

# Breve ensaio sobre a evolução histórica da engenharia de requisitos: motivações para seu surgimento\*

## *Short essay on the historical evolution of requirements engineering: motivations for its emergence*

André Luiz Câmara D'Oliveira<sup>1</sup>  
Leandro Vieira de Castro Teixeira<sup>2</sup>  
Luiz Roberto Vieira<sup>3</sup>  
Márcio Rodrigo dos Santos<sup>4</sup>

### Resumo

Este ensaio tem como objetivo expor algumas motivações que levaram ao surgimento da engenharia de requisitos, bem como sua contribuição no processo de desenvolvimento, objetivando a produção de *software* com qualidade. Atender às necessidades do cliente de forma satisfatória e com qualidade é um dos grandes desafios da indústria do *software*. O presente ensaio, por meio de pesquisa bibliográfica, apresenta um pequeno recorte da evolução histórica da engenharia de requisitos e as melhorias contínuas desse novo conceito. É apresentado um cenário de como o desenvolvimento de *software* era no passado e como é hoje.

**Palavras-chave:** Processo de desenvolvimento de *software*. Engenharia de *software*. Engenharia de requisitos. Qualidade de *software*.

### Abstract

This essay aims to explain some motivations that led to the emergence of requirements engineering, as well as their contribution in the development process, the objective of producing quality software. Meet customer needs in a satisfactory manner and quality is a major challenge in the software industry. This essay, through bibliographic research has a small cutout of the historical evolution of requirements engineering and continuous improvements of this new concept. It presented a scenario of how the software development was in the past and as it is today.

**Keywords:** software development process. Software engineering. Engineering requirements. Software quality.

---

\* Artigo recebido em 11/03/2012  
Aprovado em 10/09/2012

<sup>1</sup> Tecnólogo em Processamento de Dados pelo Centro de Ensino Unificado de Brasília – Uni-Ceub. Analista de Suporte da Companhia de Saneamento Ambiental do DF - CAESB, desempenhando atividades relativas à modelagem de dados e administração em banco de dados relacional (SQL Server).

<sup>2</sup> Possui 10 anos de experiência em análise e desenvolvimento de sistemas, com sólidos conhecimentos em levantamento de requisitos, modelagem de banco de dados relacional e elaboração artefatos de software. Já participou de diversos projetos das áreas de Telecomunicação, Educação e Saúde baseados no RUP, bem como em projetos com metodologias ágeis.

<sup>3</sup> Aluno da Pós-Graduação em Engenharia de Requisitos do Instituto CEUB de Pesquisa e Desenvolvimento – ICPD.

<sup>4</sup> Aluno da Pós-Graduação em Engenharia de Requisitos do Instituto CEUB de Pesquisa e Desenvolvimento – ICPD.

## 1 Introdução

O objeto de estudo do presente ensaio é a evolução histórica da engenharia de requisitos por ser um assunto relevante no cenário atual de desenvolvimento de *software*.

Primeiramente, é importante saber que o desenvolvimento de *software* era realizado de forma puramente artesanal. Os *softwares* desenvolvidos continham muitos erros e, tais erros geravam enormes retrabalhos tornando o processo de desenvolvimento dispendioso.

Em estudo realizado pelo The Standish Group Report (1994) quanto ao *software* produzido, observou-se que:

- 16% dos projetos foram bem-sucedidos, isto é, concluídos no prazo e orçamento previstos, com todos os recursos e funções especificados originalmente;
- 31% dos projetos foram cancelados antes de serem concluídos ou nunca foram implementados;
- 53% dos projetos foram concluídos e operacionalizados com menos recursos e funções do que foi especificado originalmente, além de extrapolarem o custo e o prazo;
- Dos projetos entregues com menos funcionalidades, o percentual de funções originalmente especificadas e entregues foi de 61%;
- Não foram divulgados dados quanto ao uso de metodologias formais.

Embasados nos fatos, os olhos do desenvolvimento de *software* se voltaram para a criação de um processo que pudesse sanar o que acima foi observado, tornando a entrega de produtos de *software* com baixa qualidade uma das principais preocupações da indústria de *software*.

Leffingwell (1997) ressalta que 40% a 60% de todos os problemas encontrados em um projeto são causados por falhas no processo de requisitos - ausência ou não utilização de um processo de definição de requisitos adequado.

A consequência da falta de um processo de requisitos eficaz tem sido a produção de *softwares* que não refletem as necessidades reais dos clientes. Com isso, faz-se

valer que o principal parâmetro para se medir a qualidade de um *software* é o grau em que ele atende às necessidades do cliente. Em termos gerais, pode-se dizer que o desenvolvimento de *software* depende do bom entendimento das necessidades dos clientes - entendimento do negócio - e do bom desenvolvimento dos requisitos.

Segundo Alves e Finkelstein (2002), a análise final da qualidade de um *software* é determinada pelo atendimento aos requisitos dos *stakeholders*.<sup>5</sup>

Este ensaio tem como objetivo expor quais as motivações que levaram ao surgimento da Engenharia de Requisitos, atividade base para as demais tarefas associadas ao desenvolvimento de *software*. É realizada uma revisão bibliográfica sobre a trajetória do desenvolvimento de *software* e a contribuição da Engenharia de Requisitos na obtenção de um processo de desenvolvimento que prima qualidade aos *softwares* produzidos pela indústria do *software*

## 2 Desenvolvimento

### 2.1 O começo

Desde o começo da era dos sistemas computadorizados, o *hardware* era considerado o componente mais importante na evolução tecnológica. Ele passava por contínuas mudanças e evoluções, sempre visando a um menor custo de processamento e armazenamento de dados. Nessa época, o desenvolvimento de *software* era feito sem administração, com pouco ou nenhum método formal e customizados a ambientes físicos proprietários.

A partir da década de 80, os avanços da microeletrônica mudaram gradativamente esse panorama. Hoje, consegue-se um grande poder de processamento a um custo muito baixo e o *software* se tornou um importante componente que nos permite aproveitar todo esse potencial. “O *software* é um elemento de sistema lógico, e não de um sistema físico. Assim, ele possui características que são consideravelmente diferentes daquelas do *hardware*” (PRESSMAN, 2006, p. 4).

Essas diferenças fizeram com que o *software* passasse de ferramenta especializada de resolução de problemas para um produto com indústria própria, focada em

<sup>5</sup> Entende-se como as partes envolvidas no projeto.

desenvolver soluções a milhares de usuários atendendo a diversas necessidades.

Muitas linguagens de programação foram criadas e centenas de milhares de código produzidos dentro das organizações. Com o passar do tempo, esses sistemas, imprescindíveis ao negócio das empresas, normalmente construídos de maneira artesanal e com pouquíssima documentação, representavam um alto custo de manutenção dentro das organizações, pois precisavam ser evoluídos constantemente para atender às necessidades reais dos seus clientes.

Até mesmo os *softwares* utilizando tecnologias de ponta tinham seus processos de desenvolvimento confusos, sempre extrapolando prazos e custos e não atendendo totalmente aos seus propósitos, ocasionando uma constante insatisfação dos *stakeholders*. No entanto, algo deveria ser feito para que os *softwares* fossem desenvolvidos com baixo custo, dentro do prazo estipulado e com alta qualidade, pois o *software* se tornou indispensável tanto dentro das organizações como nos computadores pessoais. Acreditava-se que a utilização plena da Engenharia de *Software* era o caminho para se chegar a esses objetivos.

## 2.2 A engenharia de *software*

Com a ascensão do *software*, ele passou a perceber um custo muito alto às organizações. A indústria do *software* carecia de um processo eficaz para alcançar o objetivo de produzir *software* de qualidade; com isso, entra em cena a Engenharia de *Software*.

A Engenharia de *Software* propõe métodos sistemáticos com o uso adequado de ferramentas e técnicas que levam em consideração o problema a ser resolvido, as necessidades dos clientes e os recursos disponíveis.

Para Naur (1969, p. 231), “Engenharia de *Software* é a criação e a utilização de sólidos princípios de engenharia a fim de obter *software* de maneira econômica, que sejam confiáveis e que trabalhem eficientemente em máquinas reais”.

Nos últimos anos, essa área e suas disciplinas têm amadurecido bastante, por meio da concepção de novos métodos e de técnicas que possibilitem o desenvolvimento de *softwares* mais confiáveis, de melhor qualidade, com maior produtividade e custo reduzido.

Para o Institute of Electrical and Electronic Engineers -IEEE (1990), a Engenharia de *Software* é definida

pela a aplicação de uma abordagem sistemática, disciplinada e quantificável para o desenvolvimento, operação e manutenção de *software*, isto é, a aplicação de engenharia em *software*.

Outra definição para a Engenharia de *Software* é apresentada por Sommerville (2007, p. 5) como uma disciplina de engenharia relacionada com todos os aspectos de produção de *software*, desde os estágios iniciais de especificação do sistema até a sua manutenção, ou seja, mesmo depois que ele entrar em operação. Isso revela a importância da Engenharia de *Software* dentro de todo o projeto e sua forte ligação com o resultado final do produto.

A busca da qualidade do *software* é o que motiva os desenvolvedores a usarem os processos descritos pela Engenharia de *Software*. Todos os processos descritos nas diferentes técnicas e formas de desenvolvimentos buscam o mesmo objetivo, a qualidade. Qualidade esta que vai desde a produção até a entrega do *software*.

Mesmo com a utilização dos processos definidos na Engenharia de *Software*, a qualidade do produto final ainda representava um grande problema. A comunicação entre os *stakeholders* e a área técnica de Tecnologia da Informação (TI) nas empresas era superficial e responsável pelas sucessivas insatisfações geradas por resultados que não atendiam às necessidades de negócio. As consequências eram óbvias: projetos abandonados antes do seu fim, prazos dificilmente cumpridos e os custos de construção e/ou manutenção dos *softwares* muito acima do previsto. Era realmente preciso definir um processo formal que ajudasse os engenheiros de *software* a compreenderem melhor os problemas que os *stakeholders* necessitavam resolver e o impacto que o *software* a ser desenvolvido iria exercer sobre o negócio do cliente.

## 2.3 O advento da engenharia de requisitos

Em 1993, durante o IEEE, a Engenharia de Requisitos ficou estabelecida como uma disciplina independente. Trata-se de uma subárea da Engenharia de *Software* que estuda o processo de produção e gerência dos requisitos que o *software* deverá atender.

“A Engenharia de Requisitos é a disciplina que procura sistematizar o processo de definição de requisitos” (LEITE, 1994). Segundo Leite (1994), essa atividade é de extrema importância já que a complexidade dos sistemas exige uma maior atenção ao correto entendimento

do problema antes do acordo de uma solução final. Essa é uma atividade que trata de conhecimentos não apenas técnicos, mas também gerenciais, organizacionais, econômicos e sociais (CASTRO, 1995), e está intimamente associada à qualidade do *software* (LEE, 1998).

Requisito é assim tratado por Sommerville:

O termo requisito não é utilizado pela indústria de *software* de modo consistente. Em alguns casos, o requisito é visto como uma declaração abstrata, de alto nível, de uma função que o sistema deve fornecer ou de uma restrição do sistema. No outro extremo, ele é uma definição detalhada, matematicamente formal, de uma função do sistema (SOMMERVILLE, 2003, p. 82).

Ainda conforme Sommerville (2003) é competência da Engenharia de Requisitos descobrir, analisar, documentar e validar os requisitos do sistema, onde temos cinco atividades de alto nível: o estudo da viabilidade do sistema, a obtenção e análise de requisitos, a especificação e documentação de requisitos, a validação dos requisitos e o gerenciamento de requisitos. Para algumas organizações, o processo de desenvolvimento de *software* evoluiu e o levantamento de requisitos deixou de ser uma fase e passa ao conceito de construção, iniciando com o estudo de viabilidade do projeto de *software*.

O grande passo a ser dado na utilização do processo de Engenharia de Requisitos era um entendimento profundo das necessidades dos *stakeholders* por parte dos engenheiros de *software*. E isso é imprescindível para que o produto final possa ter a qualidade necessária. Porém, segundo Pressman (2006), a Engenharia de Requisitos não deve ser encarada como a solução para os desafios já citados, mas trata-se de uma sólida abordagem sistemática para nos ajudar a enfrentá-los. Mesmo assim, pode-se afirmar que a documentação gerada pela Engenharia de Requisitos é uma base sólida para as fases posteriores do ciclo de desenvolvimento de *software*, pois, sem ela, a probabilidade de não atendimento às necessidades do cliente é muito alta.

Atualmente, a Engenharia de Requisitos está presente nos dois primeiros níveis dos modelos de maturidade de *software* CMMI e MPS.BR, o que reforça a ideia de que ela continua representando um dos principais alicerces para a produção de *softwares* de qualidade no Brasil e no mundo.

### 3 Considerações finais

O que se percebe com este ensaio, é que, há muito, a indústria do *software* vem sendo assolada pela falta ou pelo mau uso de um processo eficaz de *software* que traga qualidade aos *softwares* desenvolvidos.

O advento da Engenharia de Requisitos originou-se pelas dificuldades encontradas na trajetória do desenvolvimento de *software*. Sua maior motivação foi, e continua sendo, a busca pela qualidade dos *softwares* desenvolvidos, ou seja, fazer com que eles reflitam as reais necessidades dos clientes.

Vale salientar que os processos evoluíram bastante e que, aplicando os preceitos da Engenharia de Requisitos na dimensão em que se encontram hoje, certamente os *softwares* produzidos sofrerão uma melhora significativa.

### Referências

ALVES, C.; FINKELSTEIN, A. *Eighth International Workshop on Requirements Engineering*: Foundation for Software Quality. Essen: [s.n.], 2002.

CASTRO J. F. B. Introdução a engenharia de requisitos. In: JORNADA DE ATUALIZAÇÃO EM INFORMÁTICA DO CONGRESSO DA SOCIEDADE BRASILEIRA DE COMPUTAÇÃO, 14., 1995, Canela. *Anais...* Porto Alegre: Instituto de Informática da UFRGS, 1995.

INSTITUTE OF ELECTRICAL AND ELECTRONIC ENGINEERS IEEE Standard 610-1990. IEEE Standard Glossary of Software Engineering Terminology, IEEE Press. New York: IEEE, 2007.

LEE, W. J.; CHA, S. D.; KNOWN, Y. R. Integration and analysis of use cases using modular petri nets in requirements engineering. *IEEE Transactions on Software Engineering*, New York, v. 24, n. 12, p. 1115-1130, Dec. 1998.

LEFFINGWELL, D. Calculating the return on investment from more effective requirements management. *American Programmer*, Arlington, v. 10, n. 4, p.13-16, 1997.

LEITE, J. C. S. P. *Engenharia de requisitos: notas de aula*, Parte I, 1994. Disponível em: <<http://livrodeengenhariaderequisitos.googlepages.com/ERNOTASDEAULA.pdf>>. Acesso em: 15 nov. 2011.

NAUR, Peter; RANDELL, Brian Randell (Ed.). *Software engineering*: report on a conference sponsored by the Nato Science Committee Garmisch, Germany, 7th to 11th October 1968. Brussels, Belgium: Scientific Affairs Division NATO, 1969.

PRESSMAN, R. S. *Engenharia de software*. 6. ed. São Paulo: McGraw Hill, 2006.

SOMMERVILLE, I. *Engenharia de software*. 6. ed. São Paulo: Pearson Addison - Wesley, 2003.

SOMMERVILLE, I. *Engenharia de software*. 8. ed. São Paulo: Pearson Addison - Wesley, 2007.

THE STANDISH GROUP REPORT. *Chaos*: 1994. Disponível em: <<http://www.projectsmart.co.uk/docs/chaos-report.pdf>> Acesso em: 15 nov. 2011.

**Para publicar na revista Universitas Gestão e TI,  
entre no endereço eletrônico  
[www.publicacoesacademicas.uniceub.br](http://www.publicacoesacademicas.uniceub.br).**

**Observe as normas de publicação, facilitando e agilizando o trabalho de edição.**