

Especificação de Requisitos em Ambientes Geograficamente Dispersos – Um Estudo Comparativo

Leandro Teixeira Lopes¹, Jorge Luiz Nicolas Audy¹

¹Faculdade de Informática – Pontifícia Universidade Católica do Rio Grande do Sul
Porto Alegre – RS – Brazil

{lteixeira, audy}@inf.pucrs.br

***Resumo.** A crescente tendência de distribuição do processo de desenvolvimento tem afetado diversas áreas da engenharia de software. A engenharia de requisitos, por ser um processo que exige grande volume de comunicação e compreensão, sofre influência direta de fatores como linguagem, contexto e cultura. Entretanto, ainda existem poucos estudos sobre a engenharia de requisitos para ambientes de desenvolvimento distribuído de software. Buscando contribuir neste sentido, este estudo apresenta resultados de uma pesquisa em engenharia de requisitos em ambientes dispersos envolvendo grupos de alunos de dois países.*

1. Introdução

A crescente globalização do ambiente de negócios tem afetado diretamente o mercado de desenvolvimento de software [3]. Em busca de vantagens competitivas em termos de custos, produtividade ou qualidade na área de desenvolvimento de sistemas [7], por exemplo, diversas organizações optaram por distribuir o processo de desenvolvimento de software dentro de seu país, ou em outros países, como a Índia e o Brasil. Estas regiões oferecem, muitas vezes, incentivos fiscais ou possuem grande concentração de massa crítica em determinadas áreas.

Embora o desenvolvimento de software tenha evoluído consideravelmente nas últimas décadas, ainda são enfrentadas diversas dificuldades. Muitos projetos de software têm sido entregues atrasados ou com custos superiores ao esperado. Frequentemente, o software não realiza as funcionalidades desejadas por seus usuários. Ao optar por instanciar um ambiente de desenvolvimento distribuído, além de todas as dificuldades inerentes ao processo de desenvolvimento co-localizado, uma organização começa a enfrentar diversos desafios de adaptação, diferenças culturais, planejamento do trabalho, comunicação, entre outros [5].

No processo de desenvolvimento de software, a engenharia de requisitos (ER) destaca-se como um ponto fundamental para o sucesso dos projetos. Estudos mostram que as causas mais citadas como desafios aos projetos de software são relacionados com requisitos [11]. Uma incorreta engenharia de requisitos pode exercer um impacto negativo em um projeto de software, como a necessidade de um novo ciclo de especificação, projeto, codificação e teste, afetando diretamente os custos e prazos envolvidos. Em ambientes de desenvolvimento distribuído de software (DDS), dificuldades como distância, comunicação e cultura causam um aprofundamento dos problemas inerentes ao processo de engenharia de requisitos, que adquire um caráter ainda mais crítico [13].

Nesse sentido, este estudo apresenta resultados de uma pesquisa conduzida com alunos de pós-graduação de dois países, voltada a avaliar se a utilização de um processo de engenharia de requisitos tende a gerar especificações de requisitos de maior qualidade, quando em ambientes de desenvolvimento distribuído de software. Os alunos de cada localidade envolvida foram divididos em quatro grupos, onde dois deles receberam treinamentos voltados aos papéis que desempenhariam. Os alunos da Universidade de Illinois, em Chicago, Estados Unidos, realizaram o papel de clientes, e os alunos da PUCRS realizaram o papel de engenheiros de requisitos.

Este artigo se estrutura da seguinte forma: a seção 2 apresenta a base teórica, a seção 3 contextualiza a pesquisa. Na seção 4 é detalhada a metodologia de pesquisa. Em seguida, na seção 5 é realizada a análise dos dados. Na seqüência são apresentadas a análise crítica (seção 6) e considerações finais (seção 7).

2. Base Teórica

2.1. Engenharia de Requisitos

O sucesso no desenvolvimento de um software é medido principalmente pela forma com que ele realiza a tarefa para qual foi proposto [8]. O esforço de desenvolvimento é total ou parcialmente desperdiçado se o software, por melhor que seja a qualidade de sua codificação, não cumpre com a tarefa que foi destinado. Da mesma forma, se a base tecnológica (hardware, software e dispositivos) necessária ao software em questão não for compatível com a base existente onde ele será utilizado, todo (ou a maior parte) do trabalho de desenvolvimento pode se tornar inútil.

Para que o sucesso possa ser atingido, é fundamental que seja realizada uma tarefa de identificação e documentação das necessidades e propósitos de um software. Essa tarefa, muitas vezes, exige uma compreensão do ambiente onde o software será inserido, considerando as características do negócio, as possíveis modificações futuras e as necessidades reais envolvidas no processo.

No processo de desenvolvimento de software, a engenharia de requisitos destaca-se como um ponto fundamental para o sucesso dos projetos. Estudos mostram que as causas mais citadas como desafios aos projetos de software são relacionadas a requisitos [11]. Requisitos podem ser definidos como:

Requisitos são capacidades que um usuário necessita para resolver um problema ou atingir um objetivo [12].

Uma especificação de requisitos incorreta pode causar problemas como a necessidade de um novo ciclo de especificação, projeto, codificação e teste, afetando diretamente os custos e prazos envolvidos. A utilização de um processo consistente de engenharia de requisitos é a melhor maneira de evitar estes problemas [9]. A engenharia de requisitos pode ser definida como:

A engenharia de requisitos é a ciência e disciplina preocupada com a análise e documentação dos requisitos, incluindo análise das necessidades e análise e especificação dos requisitos, fornecendo mecanismos apropriados para facilitar as diversas atividades relacionadas [12].

2.2. Desenvolvimento Distribuído de Software

Nos últimos anos, o software se tornou um componente vital de negócios. O sucesso de uma organização cada vez mais depende da utilização do software como um diferencial competitivo. Ao mesmo tempo, a economia tem convertido os mercados nacionais em mercados globais, criando novas formas de competição e colaboração [3].

Entretanto, o mercado global de software vinha passando por diversas crises. Por um lado, um grande número de falhas em projetos. De outro, uma demanda crescente, atingida pela escassez de recursos capacitados. Nesse contexto, muitas organizações perceberam o desenvolvimento distribuído de software como uma alternativa, experimentando o desenvolvimento em locais remotos.

Diversas são as razões para a utilização do DDS. Além da demanda e baixos custos, podemos citar escala, *time-to-market*, sinergia cultural, entre outros [1][5]. Essas razões, ou um subconjunto delas, motivam um crescente número de organizações a desenvolverem software de forma distribuída.

O desenvolvimento distribuído de software, quando atinge proporções globais é chamado de desenvolvimento global de software (GSD – *Global Software Development*). As organizações responsáveis pelo desenvolvimento, quando localizadas em um país diferente daquele que contrata o serviço são chamadas de organizações *offshore*.

2.3. Engenharia de Requisitos em Ambientes de Desenvolvimento Distribuído de Software

O DDS promove um aprofundamento das dificuldades inerentes ao processo de desenvolvimento de software, ou ainda, o surgimento de novas dificuldades, como diferenças culturais e linguagem [13].

Como a distância envolvida pode compreender países distantes, comumente, a linguagem e a cultura são diferentes. Com isso, os problemas causados por ambigüidade e falta de clareza nos requisitos são intensificados. A compreensão dos requisitos ao serem lidos em uma língua diferente da nativa é mais limitada, levando a interpretações incorretas. Diferenças culturais como atitude em relação à hierarquia, riscos e valores culturais podem ampliar a possibilidade de conflitos. Sem o conhecimento presencial do ambiente onde o software será inserido, a compreensão das razões e expectativas do software por parte da equipe de desenvolvimento é reduzida [2].

A comunicação também apresenta novos desafios. Com a perda de contato face-a-face entre a equipe de desenvolvimento e os usuários, existe uma maior dificuldade de esclarecimento em caso de dúvidas. Além disso, com a utilização de meios de baixo contexto, como correio eletrônico, o contato informal entre os membros dos diversos grupos é limitado, reduzindo a confiança entre eles.

Em casos de grupos separados por diversos fusos horários, em geral, ocorre uma maior demora na tomada de decisões. Uma simples troca de mensagens por correio eletrônico para esclarecimento de um requisito pode levar dias se os horários de trabalho não coincidirem.

Reuniões por vídeo ou teleconferência podem não ser tão efetivas. Grupos com reduzida confiança tendem a evitar comprometimentos. Algumas culturas valorizam

mais questões como pontualidade e agenda que outras, ampliando as possibilidades de conflitos.

A Tabela 1 apresenta uma síntese das principais dificuldades identificadas na engenharia de requisitos em ambientes de desenvolvimento global de software. Muitos destes desafios ocorrem na engenharia de requisitos em ambientes co-localizados, entretanto a distância tende a exacerbá-los, ampliando o impacto causado.

Tabela 1 – Principais desafios identificados na ER em ambientes de GSD (extraído de Damian (2003)[2])

Diferenças na cultura e negócios do cliente
Participação adequada dos usuários do sistema
Consciência do contexto de trabalho local e comunicação informal
Relações de confiança no trabalho
Gerenciamento de conflitos e discussões abertas de interesse
Entendimento comum dos requisitos
Encontros efetivos
Demora

3. Contexto

3.1 Engenharia de Requisitos em Ambientes de Desenvolvimento Distribuído de Software

Este trabalho se desenvolve em um contexto de distância física entre membros da equipe de engenheiros de requisitos e o grupo de usuários/clientes. A dispersão entre as equipes pode ser representada utilizando uma adaptação do formato proposto por Prikladnicki [10], conforme o exemplo de cenário na figura 1.

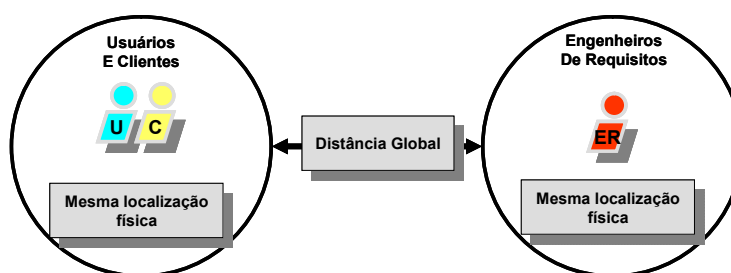


Figura 1 –Exemplo de cenário de dispersão considerado

Os principais grupos envolvidos no processo de engenharia de requisitos em ambientes de desenvolvimento distribuído de software são a equipe de engenheiros de requisitos, o grupo de usuários e clientes, e a equipe de desenvolvimento.

A **equipe de engenheiros de requisitos (ER)** é responsável pela elicitación, análise, negociação, documentação, validação e gerência dos requisitos.

O **grupo de usuários (U) e clientes (C)** representa as pessoas físicas ou jurídicas que solicitaram e contrataram o projeto, bem como os responsáveis pela utilização do produto gerado. Esta equipe fornece informações para especificação do software.

A **equipe de desenvolvimento (D)** representa os envolvidos no desenvolvimento de um determinado projeto, utilizando com entrada os requisitos especificados pelo grupo de engenheiros de requisitos. Esta equipe pode envolver gerentes de projeto, codificadores, testadores, controladores de qualidade, equipe de suporte a ferramentas, entre outros.

3.2 Definição do Problema

Tendo por base o contexto de dispersão física e diferença cultural entre as equipes envolvidas no processo de engenharia de requisitos no DDS, definimos a seguinte questão de pesquisa: Qual a contribuição obtida pela utilização de um processo de engenharia de requisitos na qualidade da especificação gerada, em um ambiente de equipes dispersas?

Nesse sentido, temos como premissa que com a utilização de um processo de engenharia de requisitos a qualidade da especificação produzida tende a aumentar, quando em ambientes de desenvolvimento distribuídos de software.

4. Metodologia

Esta pesquisa foi desenvolvida através de um trabalho conjunto entre uma turma de pós-graduação em ciência da computação da PUCRS, em Porto Alegre, Brasil, e uma turma de pós-graduação em sistemas de informação da universidade de Illinois, em Chicago, EUA. Os estudantes de cada uma das turmas foram divididos em 4 grupos, chamados, no Brasil, de grupos 1, 2, 3 e 4. Os grupos na universidade de Illinois foram chamados de grupos A, B, C e D.

Com base nessa divisão dos grupos, foi realizada uma especificação de requisitos distribuída, com os estudantes brasileiros no papel de analistas e os estudantes americanos no papel de usuários. A interação entre os grupos foi conduzida através do módulo de discussão eletrônica do software *Blackboard*. Os estudantes foram instruídos a não realizarem comunicação por outros meios, como e-mail e telefone. O sistema a ser especificado tinha como objetivo controlar o fluxo de informações relacionadas ao processo de submissão e avaliação de artigos para eventos (como congressos, workshops, etc.).

Ao final do trabalho, todos os grupos preencheram um documento de lições aprendidas, incluindo questões sobre pontos fortes e fracos do trabalho, comunicação, avaliação da qualidade do artefato de requisitos gerado e qualidade do processo de engenharia de requisitos utilizado.

Com relação à qualidade das especificações geradas, foi realizada uma auto-avaliação das especificações pelos grupos de engenheiros de requisitos. A auto-avaliação consistia de um questionário em escala Lickert (de 1 a 5) visando capturar a percepção dos respondentes quanto às características de um bom SRS, conforme [4].

Além disso, foram utilizadas questões abertas visando capturar as dificuldades enfrentadas pelos grupos (respondidas por todos os grupos), bem como as vantagens e desvantagens da utilização do processo definido (respondido apenas pelos grupos 1 e 2).

4.1 Treinamento

Dois grupos em cada localidade foram selecionados para receberem treinamentos. Estes receberam treinamento sobre colaboração em ambientes de desenvolvimento distribuído de software. Este treinamento teve como foco ampliar o conhecimento dos envolvidos com relação às dificuldades de comunicação e entendimento devido à distribuição. Além disso, são exploradas as causas para estas dificuldades, bem como formas de compartilhamento de contexto como meio de reduzir seu impacto.

Os grupos 1 e 2 receberam, além do treinamento sobre colaboração em ambientes de desenvolvimento distribuído de software, um treinamento sobre engenharia de requisitos em ambientes de desenvolvimento distribuído de software. Com isso, foi estabelecido o processo de engenharia de requisitos a ser utilizado pelos analistas, incluindo um modelo de especificação de requisitos. O modelo é constituído por uma série de definições utilizadas para classificar e relacionar as informações obtidas, bem como um conjunto de estruturas de frases visando ampliar a clareza dos requisitos. Ao final do treinamento foi fornecido um protótipo de ferramenta para auxiliar o processo de especificação de requisitos e um modelo de documento de especificação de requisitos.

5. Análise dos Dados

Os quatro grupos brasileiros desenvolveram especificações de requisitos com base nas interações realizadas com seus pares americanos. A Tabela 2 apresenta uma síntese das informações gerais a respeito das especificações geradas.

Tabela 2 - Dados gerais sobre as especificações criadas

	Grupo 1	Grupo 2	Grupo 3	Grupo 4
Idioma	Inglês	Inglês	Português	Português
Nº de páginas	27	17	19	23
Nº de palavras	6023	3101	1751	1941
Forma de representação	Linguagem Natural e Casos de Uso	Linguagem Natural e Casos de Uso	Casos de Uso	Casos de Uso
Nº de requisitos em LN	23	35	-	-
Nº de casos de uso	9	13	10	12

Os grupos 1 e 2 escreveram a especificação de requisitos em inglês, conforme sugerido no treinamento, de forma a possibilitar a validação do documento com seus pares americanos. Em contraste, os grupos 3 e 4 escreveram a especificação de requisitos em português, o que impediu a leitura do documento pelos seus pares. Dessa forma, não houve validação formal dos requisitos.

O número de páginas da especificação não representou um padrão significativo que possa ser relacionado aos treinamentos. Analisando o número de palavras dos documentos percebemos que os grupos 1 e 2 escreveram especificações mais extensas, contendo no mínimo cinquenta por cento mais palavras que as geradas pelos grupos 3 e 4.

Quanto a forma de representação utilizada, os grupos 1 e 2 seguiram as instruções passadas no treinamento, especificando requisitos em linguagem natural e em

casos de uso. Por outro lado, os grupos 3 e 4 utilizaram apenas casos de uso. O número de casos de uso especificado pelos grupos foi próximo, com uma média de 11 casos de uso.

A Tabela 3 apresenta os resultados do questionário respondido pelas equipes que criaram as especificações com relação às qualidades do documento de especificação de requisitos, conforme [4]. As respostas de 1 a 5 representam o quanto o grupo concorda que a especificação produzida possui as características listadas (1 – Discorda totalmente, 5 - Concorda totalmente)

Tabela 3 – Respostas ao questionário em relação à qualidade do documento de especificação produzido

Característica	Grupo 1	Grupo 2	Grupo 3	Grupo 4	Média 1-2	Média 3-4
Correto	4	4	3	4	4	3,5
Não-ambiguo	3	4	3	3	3,5	3
Completo	4	4	2	2	4	2
Consistente	5	5	4	3	5	3,5
Priorizado	2	3	3	2	2,5	2,5
Verificável	3	5	4	4	4	4
Modificável	5	4	4	4	4,5	4
Rastreável	5	4	2	3	4,5	2,5
Total	31	33	25	25	32	25

De maneira geral, as respostas dos grupos 1 e 2 foram mais altas que as dos grupos 3 e 4. Isto indica que ao final do exercício os grupos treinados estavam mais confiantes da qualidade da especificação gerada.

Destacam-se entre as características avaliadas a completitude, consistência e rastreabilidade. Em relação a completitude da especificação produzida, os grupos treinados responderam com valor 4, enquanto que os grupos sem treinamento responderam com valor 2. Isso indica uma maior confiança que a especificação produzida está completa por parte dos grupos 1 e 2.

Considerando a consistência, os grupos 1 e 2 responderam com valor máximo (5), enquanto que os grupos 3 e 4 responderam, em média, 3,5. Quanto à rastreabilidade, os valores respondidos pelos grupos treinados foram, em média 4,5. Os grupos não treinados responderam em média 2,5 para este atributo. Isto se deve principalmente à importância dada no treinamento para manutenção da origem de cada informação especificada.

Analisando os documentos de especificação produzidos, percebeu-se que os grupos 1 e 2 produziram documentos com um maior número de informações adicionais, definido em detalhe questões como escopo e propósito do sistema. Por outro lado, os grupos 3 e 4 concentraram-se na especificação de casos de uso, utilizando poucas linhas para descrever o sistema e fornecer informações adicionais.

Em uma questão aberta definida como: “Cite as principais dificuldades enfrentadas pelos grupos durante o processo de especificação de requisitos de maneira distribuída.”, três dos quatro grupos brasileiros citaram questões relacionadas com a comunicação, como as dificuldades causadas pela interação assíncrona e ausência de comunicação com áudio e vídeo. Entre as respostas, foi destacada a demora causada pela comunicação assíncrona em conjunto com a diferença de fuso-horário. Muitas

vezes questões curtas demoravam a ser respondidas. Esta opinião foi compartilhada por um dos grupos americanos. Em contrapartida, os outros três grupos americanos consideraram o meio de comunicação utilizado suficiente, oferecendo uma boa plataforma para troca de mensagens.

Além disso, o grupo 3 definiu que “os maiores problemas deste processo ter se dado de forma remota foi a dificuldade da percepção das expectativas do usuário, não quanto as funcionalidades, mas sim a uma visão do problema que eles possuíam (não tendo o software), e quais as melhorias que a informatização deste processo traria a eles”, demonstrando dificuldade no entendimento do contexto do grupo americano. O mesmo grupo citou ainda como dificuldade a necessidade de feedback por parte dos usuários ao final do processo.

6. Análise Crítica

A documentação de requisitos realizada pelos grupos treinados foi mais abrangente e profunda que a gerada pelos grupos não treinados. Os grupos 1 e 2 buscaram um conjunto maior de informações, possibilitando entender melhor o contexto da aplicação a ser desenvolvida. Dessa forma, dificuldades comuns ao processo de engenharia de requisitos em ambientes distribuídos, gerados pela falta de informações sobre o local onde o software em desenvolvimento será utilizado, são reduzidas.

A validação do documento de especificação junto ao grupo de usuários forneceu aos grupos treinados uma maior confiança na especificação produzida. Os grupos não treinados ao optarem por escrever o documento em português não puderam validar seu conteúdo final com o par americano. Como o documento de especificação de requisitos é utilizado, muitas vezes, como base contratual, a escrita deste documento em uma linguagem não compreendida por uma das partes gera a necessidade de outro documento para contrato, ampliando problemas devido à redundância de informações e controle de modificações, por exemplo.

Um ponto importante de destaque é que os grupos não treinados preocuparam-se somente com a descrição da funcionalidade desejada em suas especificações, não utilizando requisitos não-funcionais. Este é um problema comum à uma grande parte das especificações e processos de engenharias de requisitos, pois requisitos não-funcionais são de mais difícil descoberta [6]. Entretanto, uma aplicação desenvolvida pode se apresentar totalmente inútil se não consideradas as características não-funcionais associadas. Ambos os grupos que receberam treinamento listaram requisitos não-funcionais e mantiveram sua associação aos casos de uso relacionados.

Em conclusão, de forma geral o conteúdo das especificações geradas pelos grupos treinados em colaboração e processo de engenharia de requisitos foi superior ao dos grupos sem treinamento. A especificação foi realizada de forma mais completa, com requisitos e casos de uso mais detalhados. Além disso, diversas informações adicionais foram colocadas na especificação, como descrição dos atores, glossário, contexto, entre outros, permitindo um melhor entendimento da aplicação a ser desenvolvida.

7. Considerações Finais

A engenharia de software vem realizando excelentes progressos nos últimos anos. Modelos de processo como o RUP e modelos de maturidade como o SW-CMM têm

sido adotados largamente e com sucesso no meio empresarial. Entretanto, novos desafios estão surgindo, exigindo diferentes abordagens para processos existentes. O desenvolvimento distribuído de software é um destes desafios.

A engenharia de requisitos, como um processo largamente dependente das interações entre os envolvidos é afetada diretamente pela distribuição das equipes. Nesse sentido, este estudo apresenta informações sobre uma pesquisa em engenharia de requisitos com equipes dispersas.

Os resultados aqui apresentados apontam para a importância da definição de um processo de engenharia de requisitos em ambientes de desenvolvimento distribuído de software, mesmo que simples, bem como a necessidade de treinamento das equipes em questões como colaboração e compartilhamento de contexto. Entretanto, os resultados desse estudo são limitados pelo baixo controle da pesquisa, e pela não-aleatoriedade da seleção dos participantes. Estas limitações poderiam ser sobrepostas com a utilização de um experimento puro, foco das próximas etapas do estudo.

Referências

- [1] E. Carmel (1999). “Global Software Teams – Collaborating Across Borders and Time Zones”. Prentice Hall. 269p.
- [2] D. Damian e D. Zowghi (2003). “An insight into the interplay between culture, conflict and distance in globally distributed requirements negotiations”. Proceedings of the 36th Hawaii International Conference on Systems Sciences (HICSS’03). IEEE.
- [3] J. Herbsleb e D. Moitra (2001). “Global Software Development”. IEEE Software.
- [4] Institute of Electrical and Electronics Engineers, Inc. “IEEE Recommended Practice for Software Requirements Specification”. Std 830-1998. IEEE Computer Society. New York, USA. 1998.
- [5] D. W. Karolak (1998). “Global Software Development – Managing Virtual Teams and Environments”. IEEE Computer Society. Los Alamitos, EUA. 159p. (3)
- [6] G. Kotonya e I. Sommerville. “Requirements Engineering: process and techniques”. John Wiley. 1998.
- [7] S. McConnell. “Rapid Development”. Microsoft Press. 1996. 647p. (13)
- [8] B. Nuseibeh e S. Easterbrook (2000). “Requirements Engineering: a Roadmap”. ACM - Future of Software Engineering. pp 37-45
- [9] R. Pressman (2001). “Software Engineering: a practitioner’s approach”. 5th ed. McGraw Hill. 860p
- [10] R. Prikladnicki (2003). “MuNDDoS - Um Modelo De Referência Para Desenvolvimento Distribuído De Software”. Dissertação de Mestrado, Programa de Pós-Graduação em Ciência da Computação. PUCRS.
- [11] The Standish Group International (1995). “Chaos Report”. http://www.standishgroup.com/sample_research/index.php. (Visualizado em 27 de julho de 2004).

- [12] R. Thayer e M. Dorfman (2000). “System and Software Requirements Engineering – Second Edition”. IEEE Computer Society Press Tutorial. 2000. 528p.
- [13] D. Zowghi. “Does global software development needs a different requirements engineering process?”. Proceedings of International Workshop on Global Software Development. 2002.