

Proposta de uma Abordagem para Especificação de Requisitos Baseada em Projeto Axiomático

Ana Maria Pereira¹, Paulo César Stadzisz¹, Jean Marcelo Simão¹
Universidade Tecnológica Federal do Paraná

¹ Programa de Pós-graduação em Engenharia Elétrica e Informática Industrial (CPGEI)
a995479@pos.utfpr.edu.br, stadzisz@utfpr.edu.br, jeansimao@utfpr.edu.br

Resumo

Desde as primeiras décadas do surgimento da informática são identificados problemas relativos à definição de requisitos de sistemas de software. Entender o que o cliente espera de um sistema de software é uma das tarefas mais difíceis enfrentadas pelos analistas de negócios e requisitos. Com o objetivo de contornar este problema, foram criadas diversas técnicas de especificação de requisitos, como especificações textuais, casos de uso, etc. Apesar destas técnicas oferecerem resultados promissores na descrição dos requisitos, elas falham na cobertura das necessidades do cliente. Sendo assim, o uso de um método focado no detalhamento e avaliação dos problemas resultaria em um software mais contudentes com as necessidades levantadas. Neste contexto, este trabalho apresenta uma proposta de abordagem para a especificação de requisitos baseada em Projeto Axiomático. Este método consiste de três passos: (1) o mapeamento de problemas, necessidades e requisitos em uma matriz de rastreabilidade, (2) avaliação da matriz segundo o Axioma da Independência e, por fim, (3) a decomposição dos problemas, necessidades e requisitos. Estes passos são repetidos ciclicamente até que se atinja um nível de detalhamento satisfatório. Os resultados dos experimentos atestaram que este método oferece auxílio na elaboração de um conjunto de requisitos de sistema mais consistente.

Palavras chave:

Requisitos, Necessidades do Cliente, Problema, Projeto Axiomático, Matriz de Rastreabilidade

Abstract

Since the early decades of the rise of computer science it has been identified problems on the definition of requirements for software systems. Understanding what the customer expects from a software system is one of the hardest tasks faced by business analysts and requirements. In order to overcome this problem, have been created several techniques for requirements specification, such as textual specifications, use cases, etc. Although these techniques offer promising results in the description of requirements, they fail to cover the needs of the client. Therefore, the use of a method focused in detailing and evaluating of the problems can result in software more coherent with the client needs. In this context, this paper presents a proposed approach to requirements specification based on axiomatic design. This method consists of three steps: (1) mapping the problems, needs and requirements through a traceability matrix, (2) evaluating the matrix according to the Axiom of Independence and, finally, (3) decomposing problems, needs and requirements. These steps are cyclically repeated until it reaches a satisfactory level of detail. The results of experiments testified that this method offers good results in preparing a more consistent set of system requirements.

Keywords:

Requirements, Client Needs, Problem, Axiomatic Design, Traceability Matrix

Introdução

Desde as primeiras décadas do surgimento da informática identificaram-se problemas relativos à elaboração e desenvolvimento de projetos de software. Logo no início dos anos 70 foi criado o termo “Crise de Software” para expressar as dificuldades enfrentadas pelos desenvolvedores na elaboração de seus projetos (DIJKSTRA, 1972). Naquela época, a engenharia de software era quase que inexistente e os problemas mais comuns costumavam ser: estouro no orçamento ou no prazo; baixa qualidade dos projetos; produtos que muitas vezes não atendiam os requisitos; e códigos difíceis de manter. Nos dias atuais, embora muitos avanços tenham sido realizados na área da engenharia de software, os problemas parecem continuar sendo os mesmos. Estatísticas apresentadas pelo Chaos Report 2009 (DOMINGUEZ, 2009) demonstram que os projetos de software têm percentuais de sucesso que giram em torno de apenas 30% como apresenta a Tabela 1. Além disso, estudos relacionados à Engenharia de Software apontam fortes evidências de que os mais sérios problemas de projetos estão diretamente relacionados à definição e entendimento de requisitos (LEFFINGWELL & WIDRIG, 2003).

Neste contexto, pode-se afirmar que um dos principais desafios de um projeto de software está em especificar e construir um produto que garanta a satisfação das necessidades do cliente com alta qualidade e dentro dos prazos e custos estimados. Tendo em vista esse desafio de atender as necessidades do cliente, procura-se garantir que todas elas sejam atendidas por meio da especificação e rastreamento de requisitos. Ao bem da verdade, a maioria das abordagens de especificação de requisitos foca na solução. Entretanto, ainda assim, uma grande parte dos projetos não tem o resultado esperado.

Distribuição de resultados de projetos de software	
Percentual	Resultados
32%	Sucesso (no prazo, dentro do orçamento e com escopo completo)
44%	Mudaram (atrasaram, estourou o orçamento, e/ou reduziram escopo)
24%	Falharam (cancelados ou nunca usados)

Tabela 1 Distribuição de resultados de projetos – Adaptado de (DOMINGUEZ, 2009)

Estudando-se os trabalhos de inúmeros autores é possível perceber que boa parte deles trata superficialmente os termos “problema” e “necessidade do cliente”, como é possível observar em (LEFFINGWELL & WIDRIG, 2003), (SUH & DO, 2000) e (BALMELLI, 2007) somente para citar alguns exemplos. Por outro lado, observa-se que o claro entendimento dos problemas e necessidades do cliente costuma ser um ponto chave para o desenvolvimento da especificação de requisitos de software. Por esta razão, um dos principais objetivos deste trabalho foi estudar mais atentamente os Problemas e as Necessidades dos clientes e seu mapeamento para requisitos de software.

Outro objetivo desta pesquisa foi estudar e demonstrar a aplicabilidade do Axioma da Independência proposto por Suh (SUH, 2001) à atividade de análise do problema e especificação de requisitos de software por meio de matrizes de rastreabilidade. A abordagem proposta pretende oferecer aos engenheiros de requisitos um método eficaz para diminuir as dificuldades em compreender e modelar os requisitos do sistema que deve ser construído. Assim sendo, o objetivo principal desse artigo é propor uma abordagem de especificação baseada no Axioma da Independência da Teoria de Projeto Axiomático.

O restante deste artigo está organizado da seguinte maneira. Na seção 2 apresentam-se alguns trabalhos anteriores, tanto relacionados ao tema de Projeto Axiomático como de especificação de requisitos. Na seção 3 trata-se da abordagem proposta de especificação de requisitos baseada em Projeto Axiomático. A seção 4 apresenta a metodologia utilizada nos experimentos realizados e traz um exemplo da aplicação desta abordagem. As discussões referentes aos resultados obtidos nos experimentos encontram-se na seção 5 e, finalmente, na seção 6 apresenta-se uma breve conclusão do trabalho.

Conclusões

Este trabalho apresenta uma abordagem de especificação de requisitos de sistemas de software baseada na Teoria de Projeto Axiomático, que pode ser aplicada em conjunto com um processo de desenvolvimento de software como o Processo Unificado, abordagem proposta provê um modelo para auxiliar os analistas de negócios e requisitos na definição dos requisitos adequados para o projeto. O entendimento do negócio do cliente, bem como de seus problemas, e a identificação dos requisitos corretos contribuem para a realização de um bom projeto.


Foi possível concluir, durante esta pesquisa, que o detalhamento dos problemas e/ou desejos que levam o cliente a tomar a decisão de adquirir um produto permite que os Engenheiros de Sistemas tenham uma visão ampliada do contexto em que a solução deverá se inserir e, desta maneira, possam propor soluções alternativas. É possível que estas soluções sejam até mais eficientes do que aquelas imaginadas pelo cliente através das características e necessidades apontadas por ele. Ou seja, buscar conhecer os problemas do cliente, sem se prender as necessidades ou características citadas por ele, ajuda a equipe de engenharia de requisitos a encontrar soluções inovadoras ou melhor fundamentadas para o problema.

Outra conclusão importante deste trabalho é que a estrutura formada pela decomposição entre os diversos domínios pode ser triangular. Isso quer dizer que o número de necessidades pode ser maior que o número de problemas assim como o número de requisitos pode ser maior que o número de necessidades. O importante é, que até determinado nível de decomposição, exista uma correspondência no número de itens de cada domínio. A partir do momento em que o número de itens de um domínio não puder mais ser detalhado deixa-se de utilizar a matriz entre estes domínios e passa-se a fazer o mapeamento apenas dos outros domínios que ainda podem ser detalhados.

A respeito do mapeamento de elementos de dois domínios em uma matriz, com objetivo de avaliá-la do ponto de vista do Axioma da Independência, concluiu-se que essa atividade pode auxiliar na identificação de inconsistências na interpretação dos elementos de cada domínio. Eventualmente, essa análise pode ajudar a identificar elementos que estão faltando ou sobrando e assim levar a melhorias na matriz e, conseqüentemente, no projeto. A utilização da matriz de projeto para o mapeamento entre os diversos domínios ajuda a manter a consistência e a rastreabilidade de problemas, necessidades, requisitos e elementos de arquitetura do sistema. Embora este trabalho tenha trazido algumas conclusões importantes, o tema de Especificação de Requisitos é bastante complexo e trabalhos futuros certamente poderão trazer novos avanços que irão agregar ainda mais conhecimento a respeito deste assunto.

Referências

- [1] BALMELLI, L. (2007). An Overview of the Systems Modeling Language for Products and Systems Development. *Journal of Object Technology* , 6, pp. 149-177.
- [2] BOOCH, G., RUMBAUGH, J., & JACOBSON, I. (2006). UML – Guia do Usuário (2 ed.).
- [3] CARVER, C. S., & SCHEIER, M. F. (2007). *Perspectives on Personality*. Boston: Allyn and Bacon.
- [4] DIJKSTRA, E. W. (Oct de 1972). The humble programmer. *ACM* , 859–866.
- [5] DOMINGUEZ, J. (2009). Acesso em 14 de Junho de 2010, disponível em ProjectSmart: <http://www.projectsmart.co.uk/pdf/the-curious-case-of-the-chaos-report-2009.pdf>
- [6] DORFMAN, M., & THAYER, R. H. (1990). *Standards, Guidelines, and Examples on System and Software Requirements Engineering*. IEEE Computer Society Press.
- [7] JACKSON, M. (1995). Problems & Requirements. *Proceedings of the IEEE Second International Symposium on Requirements Engineering* (pp. 2-8). ACM Press.

- 
- [8] KIM, S. J., SUH, N. P., & KIM, S. K. (1992). Design of software systems based on axiomatic design. Anais do CIRP General Assembly .
 - [9] LEE, T., & JEZIOREK, P. N. (2006). Understanding the Value of Eliminating An Off-Diagonal Term in a Design Matrix. Proceedings of ICAD 2006. Firenze.
 - [10] LEFFINGWELL, D., & WIDRIG, D. (2003). Managing Software Requirements. Boston: Addison Wesley.
 - [11] LI, Z. (2007). Progressing Problems from Requirements to Specifications in Problem Frames. London, United Kingdom: Department of Computing - The Open University.
 - [12] PIMENTEL, A. R. (2007). Uma abordagem para projeto de software orientado a objetos baseado na teoria de projeto axiomático. Curitiba-PR: UTFPR.
 - [13] SUH, N. P. (2001). Axiomatic Design: advances and applications. New York, EUA: Oxford University Press.
 - [14] SUH, N. P. (1990). The Principles of Design. New York: Oxford University Press.
 - [15] SUH, N. P., & DO, S. H. (2000). Axiomatic design of software systems. CIRP annals. Sidney, Australia.