UNIVERSIDADE DO SUL DE SANTA CATARINA

ANA CLARA GONZAGA BARROS

QUALIDADE DE SOFTWARE E SUAS ABRANGÊNCIAS: O ESTUDO DA APLICAÇÃO EM AMBIENTE ORGANIZACIONAL ÁGIL

Rio de Janeiro

2019

ANA CLARA GONZAGA BARROS

QUALIDADE DE SOFTWARE E SUAS ABRANGÊNCIAS: O ESTUDO DA APLICAÇÃO EM AMBIENTE ORGANIZACIONAL ÁGIL

Relatório apresentado ao Curso **Tecnólogo em Gestão da Tecnologia da Informação,** da Universidade do Sul de Santa Catarina, como requisito parcial à aprovação na unidade de aprendizagem de Estudo de Caso.

Orientadora: Profª. Nilce Miranda Ayres

Rio de Janeiro

2019

ANA CLARA GONZAGA BARROS

QUALIDADE DE SOFTWARE E SUAS ABRANGÊNCIAS: O ESTUDO DA APLICAÇÃO EM AMBIENTE ORGANIZACIONAL ÁGIL

Este trabalho de pesquisa na modalidade de Estudo de Caso foi julgado adequado à obtenção do grau de Tecnólogo em Gestão da Tecnologia da Informação e aprovado, em sua forma final, pelo Curso Superior de Tecnologia em Gestão da Tecnologia da Informação, da Universidade do Sul de Santa Catarina.

Rio de Janeiro, 10 de Maio de 2019.

Nilce Miranda Ayres, Me.
Universidade do Sul de Santa Catarina

AGRADECIMENTOS

Agradeço aos meus pais que me ensinaram que através do conhecimento eu teria todas as minhas conquistas e sempre estão ao meu lado. A minha mãe que me ensinou o gosto pelos livros. Ao meu marido que é meu fã de carteirinha e sempre apoiou minha carreira mesmo enquanto era apenas meu amigo. Agradeço aos meus filhos que ao correr em volta da mesa me fizeram ler os parágrafos diversas vezes e ter certeza do que estava construindo. A eles que tornaram minha caminhada na graduação mais prazerosa, mesmo que difícil. Finalmente, agradeço a minha equipe, que me constrói enquanto líder a cada dia.

RESUMO

O presente estudo, visa analisar alguns aspectos relacionados a qualidade de um software. Acompanha processos como code review (revisão de código) que é a revisão de todo e qualquer código inserido no repositório da companhia; A modelagem de produto que é a concepção dos requisitos de um software; A execução de sprints de construção ou manutenção do software ou fragmento deste que consiste no desenvolvimento de um conjunto de demandas, sejam funcionalidades ou bugs (erros); Analisa a execução e presença de vários tipos de testes, incluindo os testes automatizados que visam garantir o funcionamento do que foi desenvolvido e testes unitários que visam validar cada parte do código escrito. Recorre ao método descritivo para realização da pesquisa. Há observação in loco do dia a dia das esquipes de uma companhia brasileira do ramo educacional. Aplica um questionário virtual que será disponibilizado aos funcionários da companhia observada. Verifica que a companhia já executa alguns processos que garantem a qualidade do software como um todo. Realiza uma análise onde observa pontos fortes, como o processo de revisão de código e a concordância unânime de que alguns tipos de testes podem ser incluídos em uma cultura ágil. Pontos fracos, como a não aplicação da revisão de código por todas as equipes. E sugeridas melhorias em alguns processos como a participação do cliente no processo de concepção do produto e o aumento da cobertura de testes automatizados.

Palavras-chave: Qualidade de software, Cultura ágil, Code review, Aplicação da Qualidade de software, Qualidade e ágil.

SUMÁRIO

1 INTRODUÇÃO **6**

2 TEMA **7**

3 OBJETIVOS **8**

3.1 OBJETIVO GERAL 8

3.2 OBJETIVOS ESPECÍFICOS 8

4.1 CAMPO DE ESTUDO 9

4.2 INSTRUMENTOS DE COLETA DE DADOS 9

5 APRESENTAÇÃO E ANÁLISE DA REALIDADE OBSERVADA **10**

5.1 ESTRUTURA DA ORGANIZAÇÃO 10

5.2 DESCRIÇÃO E ANÁLISE DA REALIDADE OBSERVADA 10

6 PROPOSTA DE SOLUÇÃO DA SITUAÇÃO-PROBLEMA **14**

6.1 PROPOSTA DE MELHORIA PARA A REALIDADE ESTUDADA 14

6.2 RESULTADOS ESPERADOS 15

6.3 VIABILIDADE DA PROPOSTA 16

7 CONSIDERAÇÕES FINAIS **17**

REFERÊNCIAS **18**

APÊNDICE 1 **19**

APÊNDICE 2 **20**

1 INTRODUÇÃO

Em meados dos anos 80 a 90, a comunidade de software mundial começou algumas iniciativas para melhorar a qualidade das entregas. No Brasil, a qualidade de software foi mais disseminada após a criação do MPS.BR (Melhoria do processo de software brasileiro) que teve início em 2003. A qualidade de software envolve inciativas de várias áreas em uma organização, sendo necessárias ações multidisciplinares que tem como finalidade a melhoria de tudo que se faz para e pelo software.

Com o objetivo de coletar as informações de aplicações práticas da qualidade, a pesquisadora acompanha as equipes multidisciplinares de uma companhia que aplica a cultura ágil no desenvolvimento de seus softwares. O estudo disponibiliza objetivos e propostas de soluções para melhoria contínua da qualidade do software.

O trabalho está organizado inicialmente exibindo os objetivos, o campo de estudo e os instrumentos de coletas de dados. Posteriormente apresenta a análise da realidade observada e a estrutura da organização. Também explicita a proposta de solução da situação problema, os resultados esperados e a viabilidade da proposta. Finalizando com as considerações finais. Vale ressaltar que este estudo de caso não tem a pretensão de se mostrar como um trabalho acabado e sim, um recorte de problemas mais comuns e explicita ações e sugestões de aplicações com o intuito de potencializar a qualidade do software estudado.

2 TEMA

A qualidade de software tem sido desenvolvida e valorizada por organizações ao redor do mundo. Segundo Wong (2006, p.131) define-se que “A qualidade é uma característica intrínseca e multifacetada de um produto.” Analisando essa definição, refletimos que a qualidade de software é reflexo de várias áreas no processo de elaboração e construção de um software. Quando um analista de teste encontra muitas falhas ou há muitas falhas em produção não é mérito ou demérito da equipe de qualidade e sim a evidenciação de outros problemas no processo de construção do software. É necessário que o software que está sendo construído tenha confiabilidade. Segundo Delamaro (2016, p.16) “Podemos definir confiabilidade como a tendência de um software em operar de maneira correta.”

Dentro da área de qualidade de software existem várias abrangências, porém neste estudo será destacada a importância de ter quatro (4) tipos de testes no desenvolvimento de software. São eles: testes unitários, testes de integração, teste de ponta e testes manuais. A questão é: Realmente precisamos de tantos tipos de teste? Segundo Gomes (2013, p33) “Diferentes tipos de testes têm objetivos distintos e seu custo de automação e tempo de execução também diferem.” Dentro deste contexto, o presente estudo busca responder o seguinte questionamento: Será que mesmo em empresas com cultura ágil, conseguimos implementar e executar todos estes(os) testes?

O presente estudo, terá reflexões e análises realizadas ao longo da vida profissional da pesquisadora. O interesse pelo tema se deu quando a pesquisadora se viu coordenando a área de qualidade e identificou vários problemas como código falho, requisito elaborado de forma incorreta, a não participação do cliente no processo, entre outros fatores. Vários destes fatores resultavam em "bugs" no software e eram tidos como de única e inteira responsabilidade da área de qualidade de software. A importância social da pesquisa está na possibilidade de trazer a reflexão para a comunidade de software e sensibilizar em relação ao dever que todos têm em realizar entregas de qualidade.

3 OBJETIVOS

3.1 OBJETIVO GERAL

Analisar aspectos relacionados a qualidade do software, incluindo a aplicação dos quatro (4) tipos de teste a serem aplicados no software durante sua fase de construção e vida útil, pretendendo entender se estes são aplicáveis à cultura ágil.

3.2 OBJETIVOS ESPECÍFICOS

Verificar a opinião de membros da comunidade de software em relação à aplicação dos 4 (quatro) tipos que testes, que são: testes unitários, testes de integração, testes automatizados e testes manuais/exploratórios e seus impactos na qualidade do software.

Analisar se há aplicação de testes automatizados.

Analisar o conceito de qualidade de código que é pouco abrangido quando a qualidade do software é analisada e por vezes quando este é desenvolvido.

Observar como a ausência de participação do cliente final nas reuniões de levantamento de requisitos fere a qualidade.

Concluir se estes tipos de testes são aplicáveis à cultura ágil.

4 PROCEDIMENTOS METODOLÓGICOS

4.1 CAMPO DE ESTUDO

A pesquisa será do tipo descritiva e analisará o enredo da construção de um ou mais softwares na SOMOS Educação, que é uma empresa privada do ramo de educação e investe em aplicações tecnológicas na educação. “As pesquisas descritivas têm como objetivo principal a descrição das características de determinada população ou fenômeno, estabelecendo correlações entre variáveis.” (GIL, 2002, p.42) As observações também serão feitas na empresa SOMOS Educação que tem escritórios de tecnologia em São Paulo capital e Rio de Janeiro capital e por meio de entrevistas coletadas em meio virtual.

4.2 INSTRUMENTOS DE COLETA DE DADOS

Quadro 1 – Instrumento de coleta de dados

Instrumento de coleta de dados	Universo pesquisado	Finalidade do Instrumento
Entrevista	Membros dos “squads” e capítulos que participam efetivamente da construção do software Plurall, o questionário será entregue a 14 (quatorze) pessoas.	Analisar e documentar a opinião dos membros de alguns squads, para entender as necessidades do cotidiano na aplicabilidade dos testes de software.
Observação direta dos participantes	Serão observados os “sprints” de construção de novas funcionalidades e manutenção do software Plurall.	Observar se são executados os 4 (quatro) tipos de testes e se é considerada a qualidade do código, além do envolvimento do cliente final na definição das tarefas a serem executadas para construir o software.
Documentos	Serão pesquisados relatórios de retrospectivas dos “sprints” e artigos digitais disponíveis na internet.	Analisar comportamentos passados e atuais nos processos de construção e desenvolvimento do software.

Fonte: Do autor.

5 APRESENTAÇÃO E ANÁLISE DA REALIDADE OBSERVADA

5.1 ESTRUTURA DA ORGANIZAÇÃO

A Kroton Educacional fundada em 1966, é um dos maiores grupos educacionais brasileiros e tem expressivo valor de mercado. Inovadora e líder no desenvolvimento e aplicação de tecnologias educacionais. A SOMOS Educação faz parte deste grupo e tem foco na educação básica, que é o nível de ensino correspondente aos primeiros anos de educação escolar. Esta, foi fundada em 2010 e tem como missão transformar a sociedade por meio da educação (SOMOS Educação, 2019). A SOMOS tem sede em Pinheiros – São Paulo e possui alguns escritórios distribuídos pelo Brasil. Entre eles, um escritório de tecnologia que é localizado no bairro Ipanema no Rio de Janeiro/RJ. Um dos principais sistemas desenvolvidos pela companhia é o Plurall que está disponível para plataformas Web e Mobile. Entre outras funcionalidades, o Plurall disponibiliza tutores que respondem a dúvidas de alunos durante seus estudos e possibilita que estes realizem testes e tarefas em sua plataforma. Também gere a criação de turmas e a organização dos alunos e equipe das escolas que utilizam o Plurall.

A equipe de tecnologia que desenvolve e mantém o Plurall está localizadas nos escritórios da Somos Ipanema, Bauru e Pinheiros. Esta foi a equipe analisada neste estudo de caso. Esta equipe é organizada em um modelo de cultura ágil chamado "squads", modelo este que foi difundido na comunidade de software pela empresa Spotify. Em resumo, um squad é um pequeno grupo de especialistas auto organizados e de diferentes funções. Um formato que geralmente é aplicado é composto por: desenvolvedor frontend, desenvolvedor backend, product owner, analista de qualidade e designer. Eles possuem responsabilidade ponta a ponta, ou seja, projetam, desenham, desenvolvem e dão manutenção a um ou mais produtos. Os squads têm um nível elevado de autonomia, onde decidem o que construir, como construir e como trabalhar juntos para construir. As pessoas observadas e que responderam o questionário deste estudo pertencem a 1 (um) ou mais squads do Plurall.

5.2 DESCRIÇÃO E ANÁLISE DA REALIDADE OBSERVADA

Foi realizada uma observação presencial do dia a dia dos squads (grupos multidisciplinares de trabalho). Cada squad trabalha com sprints de 2 (duas) semanas e nestes sprints são realizadas algumas cerimônias. Entre elas as retrospectivas que pontuam fatos positivos e negativos que ocorreram durante os sprints, além de ações a serem ajustadas ou melhoradas nos próximos sprints. Na observação presencial, foi feita uma avaliação de documentos das retrospectivas de sprints dentro do contexto da organização. Também foi observado organicamente o comportamento dos membros dos squads. Durante as observações, alguns fatos foram identificados.

O primeiro fato é que embora os envolvidos no processo de construção de software entendam que seja possível em uma cultura ágil a aplicação em conjunto dos 4 (quatro) tipos de testes destacados neste estudo (testes unitários, testes de integração, testes de ponta e testes manuais e exploratórios), não é a realidade do universo em que todos trabalham pois não há aplicação concomitante em todos os projetos devido a uma resistência quanto aos testes unitários e de integração por causa do tempo de desenvolvimento consumido por estes.

Um outro fato é que embora o cliente seja envolvido no processo de definição do escopo a ser desenvolvido numa determinada parte do software, ainda há falhas nessas definições por causa de solicitações indevidas que não são corretamente absorvidas e entendidas pela equipe de produto. Segundo Mey (2019, p.25), “enquanto estiver tentando tornar seu produto compreensível e específico, você evidenciará suposições elaboradas sobre os problemas dos clientes.”

Um fato importante identificado é que foi implantado o processo de “code review” (toda vez que um código é criado ou incrementado, passa pela aprovação de pelo menos três (3) membros da equipe de desenvolvedores do software) para que a qualidade do código seja elevada, sempre considerada e mantida por todas as equipes. Nessa prática, o conceito de teste unitário é exaltado pelos desenvolvedores e disseminado entre as equipes. Segundo Maia (2014, p.21) os “Testes unitários validam cada unidade de funcionalidade de nossa aplicação e, portanto, compõem a maioria dos conjuntos de testes.”

Além desta análise presencial, foi aplicado de forma digital um questionário (Apêndice 1) contendo 5 (cinco) perguntas dentro do contexto dos objetivos deste estudo de caso. Ao todo 14 (quatorze) pessoas responderam ao questionário (Apêndice 2). É possível analisar as respostas através de gráficos.

Gráfico 1 – Total de respostas em relação a pergunta “É possível aplicar estes testes na cultura Agile?”.

Fonte: Autor.

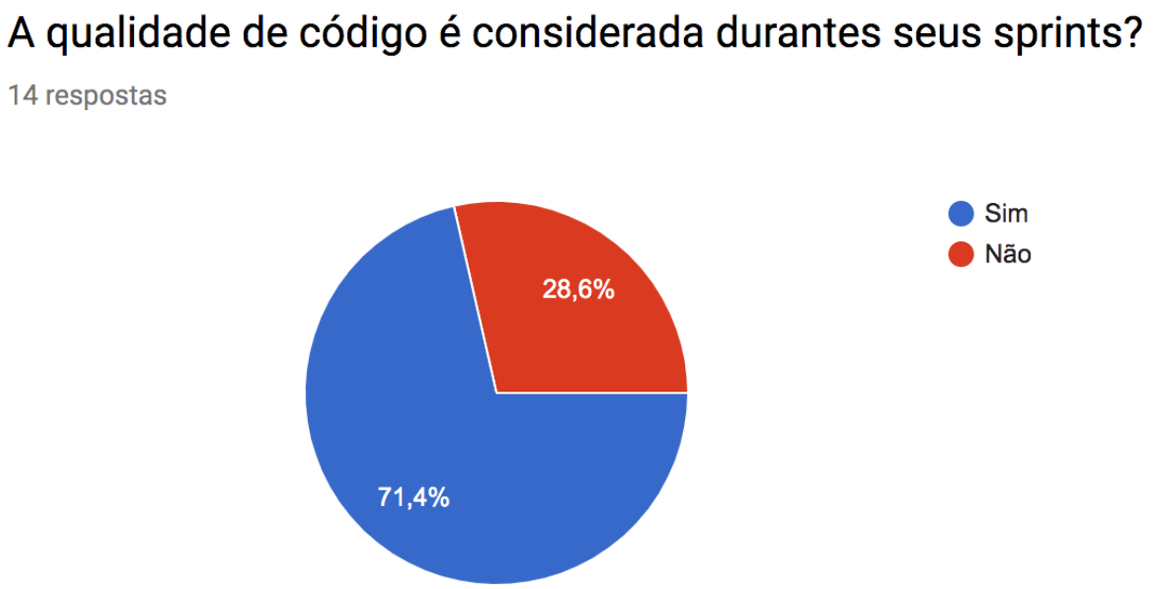


Gráfico 2 – Porcentagem das respostas para a pergunta “A qualidade de código é considerada durante seus sprints?”.

Fonte: Autor.

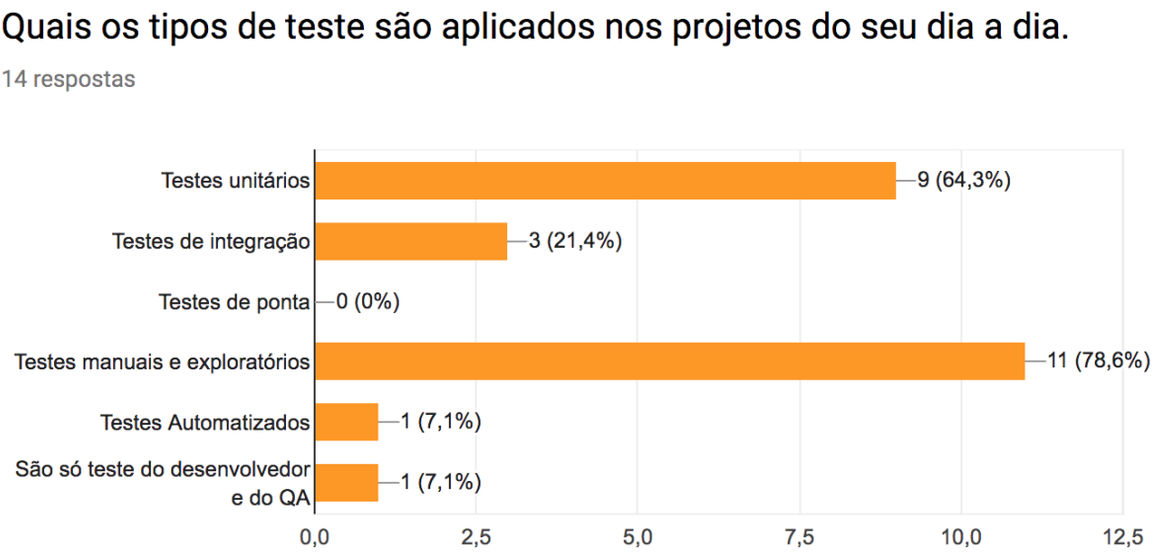


Gráfico 3 – Porcentagem de respostas para a pergunta “Quais os tipos de testes são aplicados nos projetos do seu dia a dia.”

Fonte: Autor

Verifica-se que 100% dos entrevistados (gráfico 1) acreditam ser possível aplicar os 4 (quatro) tipos de testes em um ambiente com cultura ágil. E 71,4% dos entrevistados (gráfico 2) acreditam que a qualidade do código é levada em consideração durante os sprints. Já a cobertura de testes automatizados em relação aos testes manuais ainda é expressivamente pequena (gráfico 3). Segundo Aniche (2015, p.23) “É difícil garantir qualidade de um sistema sem testes automatizados...”.

De acordo com o entrevistado Gabriel Pireda, “Não possuir testes unitários e automatizados, fere a qualidade do software”. A partir destas análises, o resultado da pesquisa concorda com a análise presencial.

Conclui-se que os principais pontos negativos, que ferem a qualidade do software são a não aplicação de testes unitários no código, uma baixa cobertura de testes automatizados e requisitos mal definidos e modelados. Como pontos positivos, é unânime entre os entrevistados que os tipos de teste problematizados neste estudo são aplicáveis em uma cultura ágil. E que é possível investir, evoluir e conviver com a melhoria da qualidade de código através destes concomitante a outras ações.

6 PROPOSTA DE SOLUÇÃO DA SITUAÇÃO-PROBLEMA

6.1 PROPOSTA DE MELHORIA PARA A REALIDADE ESTUDADA

A partir das análises, verificações e observações realizadas, sugere-se que a Somos Educação realize alteração de alguns processos, mantenha outros e faça ações nos squads envolvidos no processo de elaboração, desenvolvimento, manutenção e evolução do software Plurall. Em busca de sanar os problemas analisados e indicados pelo presente estudo seguindo os objetivos explanados, apresenta-se a seguir algumas ações e práticas a serem aplicadas.

a) Aplicação de testes unitários

A proposta é implementar a aplicação de testes unitários após a finalização da construção de um código. Consiste em toda vez que um desenvolvedor construir um trecho de código, deve construir também um teste que valide o que acabou de desenvolver. Quando este teste falhar, significa que há um problema no trecho do código. A proposta consiste em a longo prazo, esta etapa de testes ser obrigatória em todo código escrito.

b) Envolvimento do cliente no processo

Propõe-se que seja definida como obrigatória a participação do cliente na definição dos requisitos do software ou fragmento deste a ser desenvolvido. Toda reunião inicial do produto, o cliente será participante ativo na construção dos requisitos e definições de critérios de aceite. É possível realizar essa participação de forma presencial ou à distância. Aliada a essa prática, seja instalada uma etapa de validação com o cliente antes da entrega final do software ou fragmento deste.

c) Manter a qualidade do código

Após observação de que o processo de code review tem sido favorável a manutenção da qualidade de software em alguns squads. Propõe-se que este processo seja mantido onde já é aplicado, seja implantado nos que ainda não mantém essa prática e posteriormente seja definido como obrigatório em todos os squads. Para aplicação e manutenção desse processo, que seja definido um conjunto de regras mínimas para as solicitações de aprovações de códigos. Regras estas que auxiliarão a liderança no controle de qualidade e na garantia da manutenção desta. É necessário estipular um grupo de desenvolvedores que ficará responsável por cobrar essas regras e garantir que estão sendo seguidas.

d) Aumentar a cobertura de testes automatizados

Foi explanado no presente estudo que é de suma importância a presença dos testes automatizados na validação do software. Recomenda-se que seja feito um levantamento da cobertura destes testes em todos os squads e posteriormente, um planejamento de aumento da cobertura de cada um. No planejamento e em sua execução, recomenda-se que sejam levados em consideração inicialmente, os testes das funcionalidades mais complexas do sistema e finalmente sejam considerados os ditos “testes burros” que contemplam os fluxos mais simples e corriqueiros dos sistemas. Haverá necessidade de treinamento da equipe de qualidade para que execute os testes manuais e escreva os códigos dos testes automatizados.

6.2 RESULTADOS ESPERADOS

A partir do que foi proposto, espera-se maximizar a capacidade dos squads em manter a qualidade de todos os softwares desenvolvidos pela companhia. Além disso estima-se que à aplicação dos tipos de testes e processos indicados melhore consideravelmente o dia a dia das equipes envolvidas no processo de construção do software ou fragmento deste e potencialize a capacidade de construção de um produto adequado ao que foi proposto e usual.

a) Aplicação de testes unitários

A curto prazo, espera-se que haja um gasto maior de tempo durante os sprints para implantar os testes automatizados, visto que não é uma característica da maioria dos desenvolvedores e é uma nova cultura. A médio prazo os testes unitários já escritos e em produção, serão um alerta para possíveis bugs inseridos no sistema com novos códigos. A longo prazo quando a cobertura de testes unitários estiver em pleno desenvolvimento, vai se gastar menos tempo com manutenção do software e mais tempo com itens cruciais como a atualização de tecnologia aplicada ou novas funcionalidades (responsáveis pela evolução do software).

b) Envolvimento do cliente no processo

A curto prazo espera-se que o engajamento dos clientes nas reuniões de definição do produto (requisitos e afins) seja baixo pois há a necessidade de visualização de valor nesse processo por parte do próprio cliente. A médio prazo o valor vai sendo evidenciado devido aos resultados de outros clientes, à evolução do processo e os testes realizados pelo cliente antes do software chegar em produção. A longo prazo espera-se que o produto amadureça o suficiente para ser projetado e desenvolvido com a finalidade esperada pelo cliente.

c) Manter a qualidade do código

Inicialmente, espera-se que haja um choque de cultura na equipe de desenvolvimento pois a revisão de código é um processo delicado. A médio prazo espera-se que as resistências sessem e a equipe se desenvolva e evolua muito progressivamente em termos técnicos, analíticos e de lógica de programação. A longo prazo haverá uma riqueza técnica muito grande para a companhia e uma cultura de trabalho colaborativo louvável.

d) Aumentar a cobertura de testes automatizados

A curto prazo haverá uma redução de velocidade de trabalho da equipe de testes devido a automação dos códigos legados e atuais na aplicação do planejamento de aumento de cobertura. A médio e longo prazo a equipe ganhará tração e conseguirá manter a qualidade do sistema de forma ágil e mais competente. Espera-se que os tempos de testes de regressão diminuam, assim como bugs no sistema.

6.3 VIABILIDADE DA PROPOSTA

Observou-se que a direção do setor de tecnologia da Somos Educação tem interesse, está disponibilizando tempo e empenhando-se para aplicar melhorias em toda a área digital e organizar os squads a fim de elevar a qualidade do Plurall de forma geral. Porém alguns aspectos precisam ser levados em consideração. São eles:

1º Viabilizar treinamento interno das equipes que atendem o cliente para que estas possam melhor orienta-los no processo de participação da concepção do produto. Também um treinamento para a equipe de produto com o fim de implantar o processo junto ao cliente.

2º Disponibilizar lideranças para estarem a frente dos planejamentos de revisões de código e suas regras. Considerando viagens e encontros de equipes entre escritórios.

3º Considerar o aumento da equipe de testes e disponibilizar cursos de treinamento que variam entre R$70,00 e R$3.000,00 por analista.

4º Considerar no próximo orçamento trimestral os gastos com treinamentos internos, viagens e abertura de vagas de acordo com os cargos necessários.

7 CONSIDERAÇÕES FINAIS

O presente estudo, buscou analisar e explicitar aspectos que impulsionam a qualidade do software Plurall da companhia Kroton Educacional. Com o objetivo de analisar à aplicação dos tipos existentes de testes e a capacidade de manter a qualidade do produto através de vários aspectos. Aspectos estes que envolvem a qualidade do código, análise do produto e aplicação de testes específicos.

Entende-se que executando as propostas explanadas neste estudo, a companhia seja capaz de se reafirmar no mercado tecnológico educacional e tornar-se referência no que diz respeito a qualidade de software.

O maior desafio deste trabalho foi elaborar propostas que sejam tangíveis e alcançáveis, visto que o software Plurall é uma plataforma que está em produção com mais de 100 (cem) mil alunos, tem uma estrutura de código legado e uma estrutura organizacional definida (squads).

Um dos obstáculos foi encontrar resistência de alguns funcionários da companhia em fornecer informações para a acadêmica. Devido ao fato de que a qualidade de software não é um assunto simples de discorrer, mas fácil de se discordar. Recomenda-se que as propostas sejam aplicadas e que as ações sejam executadas na companhia considerando curto, médio e longo prazo. Sugere-se a continuação dos estudos desses aspectos que envolvem a qualidade de software e sua aplicação em ambientes organizacionais que tenham a cultura ágil.

REFERÊNCIAS

ANICHE, Maurício. **Testes automatizados de software: Um guia prático**, Casa do código, 2015.

DELAMARO, Marcio Et al, **Introdução ao teste de software**, Editora Elsevier, 2016.

GIL, Antônio Carlos. **Como elaborar projetos de pesquisa**. 4. ed. São Paulo: Atlas, 2002. 42p.

GOMES André FARIA, **Agile: Desenvolvimento de Software com entregas frequentes e foco no valor de negócio**, Casa do Código,2013.

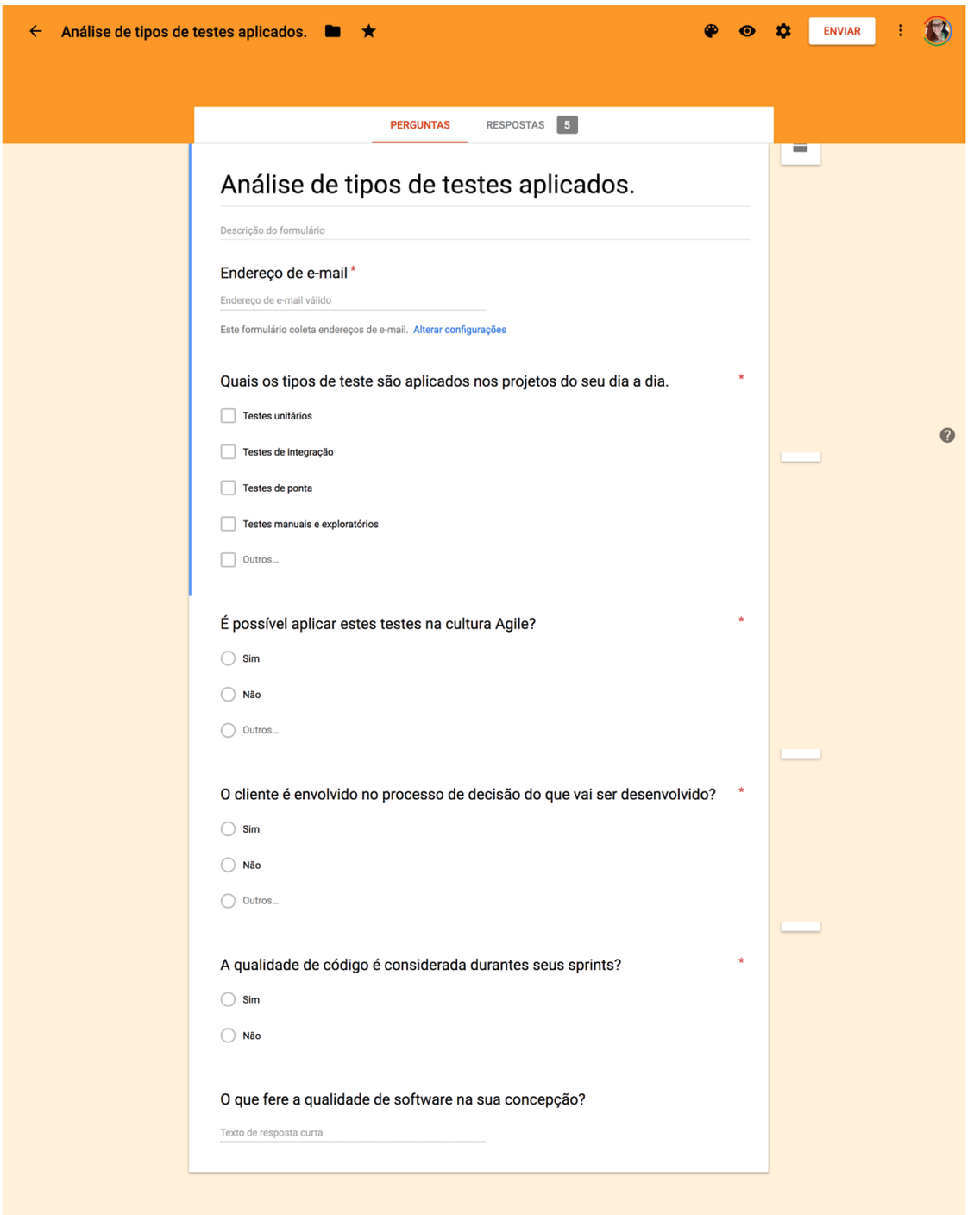
MAIA, Fábio e DUARTE, Alabê, **PERSPECTIVES ON AGILE SOFTWARE TESTING**, Thought Works, 2014. 21p.

MEY, Chris VANDER. **Lições Práticas Sobre Construção e Lançamento de Software Aprendidas no dia a dia de trabalho no Google e na Amazon.** NOVATEC, 2019.

SOMOS EDUCAÇÃO. **Quem somos.** Disponível em: http://www.somoseducacao.com.br/quem-somos. Acesso em: 5 mai. de 2019.

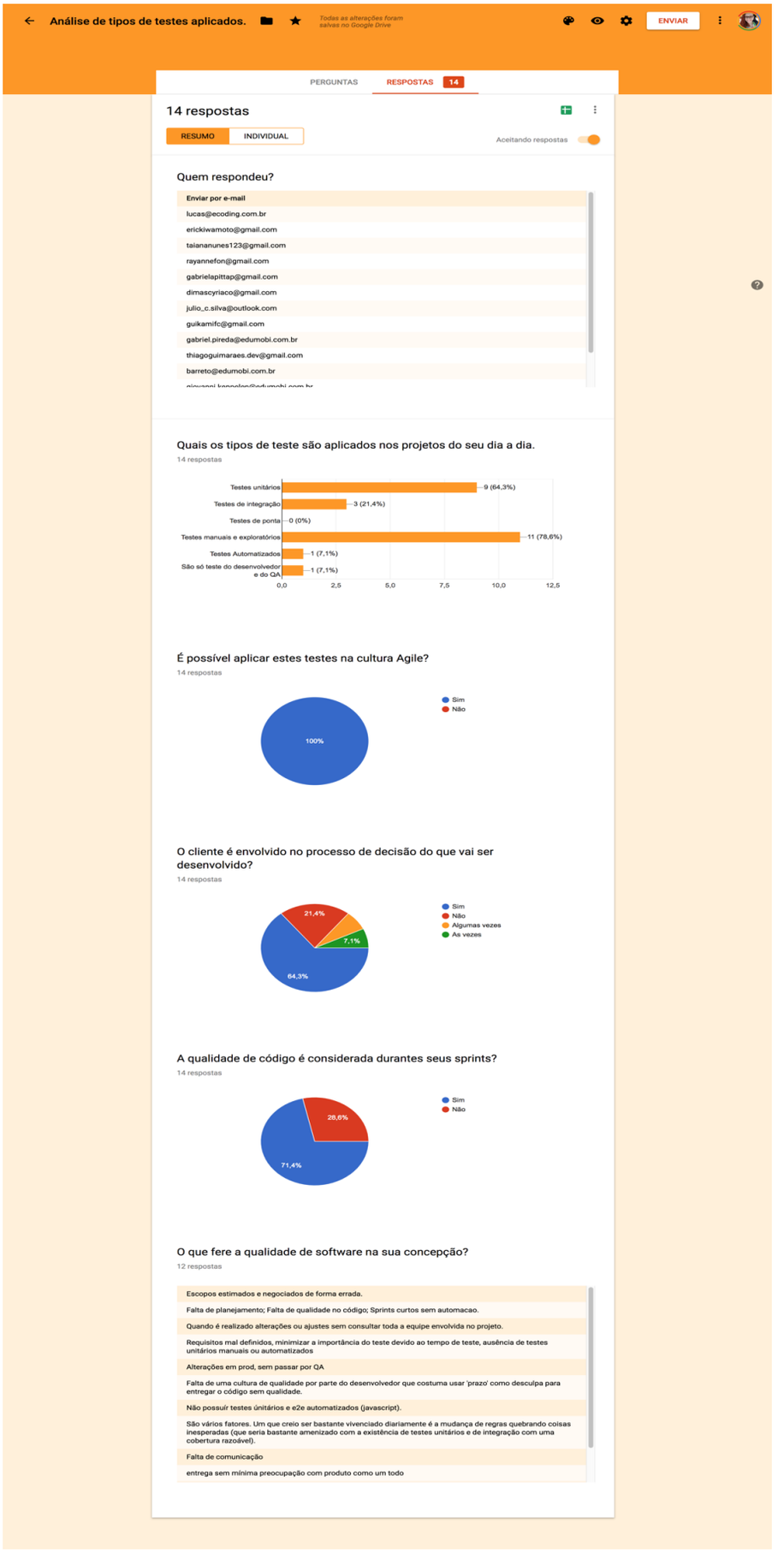
WONG, Yuk Kuen. **Modern Software Review:Techniques and Technologies.** IRM Press, 2006. p.131

APÊNDICE 1



Disponível em: https://docs.google.com/forms/d/e/1FAIpQLSd-wrY0pI2G-MJgCog532KVjntf8i_1qI-Pv_NFwLHQZCQlPQ/viewform?usp=sf_link

APÊNDICE 2


Análise de tipos de testes aplicados.
Todas as alterações foram salvas no Google Drive
ENVIAR
PERGUNTAS
RESPOSTAS 14
14 respostas
RESUMO
INDIVIDUAL
Aceitando respostas
Quem respondeu?
Enviar por e-mail
lucas@ecoding.com.br
erickiwamoto@gmail.com
taiananunes123@gmail.com
rayannefon@gmail.com
gabrielapittap@gmail.com
dimascyriaco@gmail.com
julio_c.silva@outlook.com
guikamifc@gmail.com
gabriel.pireda@edumobi.com.br
thiagoguimaraes.dev@gmail.com
barreto@edumobi.com.br
Quais os tipos de teste são aplicados nos projetos do seu dia a dia.
14 respostas
Testes unitários
9 (64,3%)
Testes de integração
3 (21,4%)
Testes de ponta
0 (0%)
Testes manuais e exploratórios
11 (78,6%)
Testes Automatizados
1 (7,1%)
São só teste do desenvolvedor e do QA
1 (7,1%)
0,0
2,5
5,0
7,5
10,0
12,5
É possível aplicar estes testes na cultura Agile?
14 respostas
Sim
Não
100%
O cliente é envolvido no processo de decisão do que vai ser desenvolvido?
14 respostas
Sim
Não
Algumas vezes
As vezes
21,4%
7,1%
64,3%
A qualidade de código é considerada durantes seus sprints?
14 respostas
Sim
Não
28,6%
71,4%
O que fere a qualidade de software na sua concepção?
12 respostas
Escopos estimados e negociados de forma errada.
Falta de planejamento; Falta de qualidade no código; Sprints curtos sem automacao.
Quando é realizado alterações ou ajustes sem consultar toda a equipe envolvida no projeto.
Requisitos mal definidos, minimizar a importância do teste devido ao tempo de teste, ausência de testes unitários manuais ou automatizados
Alterações em prod, sem passar por QA
Falta de uma cultura de qualidade por parte do desenvolvedor que costuma usar 'prazo' como desculpa para entregar o código sem qualidade.
Não possuir testes únitários e e2e automatizados (javascript).
São vários fatores. Um que creio ser bastante vivenciado diariamente é a mudança de regras quebrando coisas inesperadas (que seria bastante amenizado com a existência de testes unitários e de integração com uma cobertura razoável).
Falta de comunicação
entrega sem mínima preocupação com produto como um todo