, tal como sugeridas por [2]:

Análise do problema: a análise dos requisitos originais não pode ser realizada nestes casos. Em alguns casos será difícil até mesmo determinar se existe realmente um problema ou se o comportamento apresentado pelo sistema está de acordo com os requisitos originais do sistema, os quais não estão disponíveis para análise.

Análise e orçamento das modificações: para realização satisfatória desta análise são necessárias informações que permitam identificar os requisitos diretamente e indiretamente afetados pela modificação proposta. A ausência das informações citadas pode levar a orçamentos incorretos, dado que não será possível determinar com precisão o impacto da modificação sobre as especificações do sistema.

Implementação das modificações no documento de requisitos: na ausência da documentação original, não existe o que ser modificado. A implementação das modificações dos requisitos ficará restrita a descrever qual o comportamento apresentado pelo sistema que se pretende alterar.

Além disto, mesmo que a documentação exista, a gerência de mudança dos requisitos poderá ser prejudicada caso os requisitos estejam vinculados ao escopo dos projetos, ao invés do produto. Neste caso pode ser difícil determinar as dependências entre os requisitos estabelecidos em mais de um projeto, limitando o potencial da análise de impacto das modificações e reduzindo a precisão dos orçamentos estimados para a realização das modificações propostas.

5.2.3 Impacto no Rastreamento dos Requisitos

Como vem sendo apresentado ao longo deste capítulo, o aspecto mais afetado de ER no contexto de manutenção de software é o rastreamento de requisitos, justamente pela importância que esta informação tem para a análise de impacto de qualquer modificação proposta em um sistema.

Idealmente, o mantenedor deveria ser capaz de determinar, para cada manutenção solicitada, quais os requisitos direta e indiretamente afetados pela modificação. A seguir seria necessário determinar qual a origem de cada um destes requisitos (*backward traceability*) para estender a modificação até estas fontes. Por fim, seria necessário determinar todos os artefatos afetados pela mudança de cada uma das informações afetadas pela modificação (*forward traceability*), sejam estas informações representadas por requisitos, requisitos dependentes ou as origens dos requisitos.

Entretanto, muitas das informações necessárias para realizar esta análise não existem nos casos em que não estão disponíveis os documentos de requisitos ou os requisitos estão documentados apenas no escopo dos projetos. A ausência destas informações pode gerar dificuldades para:

- Estimar o custo da manutenção;
- Estimar o tempo necessário para realizar a manutenção;
- Estimar o risco envolvido na manutenção;
- Garantir o atendimento aos FCS em projetos de manutenção, tal como sugeridos por [18];
- Planejamento e execução de testes de regressão no sistema modificado.

Entretanto, [2] reconhece que coletar e manter as informações de rastreamento pode ser um processo caro. Por isto, sugere que as organizações devem estabelecer um conjunto de políticas de rastreamento definindo quais informações são necessárias para permitir o correto gerenciamento dos requisitos. Além disto, sugere o uso de ferramentas de software para permitir o gerenciamento da enorme quantidade de informações que podem ser necessárias. Existem no mercado diversas ferramentas CASE com as funcionalidades necessárias, bem como ferramentas especializadas em gerência de requisitos, que podem dar suporte ao processo.

6. Considerações Finais

A manutenção de software é considerada uma fase de grande importância em engenharia de software. Esta fase ocupa a maior parte do ciclo de vida de software, sendo fundamental para garantir a qualidade geral do produto à medida que as necessidades de seus usuários e do ambiente de negócios evoluem. Como a compreensão das necessidades dos *stakeholders* é fundamental para qualquer tipo de projeto de software, a ER torna-se uma área de pesquisa de grande relevância para a manutenção de software.

Buscando contribuir com a pesquisa na área de ER, este estudo apresenta uma análise crítica visando identificar os principais desafios encontrados na ER no contexto de projetos de manutenção de software. Esta análise demonstrou a criticidade de algumas das atividades da ER, quando realizadas em um contexto tão delicado e importante como a manutenção de software. Identificou-se também uma certa dificuldade no uso dos modelos e processos de ER quando analisados sob a ótica do mantenedor de software e de suas necessidades, principalmente no que se refere aos aspectos de rastreamento de requisitos.

Na próxima etapa desta pesquisa pretende-se testar e validar os resultados apresentados neste artigo, por meio do desenvolvimento de estudos de casos múltiplos em projetos de manutenção de sistemas legados de uma organização multinacional de tecnologia da informação.

Referências

- [1] Leffingwell, Dean; Widrig, Don. “Managing Software Requirements – A Use Case Approach”. Addison-Wesley. 2003, 492p.
- [2] Kotonya, G.; Sommerville, I. “Requirements Engineering: process and techniques”. New York: John Wiley & Sons Ltd, 1998, 282p.
- [3] Sommerville, Ian; Sawyer, Peter. “Requirements Engineering – a good practice guide”. New York: John Wiley & Sons Ltd, 1997, 391p.

- [4] The Standish Group International. "Chaos Report". http://www.standishgroup.com/sample_research/index.php. (Visualizado em 27 de julho de 2004). 1995.
- [5] Pressman, Roger S. "Software Engineering: a practitioner's approach". New York: McGraw Hill, 5th ed., 860p.
- [6] Brodie, M. L.; Stonebraker, M. "Migrating Legacy Systems: Gateways, Interfaces and the Incremental Approach". San Francisco: Morgan Kaufmann Publishers, Inc. 1995, 210p.
- [7] Hanna, M. "Maintenance Burden Begging for Remedy". Software Magazine, April 1993, pp. 53-63.
- [8] IEEE, Institute. "IEEE Std 1219-1998. IEEE Standard for Software Maintenance". New York: Institute of Electrical and Electronic Engineers. Inc., 1998, 52p.
- [9] Nuseibeh, Bashar; Easterbrook, Steve. "Requirements Engineering: A Roadmap". ACM - Future of Software Engineering. 2000. pp 37-45.
- [10] Zanolenci, Edna P.; Burnett, Robert C. "Abordagem de Engenharia de Requisitos em Software Legado". Anais do WER03 - Workshop em Engenharia de Requisitos, Piracicaba-SP, Brasil, 2003, pp 270-284.
- [11] ISO. "ISO/IEC 14764 – Software Maintenance". Geneva: International Organization for Standardization, 1999, 38p.
- [12] Lucia, Andrea De; Fasolino, Anna R.; Pompella, Eugenio. "A Decisional Framework for Legacy System Management". In: International Conference on Software Maintenance, 2001, 10p.
- [13] Stevens, P.; Pooley, R. "Systems Reengineering Patterns". In: Proceedings of the 6th ACM SIGSOFT international symposium on Foundations of software engineering, vol. 23 issue 6, 1998, pp. 17-23.
- [14] Liu, Kecheng; Alderson, Albert; Qureshi, Zubair. "Requirements Recovery from Legacy Systems by Analysing and Modelling Behaviour". Proceedings of the International Conference on Software Maintenance, 1999, IEEE Computer Society, Los Alamitos, pp3-12.
- [15] Ebner, Gerald; Kaindl, Hermann. "Tracing All Around in Reengineering". IEEE Software, May 2002, pp.70-76.
- [16] White, Stephanie M. "Capturing Requirements for Legacy Systems". Proceedings of the International Symposium and Workshop on Systems Engineering of Computer Based Systems, 1995. pp 251-256.
- [17] Thayer, Richard; Dorfman, Merlin. "System and Software Requirements Engineering". IEEE Computer Society Press Tutorial. 1990. 718p
- [18] Sneed, H.M. "Critical Success Factors in Software Maintenance". In: International Conference on Software Maintenance, 2003, 9p.