

Fatores que influenciam a realização de testes em projetos de software: Um estudo qualitativo

Adriano Martins de Pontes¹

Curso de Bacharelado em Sistemas de Informação – Universidade Federal da Paraíba
(UFPB) – Campus de IV
58280-990 – Rio Tinto– PB– Brasil

adriano.pontes@dcx.ufpb.br

Abstract. *The development of large software systems is a complex and error prone process. Such errors can occur at any stage of development and must be identified and removed as quickly as possible to prevent their propagation and reduce the verification costs. One of the stages of software development is testing, which serves to ensure the quality of the software. However, tests are not always performed by software engineers due to several factors. This article presents an exploratory qualitative study with the objective of identifying some of the factors that influence the performance of tests in software development projects.*

Resumo. *O desenvolvimento de grandes sistemas de software é um processo complexo e propenso a erros. Tais erros podem ocorrer em qualquer fase de desenvolvimento e devem ser identificados e removidos o mais rápido possível para impedir sua propagação e reduzir os custos de verificação. Uma das etapas do desenvolvimento de software é a de testes, que serve para garantir a qualidade do software. No entanto, nem sempre os testes são realizados pelos engenheiros de software devido a vários fatores. Este artigo apresenta um estudo exploratório de natureza qualitativa com o objetivo de identificar alguns dos fatores que influenciam a realização de testes em projetos de desenvolvimento de software.*

1. Introdução

A engenharia de software abrange processos, métodos e um leque de ferramentas que possibilitam aos profissionais desenvolverem software de qualidade, o que é garantido principalmente pela realização de testes [Pressman e Maxim 2016]. A área de testes de software corresponde a um elemento de um tema mais amplo conhecido como Verificação e Validação (V&V). Enquanto a verificação concentra as tarefas que garantem que o software implementa corretamente uma função específica, a validação abrange as tarefas que asseguram que o software foi criado e pode ser rastreado segundo os requisitos do cliente [Pressman e Maxim 2016].

Testes tem como objetivo garantir qualidade e eliminar erros. Segundo a Tricentis (2016) no ano de 2016, com base em dados públicos de empresas de desenvolvimento, os custos para reparo de erros de software foram de 1.1 trilhão de dólares e cerca de 315

¹ Trabalho de Conclusão de Curso do discente Adriano Martins de Pontes sob a orientação do docente José Adson Oliveira Guedes da Cunha, submetido ao Curso de Bacharelado em Sistemas de Informação da Universidade Federal da Paraíba, Campus IV, como parte dos requisitos necessários para obtenção do grau de Bacharel em Sistemas de Informação.

anos em carga horária investida pelos desenvolvedores, afetando aproximadamente 4.4 bilhões de usuários.

Uma estratégia de teste de software deve acomodar testes de baixo nível, necessários para verificar se um pequeno segmento de código fonte foi implementado corretamente, bem como testes de alto nível, que validam as funções principais do sistema de acordo com os requisitos do cliente. Nesse contexto, os desenvolvedores devem se responsabilizar pelo teste das unidades individuais do programa, garantindo que cada uma execute a função ou apresente o comportamento para o qual foi projetada, como também pelo teste de integração, para garantir o funcionamento da arquitetura completa do software. Os testes de sistema, que incluem a validação dos requisitos funcionais, em geral são realizados por um grupo de teste independente [Pressman e Maxim 2016].

Estudos anteriores estimaram que o teste pode consumir ao menos 50% dos custos de desenvolvimento [Beizer 1990]. Uma pesquisa detalhada realizada nos Estados Unidos [NIST 2002] quantificou como alto o impacto econômico gerado por uma infraestrutura inadequada de teste de software. Apesar das técnicas, métodos, ferramentas e metodologias utilizados para testes de software, existem vários desafios que precisam ser enfrentados [Bertolino 2007]. Entender os fatores individuais e contextuais que influenciam na realização de 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 teve como objetivo oferecer uma visão sobre o tema na perspectiva de profissionais da indústria de software da cidade de João Pessoa, na Paraíba, sendo elencados os fatores individuais e contextuais que influenciam a realização de testes. Para tanto, foram realizadas entrevistas semiestruturadas com profissionais de quatro organizações.

As demais seções deste trabalho estão estruturadas como segue: A Seção 2 apresenta os trabalhos relacionados. 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

De acordo com Gilb et al. (1993), uma estratégia de teste de software terá sucesso somente quando os testadores de software: (1) especificarem os requisitos do produto de uma maneira quantificável muito antes de começar o teste, (2) definirem explicitamente os objetivos do teste, (3) entenderem os usuários do software e desenvolverem um perfil para cada categoria de usuário, (4) desenvolverem um plano de teste que enfatize o “teste do ciclo rápido”, e (5) criarem software “robusto” que seja projetado para testar-se a si próprio.

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, Santos et al. (2017) 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 Deak et al. (2016),

concluíram 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.

Waychal e Capretz (2016) 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.

Apesar de se reconhecer a importância dos testes durante o desenvolvimento de software, nem sempre os mesmos são realizados de forma adequada. Compreender de forma empírica o que influencia esse fenômeno é importante não só para a indústria mas também para a academia de modo que possa ajustar a forma como as disciplinas relacionadas a testes são lecionadas.

3. Método

De acordo com Dybå et al (2005), 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 a testes de software e por que certas combinações de fatores individuais e contextuais levam à realização de mais ou menos testes.

Uma vez que estamos interessados em uma visão mais ampla do fenômeno, a questão de pesquisa não foi restritiva. Portanto, a principal questão da pesquisa é:

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

3.1 Coleta dos 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 e formação [Merriam 2009]. 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. Perfil dos participantes

ID	Formação	Anos de experiência no mercado	Função predominante	Tipo de Organização	Tamanho da Organização ²	Tempo da entrevista
PR01	Bacharel em Sistemas de Informação	4	Desenvolvedor	Privada	Pequeno porte	15 minutos

² <http://www.sebrae-sc.com.br/leis/default.asp?vcdtexto=4154>.

PR02	Tecnólogo em Sistemas para Internet	2	Desenvolvedor	Privada	Pequeno porte	12 minutos
PR03	Bacharel em Ciência da Computação	11	Arquiteto de software	Pública	Grande porte	32 minutos
PR04	Bacharel em Sistemas de Informação	1	Desenvolvedor	Privada	Pequeno porte	16 minutos
PR05	Bacharel em Ciência da Computação	15	Especificador	Pública	Grande porte	26 minutos
PR06	Bacharel em Sistemas para Internet	2	Desenvolvedor	Privada	Grande porte	10 minutos
PR07	Bacharel em Sistemas para Internet	2	Testador	Privada	Pequeno porte	15 minutos
PR08	Tecnólogo em Sistemas para Internet	8	Desenvolvedor	Privada	Grande porte	18 minutos

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 (Apêndice A). O guia de entrevista foi refinado através de uma entrevista-piloto.

As perguntas foram apresentadas em formato de funil, começando com questões gerais e avançando para mais específicas [Runeson e Host 2008]. As questões gerais incentivam reflexões importantes trazendo mais detalhes ao responder as questões específicas, possibilitando assim entender como experiências passadas podem influenciar preferências, sentimentos e comportamentos. Elas também servem para criar uma relação estreita e confiável entre o pesquisador e o entrevistado. Todas as questões positivas tiveram um negativo correspondente (ver P4 e P5), conforme protocolo apresentado no Apêndice A.

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 [Merriam 2009]. 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 [Bardin 2013], 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 foram realizada por meio da Análise Categorical. 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-Categorial elaborado por Oliveira (2008), o qual sugere a aplicação das seguintes etapas:

- **Preparação das informações:** Todas as entrevistas foram transcritas para serem posteriormente analisadas através da técnica de Análise de Conteúdo.
- **Transformação do conteúdo em unidades de registro (UR):** Nesta etapa determina-se a escolha do tipo de unidade de registro (palavra, frase ou parágrafo) que foi adotada pelo pesquisador no decorrer da análise. Neste trabalho, 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. Cada tema, portanto, é composto por uma ou mais URs. Os temas emergem durante o processo de análise, ou seja, não são pré-estabelecidos.
- **Classificação de temas em categorias:** Refere-se ao agrupamento dos temas e suas respectivas unidades de registro em categorias. Cada categoria está associada a um conjunto de temas, e cada tema à apenas uma categoria, conforme ilustrado na Figura 1.
- **Descrição:** Esta etapa corresponde ao resumo numérico das etapas anteriores com o objetivo de apresentar a soma das unidades de registro encontradas no texto de cada entrevista, por tema.
- **Interpretação:** Para cada tema pertencente a uma categoria específica, o investigador deve fazer suas ponderações, a fim de elucidar a compreensão e a importância da existência do tema.

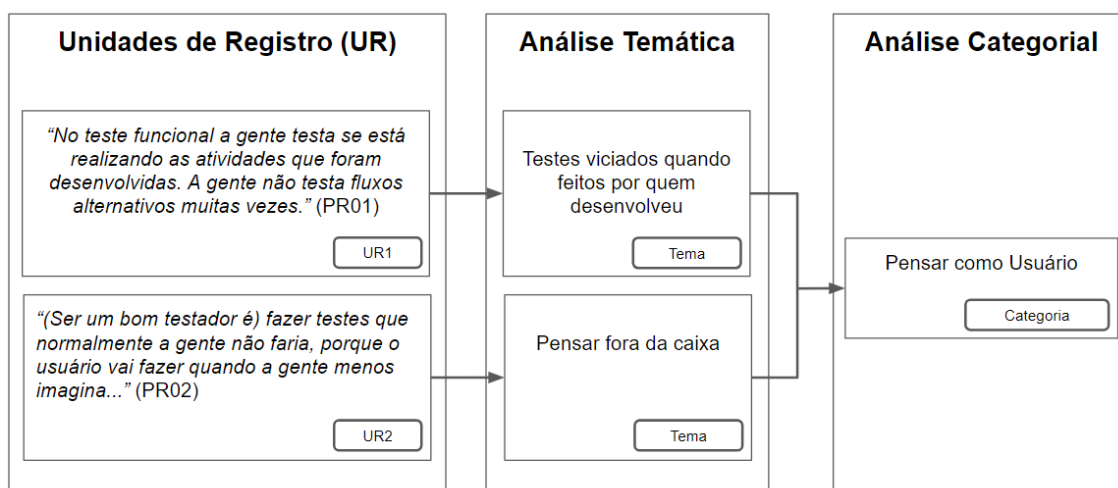


Figura 1. Exemplo de unidades de registro, temas e categoria

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, compondo o ilustrado na Figura 2

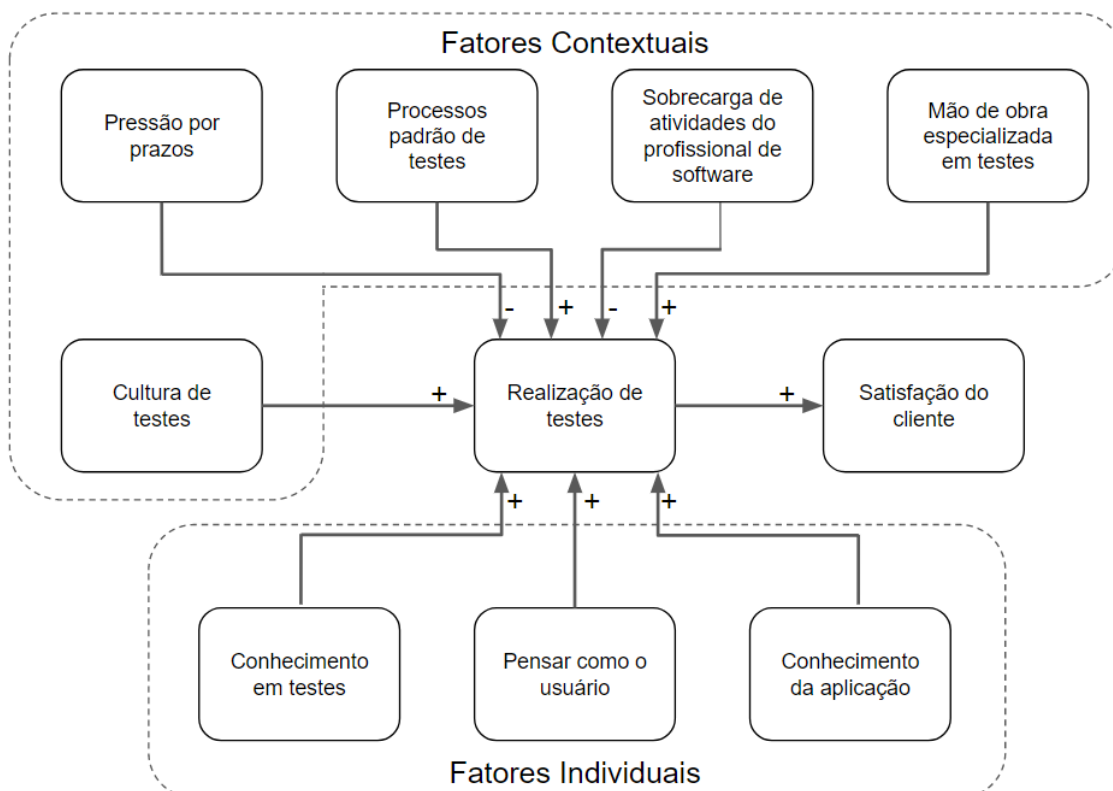


Figura 2. Modelo resultante da análise

As categorias mais mencionadas foram “*Pressão por prazos*”, sendo mencionados por todos os entrevistados, “*Pensar como usuário*”, sendo mencionado por 5 dos 8 entrevistados, e “*Ausência de processo padrão*” sendo mencionada por 3 dos 8 entrevistados. As categorias menos mencionadas foram: “*Cultura de Testes*”, sendo mencionados por 1 dos 8 entrevistados, e “*Conhecimento da Aplicação*”, sendo mencionada por 1 dos 8 entrevistados. 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 e evidências.

As seções 4.1 e 4.2 apresentam, respectivamente, as categorias referentes aos fatores contextuais e individuais. A Seção 4.3 apresenta a consequência identificada da realização dos testes.

4.1 Fatores contextuais

Os fatores contextuais são fatores provenientes do ambiente da organização no qual os profissionais trabalham que, de acordo com o relato dos entrevistados, afetam suas percepções e abordagens quanto a testes de software nos projetos. As categorias referentes aos fatores contextuais são: *“Pressão por prazos”*, *“Processos padrão de testes”*, *“Sobrecarga de atividades do engenheiro de software”*, *“Mão de obra especializada em testes”* e *“Cultura em testes”*.

Com base nos relatos dos entrevistados foi observado que 3 das 4 organizações aborda de forma básica ou insatisfatória os prazos, onde os testes são diretamente impactados, sendo despriorizados ou completamente descartados, caracterizando a categoria *“Pressão por prazos”*.

- *“Às vezes a questão da pressa é por conta do usuário e a gente acaba sofrendo uma pressão por parte da diretoria para entregar o software e acaba que alguns softwares a gente faz e não consegue testar rápido porque na verdade o teste deveria ser feito durante um tempo.” (PR01).*
- *“... talvez um dos problemas que talvez eu não tenha te falado é a da despriorização do teste, devido a pressão, devido a prazo, a pressão política ou alguma coisa desse tipo então muitas vezes o teste acaba sendo despriorizado e a gente vai colher os frutos podres mais pra frente, tem que correr atrás...” (PR03).*

A categoria *“Processos padrão de testes”* foi definida por entrevistados de 3 das 4 organizações, que parecem não ter processos para testes de software bem definidos ou pelo menos satisfatórios, impactando na abordagem dos testes pelos desenvolvedores, que acabam executando testes simples demais ou sem qualquer formalização de procedimentos ou métodos de modo a garantir uma melhor qualidade do produto a ser entregue.

- *“... a gente acaba testando só os nossos projetos. Se pelo menos outro desenvolvedor testasse o projeto feito por outro desenvolvedor. Acaba que fica meio confuso essa parte de testes nesse quesito.” (PR01).*
- *“Bom, minha parte é testar o que faço e evidenciar ao máximo possível que não quebrou outras coisas, além do que eu mexi. Caso isso ocorra tenho de mostrar a eles, criar uma pasta com as evidências e apresentar.” (PR06).*

Alguns dos entrevistados relataram uma sobrecarga de atividades em um curto período de tempo, sendo assim mapeada a categoria *“Sobrecarga de atividades do engenheiro de software”*.

- *“É mais na questão de... Muito atarefado, é mais uma questão de tempo. Chego e já tem muita coisa pra fazer...” (PR04).*
- *“... a gente teve que levar para Requisitos, as vezes desenvolver Banco de Dados e até o Desenvolvimento de Sistema, apresentação para cliente... A gente, pelo porte da empresa, a gente participa de todas as etapas do projeto.” (PR01).*

Também foram relatados problemas quanto à mão de obra destinada a testes, onde dois entrevistados relataram que os estagiários que recebiam a responsabilidade da realização dos testes, sendo que eles recebiam gradativamente atividades fora de testes e acabam não podendo mais se dedicar a tais atividades, onde a organização teria de aguardar a contratação de um novo estagiário para poder ter alguém alocado nas atividades de testes, além de que o estagiário durante a graduação tem pouco ou nenhum conhecimento em testes, como relatado na categoria “*Conhecimento em testes*” nos fatores individuais, e os temas relacionados foram agrupados na categoria “*Mão de obra especializada em testes*”.

- “... a única pessoa destinada para isso (testar), ela nunca pode fazer isso porque ela sempre está com outras coisas pra se fazer.” (PR02).
- “Sim... Inicialmente havia um estagiário focado em testes, acompanhado pelos membros da equipe e acabou que esse estagiário foi efetivado (recebendo outras atividades que não testes)...” (PR07).

Um entrevistado mencionou que a cultura de testes da organização influenciava negativamente na aplicação de testes nos projetos com os quais eles trabalham ou trabalhavam, compondo a categoria “*Cultura de testes*”, onde os testes eram abordados de forma muito genérica ou simplesmente propagando uma má prática.

- “Acho que a cultura que existe hoje lá deixa um pouco as coisas amarradas... as vezes acaba que enquanto está desenvolvendo a gente não consegue testar o software de outra pessoa. Acaba complicando um pouco esse processo.” (PR01).

4.2 Fatores individuais

Os fatores individuais são fatores advindos da experiência, interesse e conhecimento que impactam em como os entrevistados vêem e abordam testes nos projetos em que trabalham e/ou trabalharam. As categorias identificadas foram: “*Conhecimento em testes*”, “*Pensar como o usuário*”, “*Conhecimento da aplicação*”, e “*Cultura de testes*”.

A categoria “*Conhecimento em testes*” foi formada observando que poucos foram os que tiveram experiência prática com testes durante a graduação e a maioria julga como sendo “básico” o conhecimento que obtiveram sobre testes durante este período. Todos os entrevistados vêem a abordagem dos testes durante a graduação como insuficiente ou básica demais.

- “Durante a graduação a gente teve pouco enfoque nessa parte de testes, na verdade a gente só viu... Acho que não teve nenhuma disciplina voltada exatamente pra essa execução de teste, pra saber como fazer estes testes não.” (PR01).
- “Durante a graduação nenhuma, foi só abordado teoricamente, mas não foi colocado nada em prática em testes mesmo. Na verdade só com roteiros de testes e casos de teste mas só na teoria mesmo.” (PR04).

Alguns relataram não ter paciência para testes ou dificuldade para pensar em fluxos alternativos de uso ou “pensar como o usuário”, de modo a tentar prever todas as possibilidades de execução do usuário para tratar possíveis problemas nestes fluxos alternativos, caracterizando a categoria “*Pensar como o usuário*”.

- “... eu tento pensar como o usuário iria querer usar aquilo ali, e aí eu faço os testes funcionais... Só sei que não tenho paciência pra testar.” (PR01).
- “... Não consigo pensar em muitos cenários de testes, as vezes eu vejo pessoas do meu lado que conseguem com mais facilidade gerar cenários diferentes, entendeu?!...” (PR08).

Um dos entrevistados relatou que o conhecimento da aplicação impacta na qualidade dos testes, já que no contexto organizacional ele julgou como “genéricos” os testes executados pela equipe especializada, onde um profissional que conhece a aplicação, como o próprio desenvolvedor ou equipe, trazendo o que ele sabe da aplicação para os testes, complementando-os e com isso compondo a categoria “*Conhecimento da aplicação*”.

- “... Eu acho que esses testes (performance, funcionais e mais avançados) poderiam ser mais disseminados dentro das equipes ao invés de ficar em uma caixinha isolada, porque aí a gente traria aos testes o conhecimento que temos também do sistema que a gente desenvolve... ” (PR05).
- ... eu acho que ainda falta disseminar essa cultura dentro dos projetos pra... Sei lá, de repente pra a gente fazer alguns testes mais avançados, já que eles (equipe de testes) fazem esses testes básicos, já que essa “caixinha” tem que existir e a gente partir para alguns tipos de testes mais avançados, não sei... ” (PR05).

Dois dos entrevistados de organizações distintas mencionaram que a cultura de testes do profissional influenciava negativamente na aplicação de testes nos projetos com os quais eles trabalham ou trabalhavam, compondo a categoria “*Cultura de testes*”, onde os testes eram abordados de forma muito genérica ou simplesmente propagando uma má prática.

- “Acho que a cultura que existe hoje lá deixa um pouco as coisas amarradas, e também a quantidade de colaboradores, hoje em um projeto, cada pessoa tem que pegar um projeto e ir até o fim, as vezes acaba que enquanto está desenvolvendo a gente não consegue testar o software de outra pessoa...” (PR01).
- “Exatamente. São testes do tipo ‘O sistema está respondendo em menos de 5 segundos? Tá ou não tá.’. Enfim, coisa bem genérica...” (PR05).

4.3 Satisfação do Cliente

Como mostrado na Figura 2, todas as categorias resultam em um impacto na satisfação do cliente, sendo ele também o usuário ou não. Com base no analisado, todos os entrevistados demonstram que as categorias levantadas nesta pesquisa impactam na satisfação do cliente, mas um dos entrevistados mencionou diretamente esta consequência por excesso de retorno do cliente por problemas tanto relacionados a análise de projeto quanto a testes.

- *“... Porque a gente testa o caminho que é pra ser feito só que acaba que quando a gente chega no cliente ele faz outros caminhos que a gente não havia testado... E a gente tem retrabalho com relação a isso porque o cliente quer usar de uma forma que não foi programada pra funcionar, aí se a gente tivesse testes com relação a isso mitigaria esses contatos do cliente após a entrega.” - (PR01)*

Por outro lado, outro entrevistado evidenciou vantagens de uma melhor abordagem de testes em sua organização. Seu cliente valida todas as funções do sistema por meio de roteiros executados automaticamente, além da disponibilização de um roteiro em linguagem natural gerado automaticamente graças ao uso de um framework próprio deles.

- *“Tem os projetos aqui na empresa que a gente conseguiu, pra você ter uma ideia, fazer uma homologação com o cliente 100% através de roteiros de testes funcionais automatizados... É uma coisa que você não consegue mais viver sem ela... É inerente ao processo, que se você não realizar aquela atividade ali você não consegue ter uma garantia boa do que você está entregando.” (PR03)*

O trecho acima evidencia muito bem o que esta pesquisa tem como objetivo, que é ajudar a identificar os fatores que impactam na realização de software, de modo a investir nos fatores positivos e buscar eliminar se possível os negativos para atingir resultados mais próximos dos que o descrito acima.

5 Limitações da pesquisa

Diferente do paradigma pós-positivista que busca generalizar os resultados [CRESWELL 2013], 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, não necessariamente com o objetivo de generalizá-los.

Um ponto a ser considerado seria a quantidade de engenheiros entrevistados. Foram entrevistados 8 engenheiros de 4 organizações, sendo 2 de cada organização. Apesar da riqueza de informação fornecida pelos entrevistados nesta pesquisa, para um maior embasamento um número maior de entrevistados seria requerido para atingirmos a saturação.

Outro ponto quanto a este estudo foi o tempo relativamente curto para a condução desta pesquisa, dada a complexidade e demanda maior de tempo para análise, para busca por mais profissionais para serem entrevistados e para reavaliar o que foi analisado junto aos entrevistados, trazendo novos questionamentos para refinar mais o conteúdo e obter novos dados.

6 Conclusão e trabalhos futuros

Este estudo contribui para um melhor entendimento dos fatores que influenciam positiva e negativamente na abordagem prática de testes de software. Uma lista de fatores foi elencada através de uma pesquisa qualitativa baseada na experiência prática dos entrevistados nesta pesquisa, sendo os fatores contextuais: *“Pressão por prazos”*, *“Processos padrão de testes”*, *“Sobrecarga de atividades do engenheiro de software”*,

“Mão de obra especializada em testes”, e “Cultura de testes” e os fatores individuais: “Conhecimento em testes”, “Pensar como o usuário”, e “Conhecimento da aplicação”. O objetivo desta pesquisa foi levantar tais fatores e evidenciar que os mesmos impactam de forma positiva ou negativa na abordagem e visão do uso de testes por engenheiros de software.

Como trabalho futuro, sugere-se voltar a campo para validar a presente análise, colhendo mais informações ao replicar o método de entrevista com mais profissionais de mais organizações para uma maior diversificação de dados para obter mais ou confirmar melhor fatores que influenciam, positivamente ou negativamente, a realização de testes em projetos de software.

Referências

- Bardin, L. Content analysis. Editions Lisbon, 3th edition, 2013.
- Beizer, B. Software Testing Techniques (2nd ed.). Van Nostrand Reinhold Co., New York, NY, USA, 1990
- Bertolino, A. Software testing research: Achievements, challenges, dreams. In: 2007 Future of Software Engineering. IEEE Computer Society, 2007. p. 85-103.
- Creswell, J. W. Research design: Qualitative, quantitative, and mixed methods approaches. Sage publications, 2013.
- Deak, A., Stålhane, T., e Sindre, G. (2016). Challenges and strategies for motivating software testing personnel. Information and software Technology, 73, 1-15.
- Dybå, T., Kitchenham, B. A., e Jørgensen, M. Evidence-Based Software Engineering for Practitioners, IEEE SOFTWARE, Vol. 22, No. 1, pp 58-65, 2005.
- Gilb, T., Graham, D., e Finzi, S. Software inspection. Addison-Wesley Longman Publishing Co., Inc., 1993.
- Merriam, S. B. Qualitative Research: a Guide to Design and Implementation, San Francisco, CA: Jossey-Bass, 2009.
- Nist. The economic impacts of inadequate infrastructure for software testing, May 2002. <https://www.nist.gov/document/report02-3pdf>.
- Oliveira, D. C. Análise de conteúdo temático-categorial: uma proposta de sistematização. Rev. enferm. UERJ, 2008.
- Pressman, R, e Maxim, B. Engenharia de Software-8ª Edição. McGraw Hill Brasil, 2016.
- Runeson, P. e Host, M. Guidelines for conducting and reporting case study research in software engineering. Empirical Software Engineering, Vol. 14, No. 2, p. 131-164, 2008.
- 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. Would You Like to Motivate Software Testers? Ask Them How Ronnie Edson de Souza Santos, Cleyton Vanut Cordeiro de Magalhães, Proceedings of the ACM/IEEE International Symposium on Empirical Software Engineering and Measurement, 2017, p. 95-104.
- Tricentis, Software Fail Watch: 2016 in Review, 2016.

Waychal, P. K., e Capretz, L. F. (2016). Why a Testing Career Is Not the First Choice of Engineers. arXiv preprint arXiv:1612.00734.

Apêndice A - Protocolo da Entrevista com Engenheiros de Software

Apresentação • Auto-apresentação • Agradecimento ao participante • Permissão para gravar o áudio da entrevista Objetivo da Entrevista O objetivo desta pesquisa é entender os aspectos relacionados à realização de testes em projetos de software. Introdução Todas as informações fornecidas nesta entrevista serão tratadas como confidenciais. Apenas a equipe de pesquisa terá acesso às informações fornecidas. Nenhum dado que possa ser usado para lhe identificar, como seu nome e email, será exposto em qualquer fase da pesquisa. O conteúdo das entrevistas será utilizado apenas para fins da pesquisa acadêmica, não tendo assim nenhuma influência na avaliação do funcionário no desempenho das suas atividades na organização. A entrevista será gravada para posterior documentação. Ao final da pesquisa, os dados serão publicados em eventos de natureza acadêmica, mas os nomes das pessoas envolvidas serão omitidos nas respectivas publicações. Sua participação nesta pesquisa é voluntária e você pode decidir não participar ou se retirar da pesquisa a qualquer momento. Caso você decida não participar, não receberá nenhuma sanção ou penalidade. Você concorda em participar desta pesquisa? Identificação do entrevistado As informações a seguir serão utilizadas caso a equipe de pesquisa precise entrar em contato com você no futuro para esclarecimentos sobre a entrevista. Por favor, diga seu nome e e-mail.	
P1. [Antecedente] Fale um pouco de você: sua formação, idade, trajetória profissional. <i>Sondagem:</i> Quais funções você exerceu em projetos de software? <i>Sondagem:</i> Como foi sua experiência com testes durante sua formação? P2. [Antecedente] Há quanto tempo você trabalha nesta empresa? P3. [Experiência] A organização utiliza algum processo de desenvolvimento de software? <i>Sondagem:</i> Descreva-o em linhas gerais. P4. [Opinião] O que a sua organização oferece ou faz para estimular a realização de testes? P5. [Opinião] O que a sua organização faz (e/ou que não deveria fazer) que dificulta a realização de testes? P6. [Experiência] Fale sobre o projeto em que trabalha atualmente. <i>Sondagem:</i> Quais são as principais dificuldades que vocês enfrentam no projeto em que trabalha? P7. [Opinião] Quais tipos de testes são mais realizados em seu projeto? Por que? <i>Sondagem:</i> E por quais motivos não usa testes automatizados? P8. [Experiência] Qual influência você exerce nos testes realizados no projeto? P9. [Experiência] Que problemas você vivenciou pela falta de testes?	P10. [Experiência] Em que situações a equipe deixa de realizar algum teste? <i>Sondagem:</i> E qual tipo de teste é descartado primeiro? P11. [Sentimento] Que tipo de teste você se sente mais à vontade para realizar? Por que? P12. [Sentimento] Que tipo de teste você se sente menos à vontade para realizar? Por que? P13. [Opinião] Como você descreveria um colega que desempenha bem as atividades relacionadas à teste de software? P14. [Opinião] Como você descreveria um colega que desempenha mau as atividades relacionadas à teste de software? P15. [Sentimento] Como você avalia o seu desempenho como testador? P16. [Opinião] Qual a importância da realização de testes em projetos de software? P17. Você gostaria de adicionar alguma informação ou observação que não foi perguntada, mas que você considera importante para a realização de testes em projetos de software?