

Modelagem e testes de software na perspectiva de profissionais da indústria: Um estudo qualitativo

Software modeling and testing from the perspective of industry professionals: A qualitative study

Adriano Martins de Pontes
Departamento de Ciências Exatas
Universidade Federal da Paraíba
(UFPB) – Campus IV
58280-990 – Rio Tinto – PB – Brasil
adriano.pontes@dcx.ufpb.br

Pablo José L. de Lima
Departamento de Ciências Exatas
Universidade Federal da Paraíba
(UFPB) – Campus IV
58280-990 – Rio Tinto – PB – Brasil
pablo.lima@dcx.ufpb.br

José Adson O. G. da Cunha
Departamento de Ciências Exatas
Universidade Federal da Paraíba
(UFPB) – Campus IV
58280-990 – Rio Tinto – PB – Brasil
adson@dcx.ufpb.br

RESUMO

A engenharia de software possui várias fases para o desenvolvimento de um software com o objetivo de garantir um produto de qualidade. Duas fases importantes correspondem à análise e projeto, na qual são construídos modelos, e aos testes. Apesar da importância de tais etapas, algumas equipes não criam modelos para apoio na codificação, enquanto outras não realizam os testes de forma adequada. Este artigo tem como objetivo, a partir de um estudo exploratório qualitativo, identificar fatores que influenciam na análise e projeto e nos testes baseado na experiência de profissionais da indústria.

Palavras-chave

Modelagem de Software, Teste de Software, Estudo Qualitativo.

ABSTRACT

Software engineering has several steps for software development with the aim of guaranteeing a quality product. Two steps correspond to analysis and design, in which models are built, and testing. Despite the importance of such steps, some teams do not create models for coding support, while others do not perform the tests properly. This article aims to identify factors that influence the analysis and design and testing based on the experience of industry professionals through a qualitative exploratory study.

CCS Concepts

• **Software and its engineering** → **Software creation and management** → **Software development process management.**

Keywords

Software Modeling, Software Testing, Qualitative Study.

1. INTRODUÇÃO

A prática da engenharia de software consiste em uma série de princípios, conceitos, métodos e ferramentas que devem ser considerados no planejamento e desenvolvimento de um software

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

SBSI 2018, June 4th–8th, 2018, Caxias do Sul, Rio Grande do Sul, Brazil.

Copyright SBC 2018.

[1]. Nesse contexto, duas etapas importantes merecem atenção: análise e projeto e testes.

Na fase de análise e projeto de software, modelos são elaborados com o objetivo de representar a estrutura e o comportamento do sistema [1]. Ao mesmo tempo em que modelos são importantes para facilitar a comunicação e manutenibilidade do software, tem-se que considerar a agilidade requerida para o desenvolvimento, influenciado por fatores técnicos, culturais e humanos [2]. Apesar da importância da análise e projeto, nem sempre a mesma é planejada e executada [3], sendo a codificação realizada a partir da especificação dos requisitos.

A área de testes, por sua vez, concentra as tarefas que garantem que o desenvolvedor implementa corretamente uma função específica e que esta pode ser rastreada segundo os requisitos do cliente [1]. Da mesma forma, apesar das técnicas, métodos, ferramentas e metodologias utilizados para testes de software, existem vários desafios que precisam ser enfrentados, como a educação de testadores e o entendimento dos custos de testes [4].

Entender os fatores individuais e contextuais que influenciam na realização de mais ou menos modelos, assim como mais ou menos testes, impactando assim na qualidade do produto final, é algo importante para o planejamento dos projetos pelos profissionais de desenvolvimento de software. Nesse sentido, este trabalho tem como objetivo oferecer uma visão sobre o tema na perspectiva de profissionais da indústria de software a partir de um estudo qualitativo.

As demais seções deste trabalho estão estruturadas como segue: A Seção 2 apresenta a fundamentação teórica. A Seção 3 descreve o método utilizado. A Seção 4 descreve os resultados do estudo. Na Seção 5 são apresentadas as limitações da pesquisa. Por fim, a Seção 6 apresenta as conclusões e trabalhos futuros.

2. FUNDAMENTAÇÃO TEÓRICA

2.1 Análise e Projeto

No contexto do desenvolvimento de software, um desenvolvedor pode usar diagramas para representar os vários aspectos de um projeto de software. Em um *survey* realizado com 113 profissionais em desenvolvimento de software, [3] confirmaram a UML como a notação de modelagem dominante e que as abordagens centradas no modelo são mais fáceis, porém, não muito populares, já que a maioria dos participantes trabalham em ambientes centrados em código. Em [5], a partir de uma pesquisa

com 80 arquitetos sobre o uso da linguagem UML, concluíram que a aderência ao padrão da notação é baixo.

De acordo com um *survey* realizado com 3785 desenvolvedores, [6] concluíram que o uso de modelos diminui com o aumento da experiência e aumenta com maior nível de qualificação. Além disso, os modelos são usados principalmente como um mecanismo de comunicação e colaboração em grupo, raramente são atualizados após a criação inicial e geralmente são desenhados em um quadro branco ou em papel, sem preocupação com notação. Em um *survey* com 155 profissionais de software da indústria italiana, [7] classificaram a modelagem como uma técnica relevante que proporciona uma melhoria na documentação, facilidade de manutenção e aumento da qualidade do produto. No entanto, exige um grande esforço para ser produzido, sendo, frequentemente, produzidos modelos não úteis devido à falta de ferramentas de suporte e falta de competência dos profissionais.

2.2 Teste de Software

Uma estratégia de teste de software deve acomodar testes de baixo e alto nível [1]. Considerando a importância do teste para o desenvolvimento de software confiável e de alta qualidade, algumas pesquisas se concentraram no estudo da influência dos aspectos humanos relacionados aos testes. Através de um *survey* com 80 testadores, [8] investigaram como os fatores relacionados ao trabalho podem influenciar a motivação dos testadores de software. Os autores observaram que os testadores de software são fortemente motivados pela variedade do trabalho, tarefas que exigem criatividade, reconhecimento por seu trabalho e atividades que lhes permitem adquirir novos conhecimentos. No entanto, em geral, o impacto social desta atividade tem pouca influência sobre a sua motivação.

Um outro estudo realizado por [9] concluiu algo semelhante: além da variedade do trabalho e reconhecimento pelo trabalho, os problemas centrais giram em torno da falta de desafios e da necessidade de um bom gerenciamento dos testes.

Em [10] os autores investigaram os motivos pelos quais os graduados em computação não estão interessados na carreira de testador. Os autores observaram que os alunos esperam que seus trabalhos forneçam alguns elementos que, seguindo teorias recentes, são antecedentes da motivação no trabalho, como aquisição de conhecimento, criatividade e variedade de trabalho. Os autores concluíram que os alunos não acreditam que a carreira como testador irá prover tais elementos.

3. MÉTODO DE PESQUISA

De acordo com [11], a Engenharia de Software Baseada em Evidências deve prover a base científica para a realização de mudanças nos processos relacionados ao desenvolvimento de software. Esta pesquisa visa obter conhecimento sobre como os profissionais de desenvolvimento de software interpretam suas experiências no que diz respeito às fases de análise e projeto e testes em projetos de software.

Uma vez que estamos interessados em uma visão mais ampla dos fenômenos, as questões de pesquisa não foram restritivas:

QP1: *Quais fatores influenciam a análise e projeto na perspectiva de profissionais de desenvolvimento de software?*

QP2: *Quais fatores influenciam a realização de testes na perspectiva de profissionais de desenvolvimento de software?*

3.1 Coleta de Dados

Tendo em vista a diversidade e riqueza dos dados coletados, a amostragem teórica nesta pesquisa qualitativa básica objetivou incluir profissionais de diferentes organizações e com diferentes níveis de experiência [13]. A unidade de análise foi o profissional em atividade durante a pesquisa. Ao todo foram realizadas 8 entrevistas. A Tabela 1 apresenta as informações dos participantes.

Tabela 1. Informações dos Participantes

ID	Anos de experiência	Função predominante	Tipo de Org.	Tamanho da Org.	Tempo da entrevista (min)
PR01	4	Desenvolvedor	Privada	Pequena	15
PR02	2	Desenvolvedor	Privada	Pequena	12
PR03	11	Arquiteto	Pública	Grande	32
PR04	1	Desenvolvedor	Privada	Pequena	16
PR05	15	Especificador	Pública	Grande	26
PR06	2	Desenvolvedor	Privada	Grande	10
PR07	2	Testador	Privada	Pequena	15
PR08	8	Desenvolvedor	Privada	Grande	18

Os dados foram coletados através de entrevistas semiestruturadas com profissionais da indústria de software. O guia de entrevista foi composto de perguntas abertas e incluiu diferentes tipos de perguntas com o objetivo de explorar experiências e comportamentos, opiniões e valores, sentimentos, conhecimento e os antecedentes dos participantes (ver Figura 1). Foram criados dois roteiros, com perguntas semelhantes, sendo um referente à análise e projeto e o outro referente a testes. Os guias de entrevista foram refinados através de entrevistas-piloto.

<p>P1. [Antecedente] Fale um pouco de você: sua formação, idade, trajetória profissional.</p> <p>Sondagem: Quais funções você exerceu em projetos de software?</p> <p>Sondagem: Como foi sua experiência com [testes] [análise e projeto] durante sua formação?</p> <p>...</p> <p>P4. [Opinião] O que a sua organização oferece ou faz para estimular [a realização de testes] [a análise e projeto de software]?</p> <p>P5. [Opinião] O que a sua organização faz (e/ou que não deveria fazer) que dificulta [a realização de testes] [a análise e projeto de software]?</p> <p>P6. [Experiência] Fale sobre o projeto em que trabalha atualmente.</p> <p>P7. [Opinião] Quais tipos de [testes] [modelos] são mais realizados em seu projeto? Por que?</p> <p>...</p> <p>P13. [Opinião] Como você descreveria um colega que desempenha bem as atividades relacionadas [à teste de software] [à análise e projeto de software]?</p> <p>P14. [Opinião] Como você descreveria um colega que desempenha mau as atividades relacionadas [à teste de software] [à análise e projeto de software]?</p> <p>P15. [Sentimento] Como você avalia o seu desempenho [como testador] [na análise e projeto de software]?</p> <p>P16. [Opinião] Qual a importância da [realização de testes] [análise e projeto] em projetos de software?</p> <p>...</p>

Figura 1. Trecho do roteiro de entrevista.

As perguntas foram apresentadas em formato de funil, começando com questões gerais e avançando para mais específicas [12].

Todas as questões positivas tiveram um negativo correspondente (ver P4 e P5). Para atender aos requisitos éticos deste tipo de pesquisa, cada participante foi informado sobre a pesquisa e seus direitos a fim de garantir a confidencialidade dos dados fornecidos, o anonimato do participante e o direito de se retirar da pesquisa a qualquer momento [13]. A autorização foi concedida através de uma pergunta específica sobre seu acordo para participar desta pesquisa. Algumas entrevistas foram realizadas nas próprias instalações das organizações e outras através de teleconferência ao longo do mês de outubro de 2017. Todo o áudio das sessões de entrevista foi gravado com o consentimento dos participantes para posterior transcrição.

3.2 Análise dos Dados

Para a análise dos dados coletados foi empregado o método de análise de conteúdo [14], composta por três fases: (i) pré-análise; (ii) exploração do material; e (iii) tratamento dos resultados, inferência e interpretação.

A análise do conteúdo das entrevistas foi realizada por meio da Análise Categrorial. Em tal abordagem, os textos são agrupados de forma análoga através de categorias, sendo uma boa alternativa quando se busca o estudo e aprofundamento de crenças, valores, opiniões e atitudes. Este trabalho foi realizado em conformidade com o método de Análise de Conteúdo Temático-Categrorial elaborado por [15], o qual sugere a aplicação das seguintes etapas:

- **Preparação das informações:** Todas as entrevistas foram transcritas para serem posteriormente analisadas.
- **Transformação do conteúdo em unidades de registro (UR):** Escolha do tipo de unidade de registro. Foi definido como UR os recortes do texto avaliados como relevantes para compreensão do objeto de estudo.
- **Classificação de unidades de registro em temas:** Cada unidade de registro é associada a um ou mais temas, com o propósito de associar URs dentro de um mesmo contexto de significação.
- **Classificação de temas em categorias:** Refere-se ao agrupamento dos temas e suas respectivas unidades de registro em categorias, conforme ilustrado na Figura 2.
- **Descrição:** Corresponde ao resumo numérico das etapas anteriores como objetivo de apresentar a soma das unidades de registro por tema em cada entrevista.
- **Interpretação:** Para cada tema pertencente a uma categoria específica, o investigador deve fazer suas ponderações, a fim de elucidar a compreensão e a importância do tema.

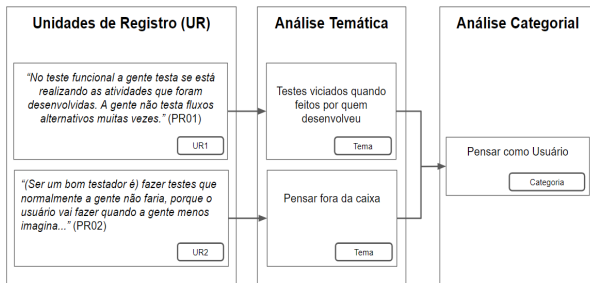


Figura 2. Exemplo de análise.

4. RESULTADOS

Com base na análise efetuada das transcrições das entrevistas realizadas com profissionais de desenvolvimento de software, foram identificados vários temas que compuseram um conjunto de

categorias, agrupados em Fatores Contextuais e Fatores Individuais, conforme ilustrado na Figura 3.

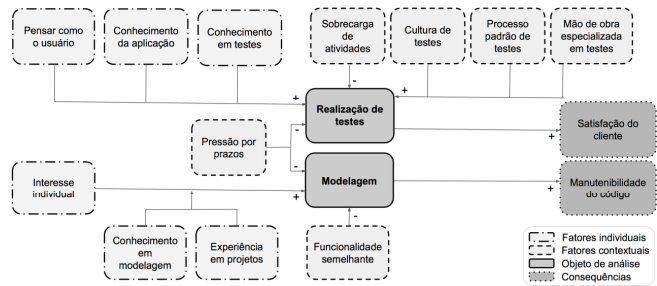


Figura 3. Resultado da análise dos dados.

A categoria **Pressão por prazos** esteve presente na maior parte das entrevistas (6 URs no âmbito de modelagem e 5 URs no âmbito de testes), seguido por **Interesse do engenheiro de software**, no âmbito de modelagem, e **Pensar como usuário**, no âmbito de testes, presentes em 5 URs. Por outro lado, as categorias **Funcionalidades semelhantes** e **Cultura de testes** estiveram presentes em apenas 1 UR. A frequência com que a categoria é repetida não necessariamente deve ser levada em consideração para conclusão que uma categoria é de maior importância para a pesquisa, pois para a pesquisa qualitativa os resultados vão além de números. Devido a restrições de espaço, apenas as evidências de algumas categorias foram descritas.

A **pressão por prazos** foi considerada como principal fator para a ausência de modelagem e realização de testes, sendo o esforço maior despendido para codificação do que foi especificado, de modo a entregar o mais rápido possível, sem contar necessariamente com a elaboração de um modelo prévio ou realização de testes rigorosos para controle da qualidade.

"(...) devido à pressão ou alguma coisa desse tipo muitas vezes o teste acaba sendo despriorizado e a gente vai colher os frutos podres mais pra frente, tem que correr atrás." (PR03).

"A única coisa que dificulta são os prazos, mas assim, a modelagem tem que ser feita, mas como os prazos são corridos, isso termina dificultando um pouco." (PR02)

A presença de um **Processo padrão de testes**, por sua vez, facilita a realização de testes de forma sistematizada, com atividades e papéis bem definidos.

"Se pelo menos outro desenvolvedor testasse o projeto feito por outro desenvolvedor. Acaba que fica meio confuso essa parte de testes nesse quesito." (PR01).

A falta de definição das responsabilidades na equipe, ficando a maior parte das tarefas centralizadas em poucos membros, gera uma **Sobrecarga de atividades**, o que dificulta a realização de testes.

"É mais na questão de (estar) muito atarefado (...). Chego e já tem muita coisa pra fazer." (PR04).

Pensar como usuário foi indicado como algo importante a ser considerado no momento da proposição de roteiros de teste, facilitando assim a elicitación de casos que fogem do "caminho feliz".

"Eu tento pensar como o usuário iria querer usar aquilo ali, e aí eu faço os testes funcionais." (PR01).

Além dos fatores citados anteriormente, o **Conhecimento da Aplicação** e **Conhecimento em Testes**, como fatores individuais, e a **Cultura de testes** e **Mão de obra especializada em testes**, como fatores contextuais, influenciam de forma positiva a realização de testes.

Quanto à modelagem, a existência de **Funcionalidade semelhante** dentro de um mesmo projeto faz com que a equipe não elabore novos diagramas, baseando-se assim no que já foi desenvolvido.

“Temos uma arquitetura bem distinguida, e normalmente a gente consegue encaixar um desenvolvimento dentro dessa arquitetura e sanar nossos problemas.” (PR01).

O **Interesse individual** foi pontuado como fator determinante na decisão sobre a elaboração ou não de modelos antes de se iniciar a codificação.

“Eu acho essa parte de modelo um pouco burocrática, na verdade o que eu gosto mesmo é de desenvolver.” (PR04)

Além disso, o **Conhecimento em modelagem** obtido durante a graduação influencia na importância que é dada à modelagem durante o desenvolvimento de um software real. Junto com a **Experiência individual**, tais fatores moderam a influência do interesse individual em relação à modelagem.

“A gente trabalhou muito essa questão de análise, de elaborar documento, elaborar modelo e propor o caso de uso, diagrama de atividade, de sequência. Eu acho que tudo que eu vi nessas disciplinas (durante a graduação) eu consigo usar no mercado de trabalho.” (PR04)

Como consequência, a realização de testes foi reconhecida pelos profissionais como essencial para garantir a **Satisfação do usuário**. Por outro lado, a modelagem foi considerada importante por proporcionar uma rastreabilidade do desenvolvimento, ajudando assim na **Manutenibilidade do software**.

5. LIMITAÇÕES DA PESQUISA

Diferente do paradigma pós-positivista que busca generalizar os resultados [13], a abordagem utilizada neste trabalho, através de uma pesquisa qualitativa, tem o propósito de adquirir os conhecimentos oriundos da perspectiva dos profissionais de desenvolvimento de software entrevistados.

A saturação dos dados a partir das entrevistas realizadas não foi obtida, sendo necessária a realização de novas entrevistas para aprofundamento das categorias identificadas, bem como o uso de outros métodos de coleta, como observação e análise de documentos, para triangulação dos dados. Ainda, faz-se necessária a realização de *member checking* para confirmação dos achados.

A inexperiência dos alunos responsáveis pela condução das entrevistas e posterior análise dos dados influenciou no resultado da pesquisa. A partir da experiência obtida nesta primeira rodada completa, espera-se que, uma vez mais maduros, as categorias mapeadas sejam refinadas.

6. CONCLUSÃO E TRABALHO FUTURO

Este estudo contribui para um melhor entendimento dos fatores que influenciam positiva e negativamente na abordagem prática de modelagem e testes de software. Uma lista de fatores foi

elencada através de uma pesquisa qualitativa baseada na experiência dos entrevistados nesta pesquisa.

Como trabalho futuro, sugere-se voltar a campo para validar a presente análise, colhendo mais informações ao replicar o protocolo com outros profissionais de outras organizações bem como a utilização de outros métodos de coleta de dados.

7. REFERÊNCIAS

- [1] Pressman, R, e Maxim, B. 2016. Engenharia de Software-8ª Edição. McGraw Hill Brasil.
- [2] Demarco, T., e Lister, T. 2013. Peopleware: productive projects and teams. Addison-Wesley.
- [3] Forward, A., Badreddin, O., e Lethbridge, T. C. 2010. Perceptions of software modeling: a survey of software practitioners. In: *5th workshop from code centric to model centric: evaluating the effectiveness of MDD*.
- [4] Bertolino, A. Software testing research: Achievements, challenges, dreams. 2007. In: 2007 Future of Software Engineering. *IEEE Computer Society*, p. 85-103.
- [5] Lange, C. F. J., Chaudron, M. R. V., e Muskens, J. 2006. In practice: UML software architecture and design description. *IEEE software*, v. 23, n. 2, p. 40-46.
- [6] Gorschek, T., e Tempero, E., e Angelis, L. 2014. On the use of software design models in software development practice: An empirical investigation. *Journal of Systems and Software*, v. 95, p. 176-193.
- [7] Torchiano, M. et al. 2013. Relevance, benefits, and problems of software modelling and model driven techniques—A survey in the Italian industry. *Journal of Systems and Software*, v. 86, n. 8, p. 2110-2126.
- [8] Santos, R. E. de S., Magalhães, C. V. C. de, Correia-Neto, J. da S., Silva, F. Q. B. da, Capretz, L. F., and Souza, R. 2017. Would You Like to Motivate Software Testers? Ask Them How. *Proceedings of the ACM/IEEE International Symp on Emp Soft Engineering and Measurement*, p. 95-104.
- [9] Deak, A., Stålhane, T., e Sindre, G. 2016. Challenges and strategies for motivating software testing personnel. *Information and software Technology*, 73, 1-15.
- [10] Waychal, P. K., e Capretz, L. F. 2016. Why a Testing Career Is Not the First Choice of Engineers. arXiv preprint arXiv:1612.00734.
- [11] Dybå, T., Kitchenham, B. A., e Jørgensen, M. 2005. Evidence-Based Software Engineering for Practitioners, *IEEE Software*, Vol. 22, No. 1, pp 58-65.
- [12] Runeson, P. e Host, M. 2008. Guidelines for conducting and reporting case study research in software engineering. *Empirical Software Engineering*, Vol. 14, No. 2, p. 131-164.
- [13] Merriam, S. B. 2009. Qualitative Research: a Guide to Design and Implementation, San Francisco, CA: Jossey-Bass.
- [14] Bardin, L. 2013. Content analysis. Editions Lisbon, 3th ed.
- [15] Oliveira, D. C. 2008. Análise de conteúdo temático-categorial: uma proposta de sistematização. *Rev. enferm. UERJ*.