

UNIVERSIDADE FEEVALE

MARISTELA OLIVEIRA DE PAULA

AUTOMATIZAÇÃO DE TESTE EM AMBIENTE ÁGIL: A
PROPOSTA DE UM MODELO DE PROCESSO

(Título Provisório)

Anteprojeto de Trabalho de Conclusão

Novo Hamburgo
2019

MARISTELA OLIVEIRA DE PAULA

AUTOMATIZAÇÃO DE TESTE EM AMBIENTE ÁGIL: A
PROPOSTA DE UM MODELO DE PROCESSO

(Título Provisório)

Anteprojeto de Trabalho de Conclusão de
Curso, apresentado como requisito parcial
à obtenção do grau de Bacharel em
Sistemas de Informação pela
Universidade Feevale

Orientador: Prof. Dr. Adriana Neves dos Reis

Novo Hamburgo
2019

RESUMO

Nos últimos anos, cada vez mais a indústria de desenvolvimento de software tem empregado em seus modelos de trabalho Metodologias Ágeis, por necessitarem suprir a demanda de um mercado que anseia pelo seu produto entregue em curto espaço de tempo sem perder a qualidade. Essa postura de mercado impacta diretamente em constantes mudanças de requisitos que, para organizações que desenvolvem o produto, geram grandes desafios, tais como análise de impacto e testes em tempo hábil, em conjunto à preocupação com uma entrega de qualidade. Com isso, são mais motivados a buscar melhorias no processo de desenvolvimento, obtendo como uma saída possível a automatização dos testes do software que consomem tempo e impactam diretamente na qualidade do produto e entrega. Porém, embora se tenha essa conscientização disseminada sobre os benefícios do uso da Automação de Testes de Software (ATS), muitas empresas falham na implantação ou na continuidade deste processo. A ausência de um modelo para ATS dentro de contexto ágil contribui para esse insucesso. Assim, fazendo uso da metodologia *Design Science Research* (DSR), este trabalho tem por objetivo propor um modelo para o processo de ATS em equipes que empregam Metodologias Ágeis. Para tal, utilizando das melhores práticas no que tange à qualidade e ao processo de desenvolvimento de software, almeja-se uma proposta com base tanto na literatura quanto na visão atual de mercado que faz uso da abordagem ágil.

Palavras-chave: Automação; Automatização; Teste de Software; Teste Ágil; Metodologia Ágil.

SUMÁRIO

MOTIVAÇÃO	5
OBJETIVOS	8
METODOLOGIA	9
CRONOGRAMA	11
BIBLIOGRAFIA	12

MOTIVAÇÃO

Testes de software agregam importância no desenvolvimento e na entrega do produto em relação à percepção de qualidade. Segundo Rios (2005), os testes têm por objetivo reduzir os riscos que possam prejudicar o negócio ligados a falhas do software. Para Naik e Tripathy (2008), além de proporcionar valor estratégico comercial, a atividade de teste deve estar alinhada com a missão e os objetivos da empresa.

Entretanto, é crescente a adoção das metodologias ágeis no processo de desenvolvimento de software em busca de atender à solicitação de um mercado que exige entregas cada vez mais rápidas, conforme destacado na pesquisa do Version One (2018). Diante deste cenário, surge, na área de teste de software, o desafio de testar mais rápido sem comprometer a qualidade. A tarefa de testes é árdua para as equipes segundo Pressman e Maxim (2016), pois a sua execução tende a consumir uma grande fatia de tempo e dos esforços técnicos dentro de um processo de desenvolvimento de software.

Em um contexto ágil é possível notar que garantir a qualidade do produto, otimizando o tempo no desenvolvimento do mesmo, requer alternativas como automatizar processos ligados diretamente à qualidade, tal como, o teste. Automatizar o teste contribui para reduzir o tempo gasto nesta atividade no decorrer do projeto, pois os ciclos curtos para o desenvolvimento do software das abordagens ágeis tornam, por exemplo, a realização de testes de regressão praticamente impossível.

Os testes de regressão consistem em retestar uma aplicação após sua implementação sempre que alguma mudança é realizada no software com o intuito de garantir que as funcionalidades já desenvolvidas anteriormente não tenham sido afetadas pelas alterações implementadas (BASTOS et al., 2007). Pelas boas práticas de engenharia e qualidade de software, este tipo de teste deve sempre ser executado ao anteceder uma entrega, mas na prática, infelizmente, tendem a não serem sempre executados, pois demandam muito tempo no escopo do projeto. Peres (2016, p. 39) defende que “Para testes de regressão e desempenho não há dúvida que o automatizado seja o mais adequado.” e Sommerville (2007, p. 147) salienta “Esta forma é geralmente mais rápida que o teste manual, especialmente quando envolve teste de regressão - reexecução de testes anteriores [...]”.

Na literatura é possível encontrar referências tanto ao termo Automação de Testes Software como Automatização de Testes de Software abordando o assunto relacionado a este trabalho. Sommerville (2007, p. 339) define automação de testes como “[...] uso de ferramentas de software para ajudar a reduzir o tempo e o esforço envolvidos nos processos de teste.”. A *International Software Testing Qualifications Board* (ISTQB) cita que “A automatização dos testes normalmente requer a automação tanto da execução do teste como da verificação do teste [...]”. Vincenzi et al. (2018) abordam o tema utilizando o termo “Automatização de teste” e o definem pelo seu objetivo por “[...] permitir que maior quantidade de testes seja realizada em menor tempo, evitando que o testador despenda seus esforços em atividades repetitivas e que poderiam ser realizadas de forma automatizada.”.

Um dos princípios adotados do Manifesto Ágil é “Nossa maior prioridade é satisfazer o cliente através da entrega contínua e adiantada de software com valor agregado.” (GOMES, 2016, p.3). Dentro desta cultura é praticamente indispensável o uso de ATS. De acordo com Cohn (2011), a ATS deve ser considerada uma parte incorporada ao processo de desenvolvimento de software em uma entrega contínua, assim dando suporte para que a equipe tenha sucesso na Sprint, alcançando os valores rapidamente. O mesmo enfatiza que a AST sob uma visão tradicional é considerada dispensável pelo fato de a realizarem de forma tardia. Logo, o teste tradicional perde espaço para o teste ágil que, por sua vez, adota os princípios do manifesto ágil para sua prática. Crispin e Gregory (2009) apontam que a grande diferença entre ambos está no rápido *feedback* que o teste ágil concede à equipe, impulsionando o projeto.

Conforme pesquisa divulgada pela QTRICENTS QTEST (2018), em torno de 72% das empresas já praticam a automatização de teste, sendo que dessas 76% são adeptas às Metodologias Ágeis. Ainda que empresas busquem adotar a AST, ela ainda é um desafio quanto à sua implementação com sucesso. Para Mulchandani (2013), esse insucesso está ligado à inexistência de um processo guiado para tal dentro das equipes. Crispin e Gregory (2009) chamam a atenção para o desestímulo das equipes gerado pela ausência de métricas no decorrer do processo. Costa (2006) ainda a falta de consciência sobre a similaridade de um processo de desenvolvimento de software ao de automação de testes, a qual contribui para o fracasso de uma boa implementação.

Na literatura é possível identificar práticas para o processo de desenvolvimento de software voltado ao contexto ágil e orientadas a teste, duas delas são o *Test Driven Development* (TDD) e o *Acceptance Test-Driven Development* (ATDD). A primeira orienta a adoção de uma abordagem na qual os testes automatizados de unidade sejam escritos antes mesmo do real

desenvolvimento, que por sua vez, será realizado com iterações baseadas nestes testes (BECK, 2002). Já o ATDD segue a mesma premissa sobre a ordem da escrita e desenvolvimento, porém dissemina a prática de envolver diferentes perspectivas (desenvolvedores, testadores e clientes) para a escrita dos testes de aceitação (GARTNER, 2013).

Embora ambos englobem práticas para o processo de desenvolvimento, estes não fazem especificações a todas as atividades de AST existentes. Assim como *frameworks* Scrum e XP (*Extreme Programming*), que atendem a Metodologias Ágeis, a utilização de testes automatizados no processo de desenvolvimento é valorizada, porém não há um modelo proposto para realizar esta implementação. Diante de tal, assume-se, então, como questão de pesquisa: “Como seria um modelo de processo para automatização de teste em ambiente ágil quanto à sua implementação para que contribua com a qualidade e uma entrega de software contínua?”.

Para a geração do proposto modelo serão considerados fatores críticos que impactam na dificuldade de implementação da ATS com base na percepção da indústria atual de desenvolvimento de software que adota práticas ágeis. Isto posto, através da revisão bibliográfica a respeito das melhores práticas sobre a qualidade e desenvolvimento de software, construir um artefato que sustente a construção do arquétipo do processo de implementação de ATS concomitante às metodologias ágeis empregadas em equipes de desenvolvimento.

OBJETIVOS

Objetivo Geral

O objetivo geral desta pesquisa é elaborar um modelo de processo para implementação de testes automatizados em um ambiente que emprega metodologias ágeis no desenvolvimento de software, de modo que a adoção do mesmo atenda paralelamente aos princípios da agilidade e contribua para a efetividade e a qualidade do processo.

Objetivos Específicos

- Sistematizar os conceitos básicos para automatização de testes de software existentes na literatura;
- Mapear os conceitos e características do que se entende por teste ágil;
- Fazer um levantamento sobre limitações e expectativas realistas sobre automação de testes em equipes reais;
- Desenvolver o modelo de processo para automatização de teste em ambiente ágil;
- Aplicar uma pesquisa na área de interesse para uma pós análise de viabilidade de adoção do modelo proposto.

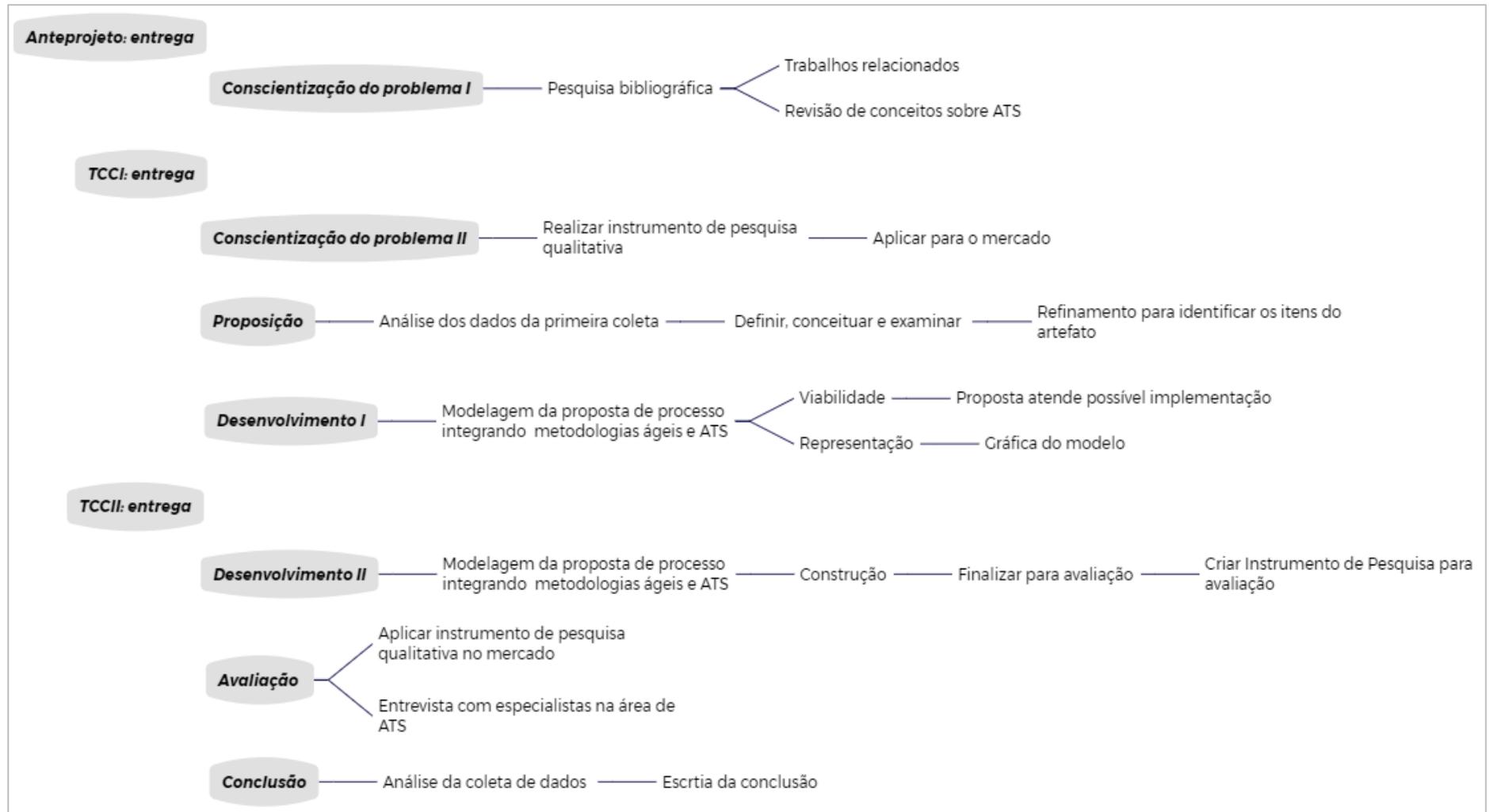
METODOLOGIA

Este trabalho é classificado como pesquisa aplicada por sua natureza, com finalidade de aplicação dos conhecimentos para construção de uma proposta de modelo que suportará sua adoção em ambientes conforme definido na pesquisa (PRODANOV; FREITAS, 2013). Será adotado o método científico *Design Science Research* (DSR) que objetiva desenvolver artefatos que possibilitem gerar soluções satisfatórias e práticas para um determinado problema (DRESCH; LACERDA; ANTUNES, 2013).

Para estruturação da pesquisa, o projeto será agrupado em fases, conforme orienta o método adotado, gerando por fim um artefato modelo caracterizado por um conjunto de proposições ou declarações que expressam relações entre elementos conceituais e representações da realidade (DRESCH; LACERDA; ANTUNES, 2013).

A Figura 1 representa as fases com suas respectivas atividades abordadas no decorrer do projeto, a fim de alcançar os objetivos da investigação. As fases de conscientização do problema e de desenvolvimento serão divididas em duas versões. Ambas com a proposta de gerar um instrumento de pesquisa em sua segunda versão, no qual para a fase de conscientização terá o foco de elencar os fatores críticos do problema para geração do artefato na percepção de mercado. E, para a de desenvolvimento, realizar a validação do modelo proposto.

Figura 1 - Passos da pesquisa conforme fases DSR



Fonte: elaborado pela autora

CRONOGRAMA

Trabalho de Conclusão I

Etapa	Meses			
	Mar	Abr	Mai	Jun
Conscientização do Problema I				
Revisão Anteprojeto				
Entrega Anteprojeto				
Conscientização do Problema II				
Proposição				
Desenvolvimento I				
Revisão TCC1				
Entrega TCCI				

Trabalho de Conclusão II

Etapa	Meses			
	Ago	Set	Out	Nov
Desenvolvimento II				
Avaliação				
Conclusão				
Revisão TCC2				
Entrega TCC2				

BIBLIOGRAFIA

BASTOS, Aderson S.; RIOS, Emerson; CRISTALLI, Ricardo S.; MOREIRA FILHO, Trayahu Rodrigues. **Base de conhecimento em teste de software**. 2.ed. São Paulo, SP: Martins, 2007.

BECK, Kent. **TDD desenvolvimento guiado por testes**. Porto Alegre, RS: Bookman, 2010.

COHN, Mike. **Desenvolvimento de software com Scrum: Aplicando métodos ágeis com sucesso**. Porto Alegre, RS: Bookman, 2011.

COSTA, Mozart Guerra. **Estratégia de Automação em Testes: Requisitos, Arquitetura e Acompanhamento de sua Implementação**. 2004. 102 f. Trabalho de Conclusão de Curso (Mestrado) – Engenharia de Computação, Instituto de Computação Universidade Estadual de Campinas (UNICAMP), Campinas, SP, 2004.

CRISPIN, Lisa; GREGORY, Janet. **Agile Testing: A Practical Guide for Testers na Agile Teams**. Boston, MA: Pearson Education Inc, 2009.

DRESCH, Aline; LACERDA, Daniel Pacheco; ANTUNES, José A. V. **Design science research: método de pesquisa para avanço da ciência e tecnologia**. Porto Alegre, RS: Bookman, 2015.

GARTNER, Markus. **ATDD by Example A Practical Guide to Acceptance Test-Driven Development**. Boston, MA: Pearson Education Inc, 2013.

GOMES, André Faria. **Agile: Desenvolvimento de software com entregas frequentes e foco no valor de negócio**. 2. ed. Novo Hamburgo, RS: Feevale, 2013.

NAIK, Kshirasagar; TRIPATHY, Priyadarshi. **Software Testing and Quality Assurance: Theory and Practices**. Hoboken, Nova Jersey:Wiley, 2008.

PERES, Hugo. **Automatizando Testes de Software Com Selenium**. 1. ed. Porto Alegre, RS: SIMPLÍSSIMO, 2016.

PRESSMAN, Roger S.; MAXIM, Bruce R. **Engenharia de software: uma abordagem profissional**. 8. ed. Porto Alegre, RS: AMGH, 2016.

PRODANOV, Cleber Cristiano; FREITAS, Ernani Cesar de. **Metodologia do trabalho científico: métodos e técnicas da pesquisa e do trabalho acadêmico**. 2. ed. Novo Hamburgo, RS: Feevale, 2013.

QTRICENTIS QTESTE. **The Evolution of Test Automation**. 2018. Disponível em: <<https://www.qasymphony.com/landing-pages/report-the-evolution-of-test-automation/>>.

Acesso em: 03 abril. 2019.

RIOS, Emerson. **Análise de riscos em projetos de teste de software**. Rio de Janeiro, RJ: ALTA BOOKS, 2005.

SOMMERVILLE, Ian. **Engenharia de Software**. 9. ed. São Paulo, SP: PERSON, 2007.

MULCHANDANI, Bharat. Software Test Conference. **Why test automation fails**. 2013. Disponível em: < http://conference.qaiglobalservices.com/stc2013/PDFs/Bharat_K_Mulchandani.pdf >. Acesso em: 30 março. 2019.

VERSION ONE. **The 12th Annual State of Agile Report**. 2018. Disponível em: < <https://explore.versionone.com/state-of-agile/versionone-12th-annual-state-of-agile-report> >. Acesso em: 17 março. 2019.

VINCENZI, Auri Marcelo Rizzo et. al. **Automatização de Teste de Software com Ferramentas de Software Livre**. Rio de Janeiro, RJ: Elsevier, 2018.