

UNIVERSIDADE DO VALE DO RIO DOS SINOS – UNISINOS
UNIDADE ACADÊMICA DE EDUCAÇÃO CONTINUADA
ESPECIALIZAÇÃO EM QUALIDADE DE SOFTWARE

Tamara Possebon Noello

MELHORIA DO PROCESSO DE TESTE EM UMA EMPRESA DE
DESENVOLVIMENTO DE SOFTWARE

Porto Alegre

2016

UNIVERSIDADE DO VALE DO RIO DOS SINOS – UNISINOS
UNIDADE ACADÊMICA DE EDUCAÇÃO CONTINUADA
ESPECIALIZAÇÃO EM QUALIDADE DE SOFTWARE

Tamara Possebon Noello

MELHORIA DO PROCESSO DE TESTE EM UMA EMPRESA DE
DESENVOLVIMENTO DE SOFTWARE

Trabalho de Conclusão de Curso apresentado como requisito parcial para a obtenção do título de Especialista em Qualidade de Software, pelo curso de Pós-Graduação Lato Sensu em Qualidade de Software da Universidade do Vale do Rio dos Sinos – UNISINOS.

Orientador: Prof. Dra. Margrit Reni Krug

Porto Alegre

2016

Melhoria do Processo de Teste em uma Empresa de Desenvolvimento de Software

Tamara Possebon Noello

Unidade Acadêmica de Educação Continuada – Universidade do Vale do Rio dos Sinos
(Unisinos)
Caixa Postal 15.064 – 91.501-970 – Porto Alegre – RS – Brasil
noellotamara@gmail.com

Abstract. *Nowadays, software quality becomes a determining factor in the market, and software testing is an important step to achieve it. The process improvement also comes as an ally, as only tests can not ensure the achievement of quality. To improve a process is necessary to perform its mapping. In this context, this paper intends to perform the improvement of the testing process in a small software development company. The article comprises the proposed improvement of the testing process as well as the evaluation results, after the application of the improved process, obtained by means of a qualitative questionnaire applied to the ones involved. The results indicate that after using the proposed process, an improvement in the tests was achieved.*

Resumo. *Atualmente, a qualidade de software torna-se fator determinante no mercado, e o teste de software é um passo importante para alcançá-la. A melhoria de processos também surge como aliada, visto que somente os testes, não podem assegurar a obtenção da qualidade. Para que um processo possa ser melhorado é necessário realizar o seu mapeamento. Nesse contexto, o presente trabalho pretende realizar a melhoria do processo de teste em uma empresa de desenvolvimento de software, de pequeno porte. O artigo engloba a proposta da melhoria do processo de teste, bem como os resultados da avaliação, após a aplicação do processo melhorado na empresa, obtidos por meio de um questionário qualitativo empregado aos envolvidos. Os resultados indicam que após a utilização do processo proposto, uma melhoria na execução dos testes foi alcançada.*

1. Introdução

No desenvolvimento de software a qualidade torna-se um fator determinante para manter a competitividade no mercado. Em geral, pode-se dizer que um produto é de qualidade quando atende aos requisitos especificados (CAMPOS, 2016). Segundo Pressman (2011, p. 360): “Um software de alta qualidade gera benefícios para a empresa de software bem como para a comunidade de usuários finais”.

Pressman (2011, p. 370) define a qualidade de software como o gerenciamento adequado da qualidade, sendo capaz de produzir software de valor para os desenvolvedores e usuários. “Ela é o resultado de um bom gerenciamento de projeto e uma prática consistente de engenharia de software.” Para Duarte e Falbo (2000) a qualidade de um produto não é alcançada espontaneamente, ela deve ser implantada.

Sendo assim, pode-se dizer que a qualidade de software está diretamente ligada a qualidade do processo de desenvolvimento.

Uma das principais formas de se conquistar a qualidade do produto é através do processo de teste de software. A utilização de técnicas para o gerenciamento da qualidade, atribuídas a tecnologias de softwares atuais, juntamente com a evolução nos testes, ocasionam uma melhora no nível de qualidade do produto (PRESSMAN, 2011).

Um processo pode ser definido como um conjunto de atividades capazes de produzir um produto, um serviço ou uma informação. (ALVARENGA et al., 2013). Para Mello e Salgado (2005, p. 1), “o processo é um conceito fundamental no projeto dos meios pelos quais uma empresa pretende produzir e entregar seus produtos e serviços aos seus clientes”.

O teste de software, por sua vez, contempla atividades que devem ser planejadas e executadas em conjunto e com antecedência, e possui dois objetivos específicos que são: certificar que o software atende aos requisitos especificados e encontrar inconformidades no sistema (PRESSMAN, 2011). Segundo Sommerville (2011, p. 144): “O teste é destinado a mostrar que um programa faz o que é proposto a fazer e para descobrir os defeitos do programa antes do uso”. Por outro lado, Pfleeger (2007, p. 314) afirma que: “Os procedimentos de teste devem ser amplos o suficiente para exercitar as funções do sistema a fim de satisfazer a todos: usuários, clientes e desenvolvedores. Se os testes não forem completos, os defeitos podem permanecer sem serem detectados”.

Cada vez mais, a metodologia ágil está presente no processo das empresas. Ela faz com que o teste fique mais próximo do desenvolvimento, pois a equipe deve se responsabilizar pela programação, testes e documentação das tarefas em andamento. A proposta desse modelo é que os projetos sejam conduzidos de um modo adaptável a sua realidade. Para isso utiliza-se o desenvolvimento iterativo, com iterações curtas, e ao fim de cada uma delas, disponibiliza-se uma versão completa e pronta para ser utilizada (OLIVEIRA, 2003).

De forma isolada, o teste de software não pode garantir a qualidade do produto. Nesse sentido, a melhoria de processos torna-se uma ação básica nas empresas, além de ajudar na diminuição dos custos (PAIM et al., 2011). Donner et al. (2001), consideram que o mapeamento do processo faz-se necessário para que seja possível melhorá-lo. “O mapeamento de processos é uma ferramenta de auxílio visual para retratar relações de processos de trabalho, ilustrando suas entradas, saídas e atividades” (ALVARENGA et al., 2013, p. 5). Mello e Salgado (2005), afirmam que o mapeamento detalha todas as fases do processo, possibilitando a visualização e o entendimento para todos os envolvidos. As técnicas utilizadas viabilizam a identificação de falhas e oportunidades de melhorias, além de apontar quais são as etapas mais críticas no processo.

Em virtude disso, este artigo relata o projeto, que teve como objetivo principal, propor melhorias no processo de teste de uma empresa de desenvolvimento de software, de pequeno porte, através da observação e análise do processo até então utilizado para propor as alterações. Para atingir o objetivo principal, definiram-se os seguintes objetivos específicos:

- a) Mapeamento e análise do processo de teste atualmente utilizado;

- b) Levantamento de pontos positivos e negativos do atual processo;
- c) Identificação de melhorias a serem aplicadas ao processo;
- d) Proposta do processo de teste melhorado;
- e) Aplicação do processo proposto no projeto em andamento, e;
- f) Avaliação do processo melhorado junto aos colaboradores da empresa.

Cabe ressaltar que, as melhorias foram sugeridas baseando-se nos levantamentos bibliográficos realizados, e a adaptação do processo proposto à realidade da empresa foi verificada com a equipe de desenvolvimento em conjunto com o seu gestor. A avaliação do processo de teste proposto foi realizada através da aplicação de um questionário, o qual foi respondido por colaboradores envolvidos nas atividades do processo na empresa analisada.

Este trabalho está constituído da seguinte forma: A seção 2 exibe uma visão geral sobre a qualidade de software, incluindo a melhoria e o mapeamento de processos, bem como testes de software. Na seção 3 apresentam-se os trabalhos relacionados. Em seguida, na seção 4 aborda-se a metodologia aplicada no presente trabalho. As seções 5 e 6 expõem, respectivamente, a situação atual da empresa, a aplicação do processo de teste proposto e o seu resultado. Por fim, a seção 7 é responsável pela conclusão.

2. Referencial Teórico

O referencial teórico utilizado para elaboração e execução da pesquisa realizada neste trabalho é apresentado nesta seção.

2.1. Qualidade de Software

No contexto do desenvolvimento de software, compreende-se que a qualidade deve ser algo que atenda às exigências dos seus usuários, e que satisfaça uma série de características (DUARTE e FALBO, 2000). As principais delas, definidas pela ISO 9126, são: funcionalidade, confiabilidade, usabilidade, eficiência, facilidade de manutenção e portabilidade (PRESSMAN, 2011).

Pressman (2011) afirma que a qualidade de software é consequência do gerenciamento eficaz do projeto, aliado à utilização de práticas da engenharia de software. A administração correta da qualidade torna possível a criação de um produto útil, sem erros e que atenda às expectativas do cliente.

A garantia da qualidade serve como forma de auxílio na obtenção da qualidade de software, pois ela estabelece mecanismos, processos e padrões. “O processo de gerenciamento de qualidade verifica os entregáveis de projeto para garantir que eles sejam consistentes com os padrões e objetivos organizacionais”. É importante que as empresas instaurem uma cultura de qualidade, e que os colaboradores considerem isso como um ponto positivo (SOMMERVILLE, 2011).

Para Sommerville (2011, p. 458), a qualidade do software e a qualidade do processo de desenvolvimento são relativas. O processo de desenvolvimento utilizado tem influência na qualidade do software entregue, e “bons processos são mais suscetíveis de conduzir o software de boa qualidade”. Pressman (2011) reitera que a melhoria do processo agrega indicadores de qualidade, que devem proporcionar a

qualidade no processo, e resultar em uma maior qualidade no produto. Entretanto, segundo Sommerville (2011, p. 497), “a melhoria de processos é uma atividade de longo prazo, pois cada uma das fases do processo de melhoria pode durar vários meses”.

2.2. Melhoria de Processos

“A melhoria de processos implica a compreensão dos processos existentes e sua mudança para aumentar a qualidade de produtos e/ou reduzir custos e o tempo de desenvolvimento” (SOMMERVILLE, 2011, p. 493).

Primeiramente, é fundamental identificar o processo em que há necessidade de melhoria, e definir de forma pontual, as oportunidades e/ou expectativas de melhoria (VALLE e OLIVEIRA, 2009). Para que um processo possa ser melhorado é necessário detalhar como ele está atualmente, incluindo seus pontos fortes e pontos fracos. Inicialmente o processo deve ser mapeado, e em seguida, remodelado com o propósito de eliminar as atividades que não acrescentam valor. Dessa forma é possível melhorar a eficiência do processo, bem como o prazo de entrega do produto (DONNER et al., 2001). Todas as atividades do processo devem trabalhar juntas e estar alinhadas em um propósito em comum (HAMMER, 2002).

2.3. Mapeamento de Processos

Conforme Villela (2000), o mapeamento de processos é uma ferramenta que auxilia na melhoria dos processos existentes ou na implantação de uma nova estrutura de processos na organização. Proporciona também, um melhor entendimento dos processos atuais, viabilizando a eliminação ou melhoria dos processos que demandam mudanças.

O mapeamento deve apresentar a sequência de atividades que são realizadas em determinado processo de forma gráfica, através de fluxograma. A utilização do fluxograma proporciona uma ampla visualização do processo, além de tornar mais fácil a participação dos envolvidos. O desenho do processo deve descrever com exatidão quais são as relações nas áreas funcionais da organização (SCARTEZINI, 2009). Devem-se representar as atividades, operações básicas, eventos de início e fim, decisões que devem ser tomadas, pessoas e seus papéis e responsabilidades, bem como definir qual será o nível de detalhamento do processo que está sendo mapeado. Existem diversas técnicas para se mapear um processo (VALLE e OLIVEIRA, 2009).

“Os processos organizacionais sem controle têm a tendência natural de se deteriorar progressivamente, gerando como efeito, serviços de qualidade cada vez pior”. Portanto, é pertinente identificar problemas no processo e determinar as suas causas, tornando possível a correção ou prevenção dos mesmos. (SCARTEZINI, 2009, p. 20). Existem diversas maneiras de se identificar os problemas. Monitorar o processo é uma delas, para isso é interessante à definição de indicadores de desempenho¹. Vaz (2008) sugere a realização de revisões nas atividades do processo. Valle e Oliveira (2009)

¹ Indicadores de desempenho são “formas de representações quantificáveis das características de um processo e de seus produtos ou serviços” (SCARTEZINI, 2009)

recomendam, como forma de colaboração na otimização do processo, colher as sugestões dos envolvidos.

Após a descoberta dos problemas no processo é necessário corrigi-los ou eliminá-los para que o mesmo possa ser melhorado. Scartezini (2009) sugere algumas alternativas, desde a aplicação de ações corretivas ou preventivas até a elaboração de um plano de ação. Valle e Oliveira (2009) citam algumas práticas que objetivam o aprimoramento do processo: evitar repetições das atividades, agrupar atividades que se complementam e eliminar gargalos. Mansar e Reijers (2007) descrevem melhores práticas para serem seguidas ao redesenhar um processo. Algumas são: eliminar tarefas desnecessárias, aplicar novas tecnologias, delegar poder de decisão aos colaboradores, realocar tarefas corretamente, considerar tarefas que possam ser executadas em paralelo, minimizar a quantidade de áreas, grupos e pessoas envolvidas em um projeto.

2.4. Testes de Software

O controle de qualidade incorpora diversas atividades, e uma delas é o teste de software. Seu principal objetivo é a descoberta de erros, e para isso é necessário planejá-los e conduzi-los de forma eficiente (PRESSMAN, 2011).

Para Myers (2012), o software deve ser previsível e consistente, não apresentando surpresas para os usuários. O teste é um processo, ou uma série de processos, projetados para ter certeza que o sistema faz o que se propõe a fazer, e por outro lado, não faz o que não foi planejado.

O teste de software refere-se à execução de um produto de software com a finalidade de verificar se as suas especificações foram atendidas, bem como determinar se o mesmo está funcionando corretamente (NETO, 2010).

Conforme Pfleeger (2007), os testes devem ser completos, ou então, os defeitos podem não serem identificados. “Um bom teste tem alta probabilidade de encontrar um erro. Para atingir esse objetivo, o testador deve entender o software e tentar desenvolver uma imagem mental de como o software pode falhar” (PRESSMAN, 2011, p. 430).

Gouveia (2004) afirma que o processo de teste deve ser iniciado cedo, pois caso os defeitos sejam encontrados tardiamente, os mesmos tornam-se custosos e mais difíceis de serem corrigidos. Myers (2012) destaca alguns princípios vitais para o teste, que podem ser verificados na Tabela 1.

Tabela 1. Princípios Vitais para Testes de Software

Número do Princípio	Princípio
1	Uma parte importante de um caso de teste é a definição de uma saída ou resultado esperado.
2	Um programador não deve tentar testar o seu próprio programa.
3	A equipe de programadores não deve testar os seus programas.
4	Qualquer processo de teste deve incluir uma inspeção minuciosa dos resultados de cada teste
5	Casos de teste devem ser escritos para condições de entrada inválidas e

	inesperadas, assim como para as válidas e esperadas.
6	Inspecionar um programa para ver se ele faz o que é proposto a fazer é apenas metade da batalha; a outra metade é ver se o programa não faz o que não é proposto a fazer.
7	Evite casos de teste descartáveis, ao menos que o sistema seja realmente descartável.
8	Não estimar um esforço de teste menor supondo que erros não serão encontrados.
9	A probabilidade de existir erros em uma parte do sistema é proporcional ao número de erros já encontrados nessa parte.
10	O teste é uma tarefa extremamente criativa e intelectualmente desafiadora.

Fonte: Adaptado de Myers (2012, p. 13)

Para Vicente (2010) além de planejar os testes, devem-se realizar os seguintes passos: Preparar os casos de teste, organizar os dados que serão utilizados nos testes, executar os casos de teste e avaliar os testes confrontando os resultados esperados com os reais. Na Figura 1, pode-se visualizar um modelo de processo de teste, que faz uso dos passos citados anteriormente.

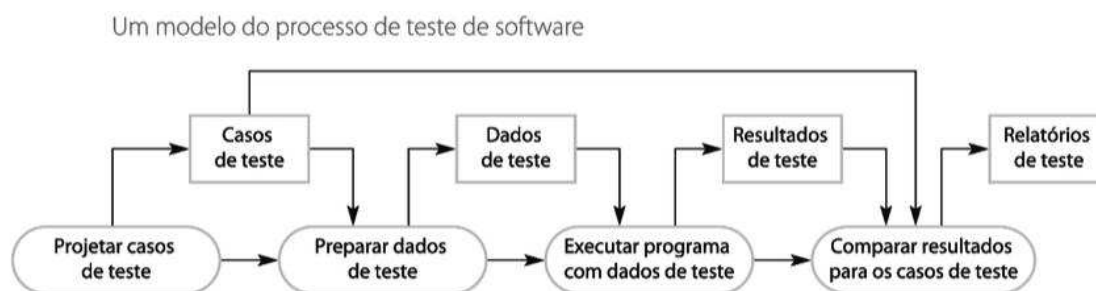


Figura 1. Modelo de Processo de Teste

Fonte: Sommerville (2011, p. 147)

Neto (2010) cita alguns elementos auxiliares na execução do teste:

- **Caso de teste:** Relata uma situação que será testada. Possui um conjunto de entrada, exceções da execução e resultados esperados.
- **Procedimento de teste:** São os passos necessários para que um caso de teste ou um conjunto de casos de teste sejam executados.
- **Critério de teste:** Auxiliam na seleção e avaliação de casos de testes.

Há duas técnicas para realização dos testes, a funcional e a estrutural. Vicente (2010) afirma que no teste funcional os casos de teste são oriundos da especificação dos requisitos. Para Gouveia (2004), testes funcionais também são conhecidos como teste de caixa-preta, pois para a escrita dos casos de teste não é necessário saber como o software foi implementado.

Diferente do teste funcional, o estrutural se baseia na estrutura do programa para escrita dos seus casos de teste. Devido a isso, o mesmo também pode ser chamado de teste caixa-branca (VICENTE, 2010).

Os testes devem ser incluídos nos diferentes níveis do desenvolvimento, bem como serem adequadamente planejados, em cada uma das fases, conforme o desenvolvimento do produto (NETO, 2010). Na Figura 2, é possível verificar qual o nível de teste se enquadra em cada etapa de desenvolvimento do software.

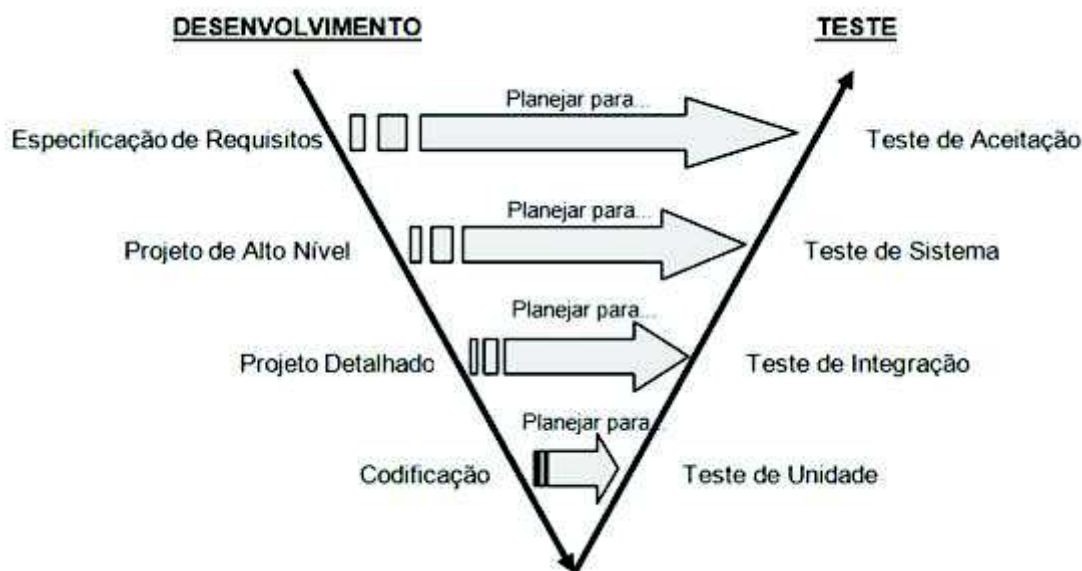


Figura 2. Níveis de Desenvolvimento e de Teste

Fonte: Neto (2010, p. 56)

Pode-se observar, através da Figura 2, que o teste deve ser executado ao longo da evolução do desenvolvimento do software, assim como, deve ser planejado em todas as fases do projeto.

2.4.1. Teste Unitário

“O teste unitário é o processo de testar os componentes de programa, como métodos ou classes de objeto” (SOMMERVILLE, 2011 p. 148). Seus principais alvos são os métodos de objetos, e trechos de código (NETO, 2010). Possui como objetivo encontrar defeitos na lógica e na implementação de cada unidade, apresentando evidências de que a mesma funciona corretamente de forma isolada. Como cada unidade é testada separadamente, e os testes podem ser executados conforme o código é finalizado, geralmente é o próprio desenvolvedor quem escreve os testes (VICENTE, 2010).

2.4.2. Teste de Integração

O teste de integração pode ser definido como “o modo que o teste é conduzido para integrar componentes em um sistema” (GOUVEIA, 2004 p. 14). Esse tipo de teste verifica se todas as partes do programa funcionam corretamente ao serem integradas. Deve ser realizado após o teste unitário, dando destaque na construção da estrutura do software (VICENTE, 2010).

2.4.3. Teste de Sistema

O teste de sistema pode ser visualizado como diversos testes, possuindo como objetivo exercitar o sistema. “Embora cada um dos testes tenha uma finalidade diferente, todos

funcionam no sentido de verificar se os elementos do sistema foram integrados adequadamente e executam as funções a eles alocadas” (PRESSMAN, 2011, p. 419). Os testes devem ser executados simulando o uso do software por um usuário final, ou seja, “nos mesmos ambientes, com as mesmas condições e com os mesmos dados de entrada que um usuário utilizaria no seu dia-a-dia de manipulação do software” (NETO, 2010, p. 56).

2.4.4. Teste de Aceitação

O teste de aceitação é realizado pelos usuários do software, onde são simuladas operações de rotina do sistema, com o objetivo de analisar se o mesmo se comporta conforme o que foi acordado (NETO, 2010).

2.4.5. Teste de Regressão

As alterações no software podem ocasionar problemas nas funcionalidades já existentes, ou seja, novos defeitos podem ser introduzidos. Devido a isso, um conjunto de testes deve ser executado para validar que as modificações realizadas, bem como as demais funcionalidades, estejam funcionando corretamente (VICENTE, 2010).

O teste de regressão é uma estratégia de teste que pode ser adotada em qualquer nível, e “consiste em se aplicar, a cada nova versão do software, ou a cada ciclo, todos os testes que já foram aplicados nas versões ou ciclos de teste anteriores do sistema” (NETO, 2010, p. 56).

2.5. Testes de Software em Métodos Ágeis

Diferente da metodologia tradicional, a metodologia ágil leva em consideração que os sistemas são modificados durante o seu desenvolvimento, devido ao fato que durante esse processo acaba-se compreendendo melhor qual é o problema a ser resolvido. “Nas metodologias ágeis, os processos são orientados para adaptar o sistema a mudanças durante todo o tempo”. Os projetos são conduzidos de forma adaptativa, por meio do desenvolvimento iterativo. Na maioria dos casos as iterações devem ser curtas, e ao final, um software com novas funcionalidades, pronto para ser utilizado, deve ser entregue ao cliente (OLIVEIRA, 2003, p. 13).

Como os métodos ágeis ocorrem em ciclos de desenvolvimento curtos, os testes devem ser frequentes, e os defeitos identificados o quanto antes (VICENTE, 2010). A Figura 3 faz um comparativo entre os testes tradicionais e os testes ágeis.

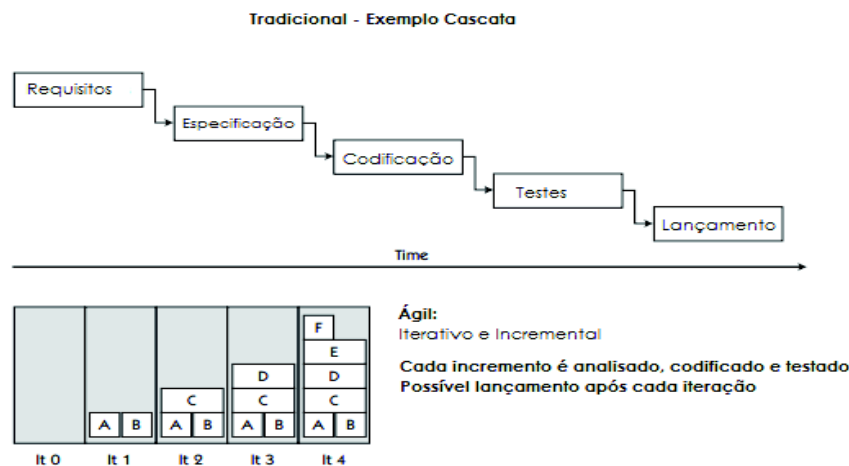


Figura 3. Testes Tradicionais X Testes Ágeis

Fonte: Adaptado de Crispin e Gregory (2009, p. 13)

Como se pode verificar no diagrama da Figura 3, na abordagem tradicional o teste é realizado apenas no final do processo e antes do seu lançamento. Por sua vez, o teste ágil é iterativo e incremental, e os testadores testam cada incremento assim que eles são implementados. A equipe desenvolve e testa uma parte do código até ter certeza que está funcionando corretamente, e em seguida parte para a próxima parte que precisa ser construída (CRISPIN E GREGORY, 2009).

No contexto da agilidade, todos os membros da equipe são responsáveis pela qualidade, por isso os testadores devem fazer parte do time de desenvolvimento (VICENTE, 2010). Para Crispin e Gregory (2009), todos os integrantes de uma equipe ágil estão focados em fornecer um produto de alta qualidade e que fornece valor ao negócio. Os testadores ágeis devem fazer mais do que apenas executar tarefas de teste, eles trabalham para garantir que o seu time ofereça a qualidade que seus clientes precisam.

Habib (2016) afirma que testes baseados em riscos, testes de regressão automatizados e testes reativos funcionam bem com métodos ágeis. Os testes baseados em riscos definem uma prioridade para os riscos, e realizam os testes conforme essa prioridade, o que garante que as funcionalidades mais importantes serão testadas antes das demais, tendo mais tempo para a correção dos defeitos encontrados. Como há modificações frequentes, os testes de regressão automatizados tornam-se cruciais, e os testes reativos possibilitam explorar diversos aspectos no sistema.

Uma abordagem de teste ágil, oriunda da metodologia XP (*Extreme Programming*), é o desenvolvimento dirigido a testes, ou TDD (*Test-Driven Development*). Nessa abordagem o desenvolvimento é acompanhado por um teste. O desenvolvedor escreve os testes, realiza a codificação e aplica os testes escritos anteriormente (SOMMERVILLE, 2011).

Para Vicente (2010, p. 45), no processo ágil de desenvolvimento de software, os testes atribuem papéis importantes. “O teste apoia a comunicação entre desenvolvedores e clientes, fornece *feedback* sobre quais as funcionalidades do sistema estão funcionando e apoia a manutenção do sistema, pois as mudanças tendem a ser mais seguras se a equipe tiver um bom conjunto de testes para o sistema.”

Crispin e Gregory (2009) citam alguns princípios considerados importantes para um testador ágil, que são: Fornecer *feedback* contínuo, entregar valor para o cliente, permitir a comunicação face-a-face, ter coragem, manter o processo simples, praticar a melhoria contínua, responder as mudanças, se auto organizar, concentrar-se em pessoas e se divertir.

3. Trabalhos Relacionados

Nesta seção são apresentados alguns trabalhos relacionados com a melhoria de processo de teste de software.

Crespo et al. (2004), desenvolveram uma metodologia para implantar ou melhorar o processo de teste, nas empresas desenvolvedoras de software, baseada na Norma IEEE 829-1998, que foi validada através da aplicação em uma micro empresa de desenvolvimento. Camargo (2012), em sua dissertação, por meio de um estudo prático, apresentou uma estratégia para definir um processo de teste ou melhorar o processo existente nas pequenas empresas. Sartori (2005) propôs um modelo para avaliação e implantação de melhorias na área de teste de software em micro e pequenas empresas, que se chama Melhoria do Processo de Teste para Pequenas Empresas.

Com base nos modelos de maturidade e normas internacionais, Silva (2015), desenvolveu em seu trabalho, um modelo de maturidade, e uma abordagem de implementação de um processo de teste, que considera as limitações das micro e pequenas empresas de desenvolvimento de software. Pretz e Bertolini (2016) elaboraram uma proposta, se baseando na metodologia TDD, para o processo de teste em uma empresa de pequeno porte.

No contexto da melhoria de processos, pode-se citar a dissertação de Anacleto (2004), que descreve um método e um modelo de avaliação de processos, adaptados da ISO/IEC 15504, para que as empresas possam iniciar um programa de melhoria voltado para as metas de negócio da organização.

Após comparar a proposta do presente trabalho com os trabalhos correlatos, pode-se observar que há uma semelhança com relação à aplicação de melhorias nos processos de teste já existentes em algumas empresas de desenvolvimento de software. Na maioria das pesquisas relacionadas foram apresentados modelos baseados em normas específicas de qualidade, para implantação e em alguns casos avaliação, do processo de teste nessas empresas. Em contrapartida, na pesquisa desenvolvida nesse trabalho, realizou-se a proposta de um processo de teste melhorado, conforme a realidade de uma empresa específica, seguindo as boas práticas citadas na literatura.

4. Metodologia

Nesta seção, aborda-se a metodologia adotada no presente trabalho. A classificação da pesquisa, os métodos de trabalho e as etapas executadas durante o desenvolvimento do projeto, são apresentados a seguir.

4.1. Classificação da Pesquisa

Para elaboração deste trabalho, utilizou-se a pesquisa qualitativa de natureza aplicada, pois conforme Cresswell (2010, p. 206): “A investigação qualitativa emprega diferentes

concepções filosóficas; estratégias de investigação; e métodos de coleta, análise e interpretação dos dados”. Já Klein et al. (2015) consideram que esse paradigma ressalta a criação do mundo pela consciência humana, portanto, os indivíduos apresentam uma importância na geração do conhecimento. Não é necessária a busca de indícios que justifiquem determinadas suspeitas antes de iniciar a pesquisa, pois elas se consolidam durante a execução da mesma (LUDKE e ANDRÉ, 2013). Para Azevedo et al. (2011), o objetivo dessa pesquisa são os cenários complexos e específicos.

Gil (2010, p. 27) afirma que a pesquisa aplicada engloba análises que propõem a resolução dos problemas descobertos no ambiente do pesquisador. São “pesquisas voltadas à aquisição de conhecimentos com vistas à aplicação numa situação específica”. Portanto, do ponto de vista do objetivo da pesquisa pode-se dizer que a mesma é exploratória. Seu propósito é viabilizar melhores informações sobre o assunto, auxiliar na determinação do tema para o trabalho, precisar quais os objetivos ou hipóteses da pesquisa, e até mesmo encontrar um novo enfoque para o trabalho (ANDRADE, 2010). A pesquisa exploratória é “um tipo de pesquisa que permite a obtenção de novos conhecimentos, bem como ampliação e complementação acerca do tema abordado” (AZEVEDO, 2011, p. 41).

Para auxiliar no estudo e na análise da literatura, utilizou-se a pesquisa bibliográfica. “A pesquisa bibliográfica é elaborada com base em material já publicado” (GIL, 2010, p. 29). Ela proporciona aos pesquisadores o contato com os documentos que tratam do tema que está em estudo (SILVA et al., 2009). Azevedo (2011, p. 43) reitera que a pesquisa bibliográfica “permite ao pesquisador utilizar-se de uma série de recursos disponíveis sobre um determinado tema de estudo. Para pesquisar o fenômeno, fato ou evento propriamente dito, o pesquisador recorre a pesquisas já realizadas, e a materiais já publicados”.

Adotou-se como metodologia de pesquisa a pesquisa-ação e o método de observador participante. A pesquisa-ação possibilita para um participante do processo, que apresenta o desejo de melhorar uma situação, a realização da pesquisa. Permitindo assim, a execução de modificações necessárias durante a execução do estudo (ENGEL, 2000). Segundo Tripp (2005), a mesma realiza o planejamento, implementação, desenvolvimento e avaliação de uma mudança, e é utilizada como forma de melhorar determinada atividade, tornando possível obter um bom nível de entendimento do processo durante a sua implementação. Para Shaughnessy (2012), o método observador participante permite que o observador participe das atividades que estão sendo examinadas. Angrosino (2011) reitera que o pesquisador junta-se à comunidade em questão. “Como os observadores participantes têm as mesmas experiências que as pessoas sob estudo, eles podem obter os *insights* e visões importantes” (SHAUGHNESSY, 2012 p. 114).

4.2. Métodos de Trabalho

A execução do trabalho teve como princípio o mapeamento das atividades do processo de teste utilizado em uma empresa de desenvolvimento de software, de pequeno porte. Os passos necessários para o desenvolvimento deste trabalho podem ser visualizados na Figura 4.

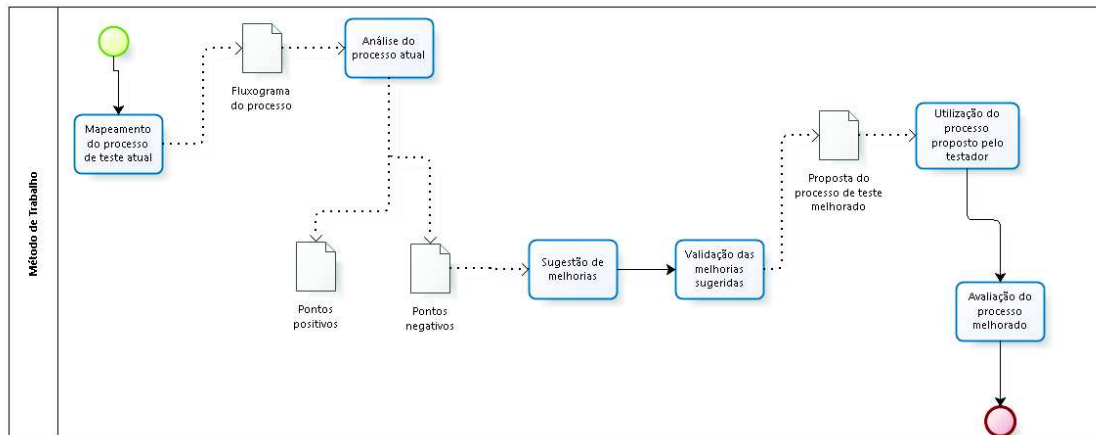


Figura 4. Passo a passo utilizado no desenvolvimento do trabalho

Fonte: Elaborado pela autora

Conforme se pode verificar na Figura 4, inicialmente realizou-se o mapeamento do processo de teste utilizado pela empresa. As etapas foram mapeadas através da experiência do testador, que as relatou com detalhes. Para análise, adotou-se como base a literatura, mais especificamente na área de testes de software, com a finalidade de realizar uma comparação entre as atividades do processo. Baseado nessa análise foi possível chegar aos pontos positivos e negativos do mesmo.

A partir dos pontos negativos encontrados, e com base na literatura, possíveis melhorias foram elencadas. O novo processo de teste foi proposto por meio da validação dessas melhorias com os membros do time de desenvolvimento, em conjunto com o *Product Owner*. Antes de ser avaliado, o processo foi aplicado na empresa. O testador foi responsável pela sua utilização, que ocorreu em um período de 30 dias, o equivalente a duas *Sprints*.

Na avaliação do processo proposto aplicou-se um questionário para coletar a opinião dos envolvidos no processo. As questões do questionário foram adaptadas da escala *Likert*.

Segundo Costa e Júnior (2014, p. 5), a escala *Likert* “consiste em tomar um construto e desenvolver um conjunto de afirmações relacionadas à sua definição, para as quais os respondentes emitirão seu grau de concordância.” A facilidade ao manusear as respostas é uma vantagem dessa escala, devido ao fato de que a seleção do grau de concordância é uma tarefa simples ao respondente. Como se pode perceber, através da Tabela 2, as respostas recebem uma pontuação, que servem para ponderar a reação do pesquisado com relação à pergunta (SANCHES et al., 2011).

Tabela 2. Grau de Concordância das Respostas

Grau de Concordância	Pontuação
Discordo Totalmente	1
Discordo Parcialmente	2
Não Concordo Nem Discordo	3
Concordo Parcialmente	4
Concordo Totalmente	5

Fonte: Adaptado de Costa e Júnior (2014, p. 5)

No questionário aplicado, ao invés de utilizar o grau de concordância proposto, as respostas foram adaptadas para “Muito Bom”, “Bom”, “Regular”, “Ruim” e “Péssimo”, e também “Muito Importante”, “Importante”, “Pouco Importante” e “Não é Importante”. Esse questionário pode ser visualizado no APÊNDICE A.

Assim que o questionário foi respondido por todos os envolvidos, realizou-se a análise das respostas, as quais podem ser observadas no APÊNDICE A. Na pesquisa qualitativa, “a análise dos dados tende a seguir um processo indutivo”, uma vez que o pesquisador possui um contato direto com o ambiente e com o processo que está sendo avaliado (LUDKE e ANDRÉ 2013, p. 14).

5. Mapeamento do Processo Atual

Este capítulo tem como objetivo apresentar características que contextualizam a empresa estudada, além de apresentar o processo de teste atualmente utilizado, apontando-se os pontos positivos e os que requeriam melhorias, o que possibilitou a proposta da melhoria do processo de teste da empresa.

5.1. Empresa

Para a pesquisa qualitativa tomou-se como base uma empresa de desenvolvimento de software, de pequeno porte, fundada em abril de 2010 na cidade de Passo Fundo – Rio Grande do Sul.

A empresa desenvolve um sistema de gestão ERP (*Enterprise Resource Planning*), que atende a diversas empresas do ramo comercial, que atuam no varejo e atacado, e também empresas prestadoras de serviços. Ele foi construído com base nos modelos científicos de administração e contabilidade.

A sede da empresa está localizada em Passo Fundo – RS, porém, a mesma dispõe de consumidores presentes em outros estados do Brasil. Como forma de auxílio, e também para estar mais próxima desses clientes, há parceiros comerciais presentes em alguns desses estados.

O ERP deve ser altamente configurável para satisfazer as diferentes e variadas necessidades dos seus usuários, e também das suas áreas de atuação. A empresa executa um único projeto, o qual se destina a manutenção e ao desenvolvimento de novas funcionalidades para o sistema, uma vez que a instituição não trabalha com projetos

customizados. Dessa forma, o software deve estar apto a atender aos aspectos fiscais e gerenciais de diversos clientes.

A linguagem de programação utilizada pela empresa é Java, e o software desenvolvido roda em uma plataforma *web*. Há um servidor disponível para o ambiente de produção, o qual centraliza todas as instalações dos clientes.

Para o desenvolvimento do ERP, a empresa adota alguns princípios da metodologia ágil *Scrum*. Uma metodologia ágil e flexível, utilizada para estabelecer um processo de desenvolvimento através de um fluxo interativo e incremental, focado na equipe e com iterações curtas. O mesmo é “indicado e utilizado para o desenvolvimento de softwares em ambientes complexos, onde os requisitos mudam com certa frequência, sendo o caminho utilizado para aumentar produtividade nesses tipos de sistemas” (BISSI, 2007, p. 3).

O time de desenvolvimento é composto por dois desenvolvedores, onde um deles também realiza o papel de *Scrum Master*, e um testador. Esse último, desempenhado pela autora do trabalho. O *Product Owner* é representado pelos sócios da empresa, que exercem os cargos de gerente, geral e de desenvolvimento. Qualquer solicitação dos usuários ou alteração necessária no software deverá ser aprovada por pelo menos um deles.

Um dos propósitos da empresa é garantir a qualidade do sistema. Porém, durante as últimas 8 *Sprints*, equivalentes aos últimos 4 meses, foram relatados, aproximadamente, 20 defeitos pelos clientes no ambiente de produção. Também houve a geração de 2 versões de emergência, que não estavam previstas nas *Sprints*, para correção de defeitos considerados graves, por ocasionar travamento em determinadas funcionalidades. Essa situação gera transtorno para os usuários, e também para a equipe de desenvolvimento, pois transfere o foco do andamento da *Sprint*, para correção de erros encontrados em produção, e que não foram planejados.

5.2. Situação Atual

O processo se inicia com as solicitações dos clientes, que deverão ser aprovadas pelo *Product Owner* e transformadas em cartões de desenvolvimento. Para cadastro e organização desses cartões, a ferramenta MantisBT² foi adotada pela equipe. A partir disso, o *Scrum Master* organiza o quadro de *kanban*³ das *Sprints*. O prazo de entrega das demandas é apurado conforme a sua prioridade em conjunto com a análise das demais requisições que estão no *backlog*, e que já foram acordadas com os clientes.

A empresa utiliza um quadro de *kanban* virtual, o qual é proveniente de uma customização no software Mantis. O mesmo dispõe das seguintes colunas: “*To Do*”, “Validar”, “Em Validação”, “Validado”, “Retorno”, “Fazendo”, “Implementado”, “Inspeccionar”, “Verificando” e “Concluído”. Cada cartão possui uma categoria, e elas se distinguem no quadro através de cores. As categorias são: funcionalidade, melhoria, defeito, tarefa e testes. E suas cores, respectivamente: verde, azul, vermelho, amarelo e

² MantisBT é uma ferramenta *open source*, que auxilia no rastreamento dos defeitos. Mais informações disponíveis em: www.mantisbt.org

³ “O *kanban* é uma simbologia visual utilizada para registrar ações.” (SILVEIRA, 2016)

roxo. O quadro *kanban* pode ser observado na Figura 5, enquanto o fluxograma do processo é apresentado na Figura 6.

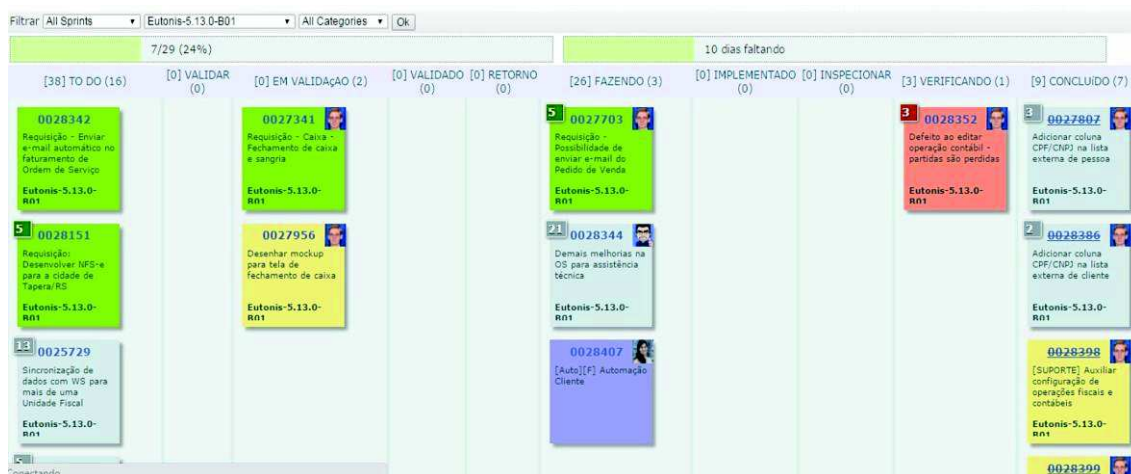


Figura 5. Quadro de *Kanban* virtual utilizado pela empresa

Fonte: Elaborado pela autora

Pode-se destacar como um ponto positivo no processo a utilização do *kanban* (Figura 5), o que facilita a visualização das tarefas por parte dos colaboradores da equipe e o controle do cronograma.

Os cartões seguem um fluxo no quadro de *kanban*, que pode ser visualizado na Figura 6. Inicialmente o *Product Owner* analisa a demanda do cliente, verificando se ela irá agregar valor ao software. Se a mesma for aprovada, o *Scrum Master* analisa a sua prioridade e o seu prazo de entrega, cadastra um cartão de desenvolvimento e aloca essa tarefa na coluna *TO DO* da *Sprint* em que ela será desenvolvida. Caso contrário, se a solicitação não for aceita, a mesma é descartada.

No início da *Sprint*, o desenvolvedor assume a tarefa arrastando-a para a coluna Fazendo. Se for necessária a aprovação do cliente, um protótipo é elaborado e o cartão vai para Validar. O responsável entra em contato com o cliente e o envia o protótipo. Nesse momento o cartão permanece Em Validação até ser aprovado. Com o consentimento do cliente, a tarefa é conduzida para Validado, e assim que o programador retoma o seu desenvolvimento, a mesma volta para Fazendo até que seja finalizada, e então o desenvolvedor leva o cartão para a coluna Implementado.

As tarefas saem de Implementado e vão para Inspeccionar cada vez que a versão de teste é atualizada. À medida que se iniciam os testes, os cartões são conduzidos para Inspeccionar. Se os testes não encontrarem defeitos ou inconformidades, o testador conclui a tarefa, levando-a até a coluna Concluído. No entanto, se o testador constatar que há inconformidades, o mesmo é retornado, e permanece no Retorno até que o programador faça os ajustes necessários. Nesse caso, as etapas referentes ao desenvolvimento e testes são praticadas novamente.

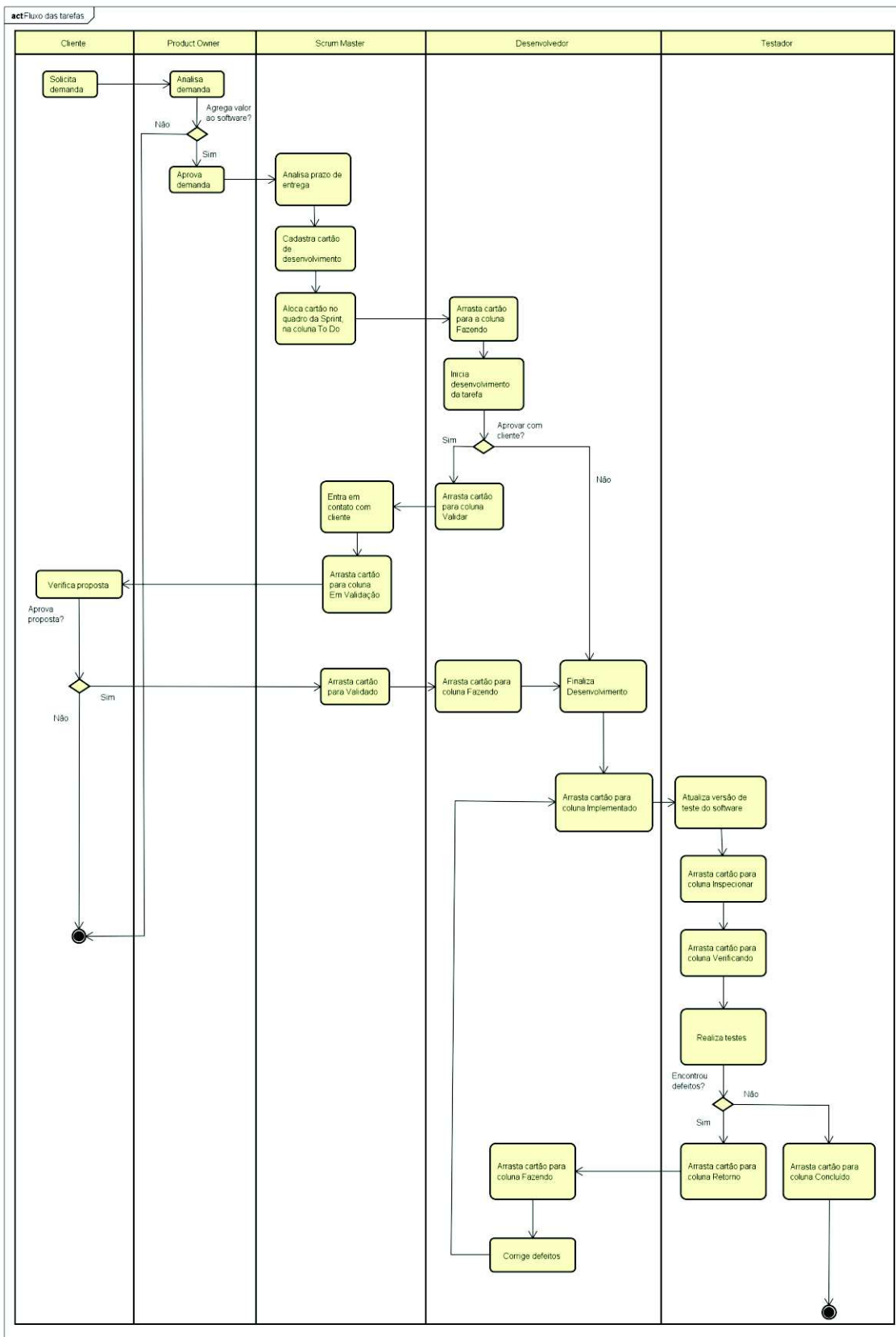


Figura 6. Fluxo do quadro *Kanban* usado pela empresa

Fonte: Elaborado pela autora

Como se pode notar, na Figura 6, a coluna *To Do* demonstra o que deve ser desenvolvido durante a *Sprint*. Validar, Em Validação e Validado são utilizadas quando é preciso a aprovação do cliente. Fazendo e Implementado distinguem a fase do desenvolvimento da tarefa. Inspeccionar e Verificando expõem, respectivamente, que os testes podem ser realizados, e estão sendo executados. A coluna Retorno é para identificar que foram encontradas inconformidades, e Concluído para caracterizar que o cartão foi finalizado com sucesso.

As principais atividades praticadas na empresa são: Avaliação das requisições, aprovação das solicitações dos clientes, desenvolvimento das tarefas, automação de teste e execução de testes manual. A realização dessas atividades, respectivamente, são responsabilidades do *Scrum Master*, *Product Owner*, desenvolvedor e testador. Ocorre também a estimativa dos cartões com a técnica *planning poker*⁴, e a liberação da versão entregue para os envolvidos.

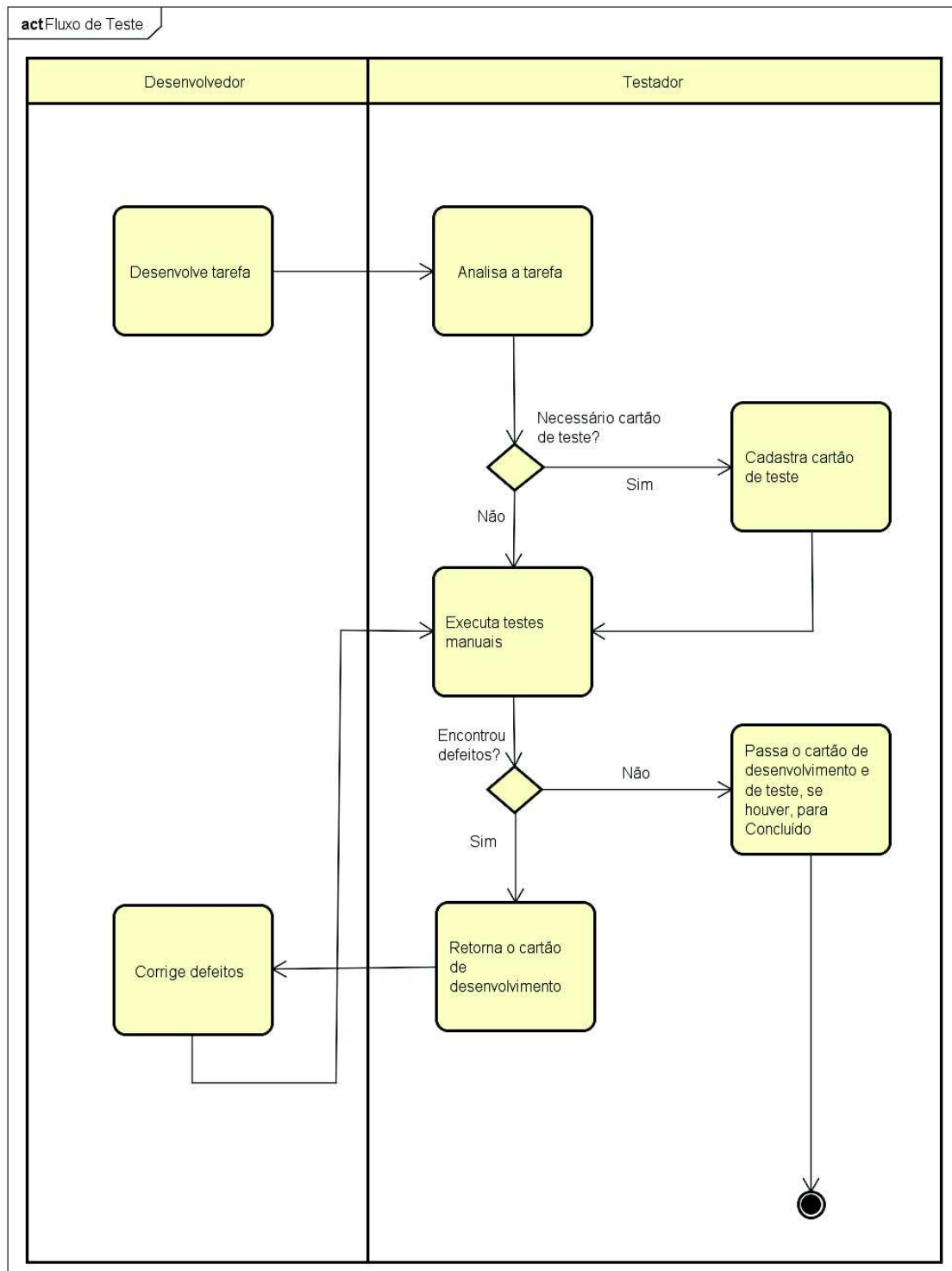
A empresa realiza apenas testes funcionais. Os principais elementos testados são as regras de negócio, validações em campos, *layout* e tela. Também são realizados testes exploratórios, com cenários positivos e negativos. A execução dos testes se dá de forma manual. Existem poucos cenários cobertos através da automação, e os mesmos são ampliados, em geral, no início da *Sprint*.

A Figura 7 mostra o mapeamento do processo de teste atualmente utilizado na empresa. Assim que o desenvolvimento de uma tarefa é finalizado, o testador realiza uma breve análise para verificar se é necessário cadastrar um cartão de teste. Geralmente, esses cartões são cadastrados quando a tarefa está relacionada a uma nova funcionalidade, ou é um defeito ou melhoria com impacto relevante no software. Os cartões de testes são separados pelo seu tipo, automatizado e manual, que servem para facilitar na diferenciação dos cenários que serão automatizados, e os testes realizados manualmente pelo testador.

Após a análise da tarefa ou relato do cartão de teste, os testes são iniciados. Os mesmos são efetuados manualmente pelo testador. Ao final da sua execução, se inconformidades forem encontradas, o cartão de desenvolvimento é retornado para que programador faça os ajustes necessários. Em seguida, os testes são novamente executados, para validar se as correções foram realizadas de forma correta.

Os cartões de desenvolvimento são concluídos apenas quando o testador constata, após a execução dos testes, que não há mais defeitos na tarefa. Se um cartão de teste foi cadastrado, o mesmo também é finalizado. Os demais defeitos descobertos, não oriundos das alterações que ocorreram na *Sprint*, são relatados na ferramenta Mantis, e notificados ao *Scrum Master*, que verifica a sua prioridade e determina uma *Sprint* para corrigi-los.

⁴ O *planning poker* é uma técnica de estimativa, que se baseia no consenso da equipe, através de um jogo de cartas. (CUNHA, 2009)



powered by Astah

Figura 7. Processo de teste utilizado atualmente pela empresa

Fonte: Elaborado pela autora

A partir da análise do fluxograma apresentado na Figura 7, constata-se que os testes são pontuais, visto que os mesmos são executados apenas para as alterações realizadas na *Sprint*. Por outro lado, pode-se observar que o testador finaliza o cartão de

desenvolvimento, apenas quando os testes demonstram que não foram encontradas inconformidades.

5.3. Pontos Positivos e Negativos do Processo de Teste Atual

Através do mapeamento e análise do processo, e com base na literatura, mais especificamente na área de teste de software e melhoria de processos, chegou-se aos pontos positivos e negativos do mesmo.

Destacou-se como pontos positivos:

- No quadro *kanban*, os cartões de teste possuem uma cor específica, assim como as demais categorias utilizadas pela equipe. Essa situação facilita a visualização no quadro, bem como possibilita aos envolvidos um melhor entendimento sobre o que está relacionado ao desenvolvimento, e aos testes, na *Sprint* em questão;
- Nos cartões de teste há a possibilidade de informar o seu tipo, se o mesmo é Manual ou Automatizado. Essa informação é importante para obter indicadores de esforço em cada tipo de teste;
- O testador é o responsável em concluir o cartão de desenvolvimento, e só o faz quando constata que não há mais defeitos na funcionalidade que foi alterada ou criada, e;
- Para os defeitos e/ou melhorias encontradas, não oriundas das modificações que estão em fase de teste, um novo caso é relatado na ferramenta Mantis. Dessa forma, os cartões não recebem retornos nessas ocasiões, apenas quando essas inconformidades são provenientes do seu desenvolvimento. Esses casos são corrigidos assim que possível, seguindo a sua prioridade, que é definida pelo *Scrum Master*.

Como pontos negativos podem-se considerar:

- Os cartões de teste não são cadastrados para todos os testes realizados na *Sprint*. E também, quem define se esses cartões serão cadastrados é o próprio testador. O que acaba ocultando o verdadeiro esforço dos testes;
- A realização de testes pontuais, apenas para as funcionalidades alteradas durante a *Sprint*, aliadas a não execução do teste de regressão. A não existência de um roteiro para execução de testes de regressão, juntamente com a execução de testes pontuais, implicam na qualidade da versão entregue, pois não é possível assegurar que o que estava funcionando anteriormente, continua funcionando nessa versão. Garante-se apenas, que as alterações realizadas estão corretas;
- Os testes automatizados já existentes não são executados. A automação de testes auxilia, principalmente nos testes de regressão, fazendo com que o testador tenha mais tempo para se dedicar a execução de testes manuais para cenários diversos. Cabe ressaltar, que essa situação também traz um custo para a empresa, pois houve um esforço para realizar a automação, e;
- A falta de indicadores. Através dos indicadores é possível realizar medições, e por consequência manter uma melhoria contínua no processo. Sommerville

(2011) afirma que através da medição, é possível avaliar se os métodos, ferramentas e processos estão sendo eficazes.

6. Proposta de Melhoria no Processo de Teste da Empresa

Este capítulo tem como objetivo apresentar as melhorias que foram evidenciadas, a partir dos pontos negativos levantados no processo utilizado atualmente pela empresa, bem como, detalhar a proposta do novo processo de teste baseado nessas melhorias, além da aplicação do processo melhorado na empresa, bem como os resultados obtidos após a sua avaliação.

6.1. Melhorias Sugeridas

As melhorias listadas a seguir foram sugeridas com base na literatura, e levando em consideração os pontos negativos do processo.

- No início da Sprint, após o quadro *kanban* ter sido definido pelo *Scrum Master*, os cartões de teste deverão ser cadastrados. Cada cartão de desenvolvimento deverá ser vinculado a um cartão de teste. A pontuação desses cartões será realizada juntamente com as tarefas da *Sprint*;
- Realizar o gerenciamento dos casos de teste, armazenando-os em um local de fácil acesso. Ao cadastrar os cartões de teste, caso não exista um caso de teste para a funcionalidade a ser testada, o mesmo deve ser escrito e anexado ao cartão;
- Planejar a automação de testes, definir as principais funcionalidades que devem ser automatizadas, bem como seus cenários. Criar os cartões para a escrita dos testes, alocando-o nas próximas *Sprints*, conforme a sua prioridade. Os casos de teste deverão ser anexados a esses cartões;
- Quando as funcionalidades que já foram automatizadas sofrerem alguma alteração, os testes deverão ser atualizados. Para isso, terá de ser cadastrado um cartão com as modificações necessárias, assim prevendo-o, se possível na *Sprint* corrente;
- Criação de um roteiro para execução de testes de regressão. Determinar quando os testes automatizados serão executados. Como a maioria dos cenários de teste ainda não estão automatizados, decidir quais serão os casos de testes que devem ser executados de forma manual;
- Realizar o mapeamento dos cenários dos clientes para criar ambientes de teste similares ao ambiente de produção. Esses cenários são importantes para execução dos testes de regressão, tornando os testes mais condizentes com a realidade dos clientes, e;
- Estabelecer alguns indicadores de qualidade e métricas para iniciar um processo de medição, tornando possível a aplicação de melhorias futuras.

6.2. O Processo Proposto

Com base nas melhorias supracitadas, torna-se viável a aplicação de mudanças no processo de teste atualmente utilizado pela empresa. Em virtude disso, a proposta do processo de teste melhorado será detalhada a seguir.

A partir das demandas dos clientes, já aprovadas pelo *Product Owner*, o *Scrum Master* realiza o planejamento da *Sprint*, designando as tarefas que serão desenvolvidas no quadro de *kanban*.

Com a *Sprint* planejada, o testador cadastra os cartões referentes aos testes que serão realizados, e vincula-os as tarefas de desenvolvimento. A Figura 8 mostra o mapeamento das melhorias propostas para o planejamento dos testes.

No momento do cadastro dos cartões, será verificado se existe caso de teste referente à funcionalidade que será testada. Caso exista, o mesmo será atualizado com base nas alterações efetuadas. Se não existir, o testador deve criá-lo. Em ambas as situações, o caso de teste é vinculado ao cartão.

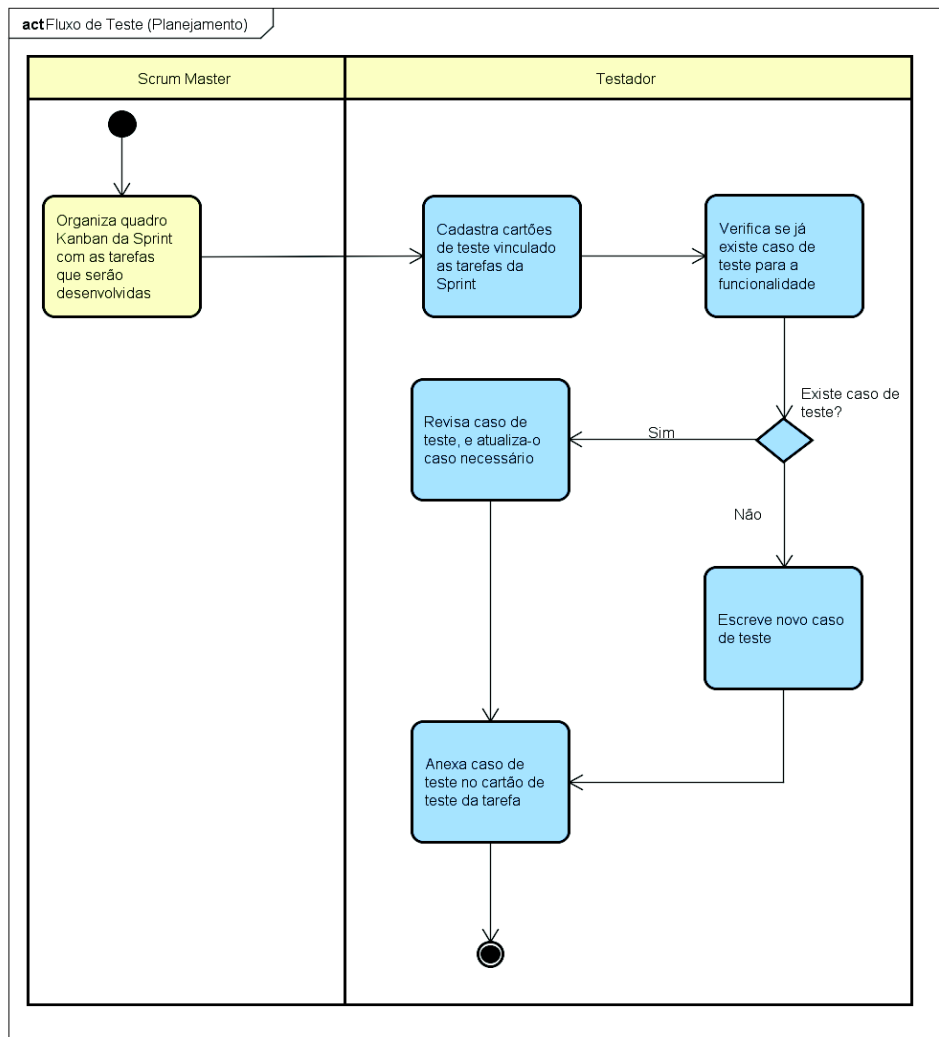
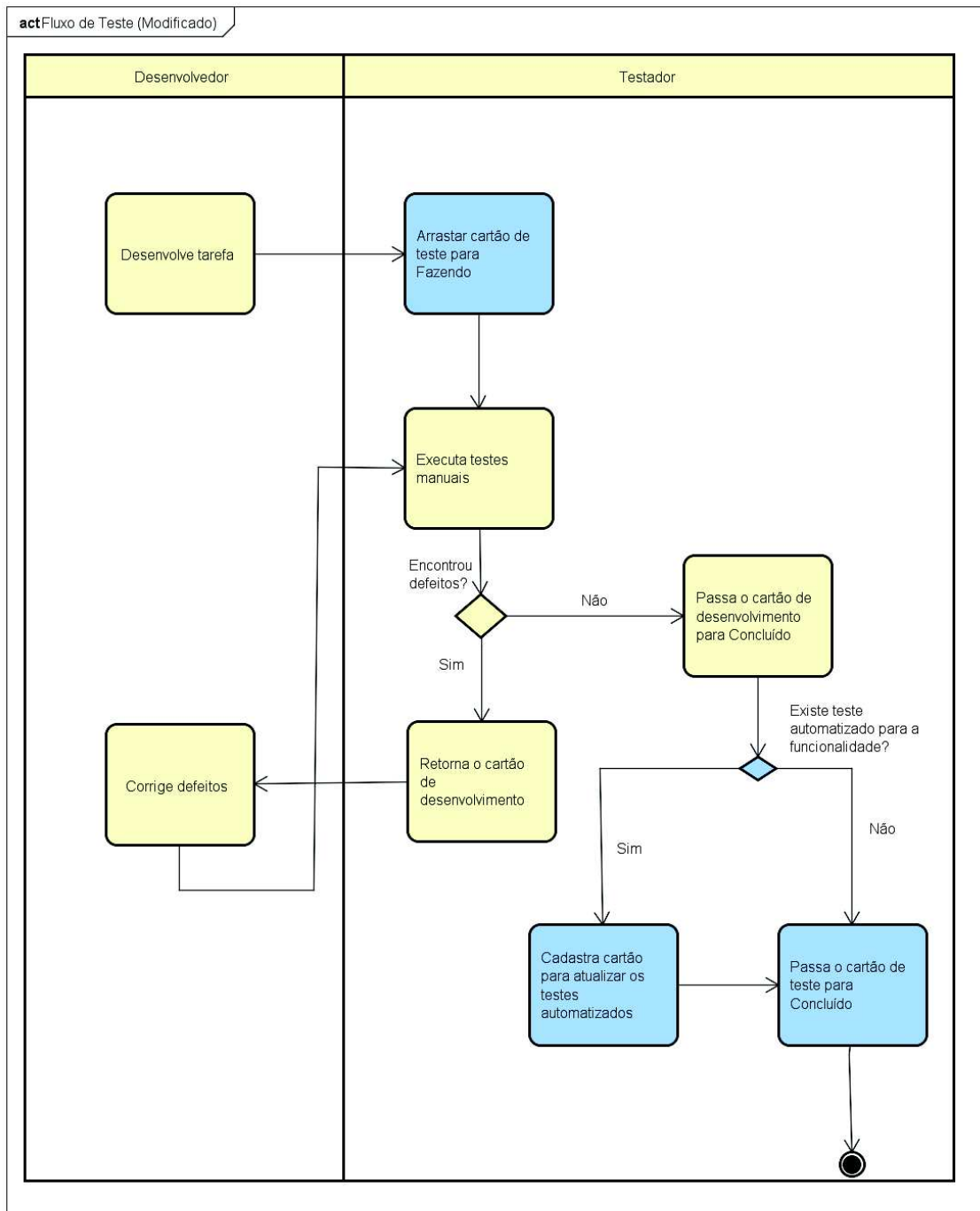


Figura 8. Planejamento dos testes que serão realizados na Sprint

Fonte: Elaborado pela autora

Como pode ser observado na Figura 8, além de relatar os cartões de teste, para todos os testes que serão realizados na Sprint, previamente, o testador pode controlar os casos de teste e mantê-los sempre atualizados.

Assim que o desenvolvedor finaliza o desenvolvimento da tarefa, o testador assume o cartão de teste, previamente cadastrado, e realiza os testes necessários. O processo melhorado para execução dos testes manuais pode ser visualizado na Figura 9.



powered by Astah

Figura 9. Processo de teste proposto a partir das melhorias sugeridas

Fonte: Elaborado pela autora

Como demonstra o fluxograma da Figura 9, durante a execução dos testes, caso sejam encontrados defeitos na tarefa, o cartão de desenvolvimento retorna ao desenvolvedor para que sejam realizadas as correções necessárias. Em seguida, os testes são novamente executados, para averiguar se os ajustes realizados estão em conformidade. O testador finaliza os testes e conclui os cartões, de desenvolvimento e de teste, quando constata que não há inconformidades. O ambiente para execução dos testes baseia-se no ambiente de produção, conforme mapeamento dos cenários dos clientes.

Também pode ser observado na Figura 9, que o testador verifica, ao finalizar a execução do teste, se a funcionalidade em questão já está automatizada. E se necessário, cadastra um cartão para atualização desses cenários.

A automação dos testes funcionais se dará a partir da priorização das principais funcionalidades do sistema. Uma vez priorizadas, as tarefas são criadas e alocadas nas próximas *Sprints*. Os testes serão desenvolvidos pelo testador, e normalmente, na primeira semana da *Sprint*, visto que as principais funcionalidades estão em fase de desenvolvimento.

Os testes de regressão serão realizados assim que o desenvolvimento da versão for finalizado, conforme o roteiro previamente elaborado. Os testes automatizados, em conjunto com alguns testes manuais exploratórios, serão executados. Em um primeiro momento, apenas as funcionalidades consideradas mais importantes do sistema, além das funcionalidades que já estão automatizadas, receberão os testes de regressão. Em razão de que é necessário ampliar os cenários automatizados para a maioria das funções existentes, portanto, inicialmente a regressão será realizada de forma manual em algumas situações.

Por fim, e buscando uma melhoria contínua, definiu-se inicialmente, a utilização de dois indicadores, que são: o número de defeitos relatados pelos clientes, e a quantidade de defeitos encontrados pela equipe nas versões já concluídas. A partir dessa medição, torna-se possível identificar como as versões são finalizadas e entregues aos clientes. E a partir dos resultados, verificar se é viável realizar novas alterações em algumas etapas do processo de teste.

6.3. Aplicação do Processo Proposto

Depois da validação das melhorias sugeridas e da proposta do novo processo de teste, o mesmo foi aplicado ao projeto em andamento na empresa. O testador foi o responsável pela sua execução, por duas *Sprints* antes de ser avaliado, devido ao curto período de tempo disposto para o desenvolvimento desse trabalho.

Na primeira *Sprint*, durante o planejamento dos testes, verificou-se que não existiam casos de teste para as funcionalidades que seriam alteradas. Dessa forma, foi necessário criá-los, o que acabou prolongando o processo. Foram criados 11 casos de teste, os quais foram armazenados em um servidor local, além de serem anexados aos cartões de teste cadastrados. A pontuação dos cartões de teste foi realizada juntamente com as demais tarefas da *Sprint*, e os desenvolvedores tiveram certa incerteza no início, devido ao fato de que essa etapa não era executada anteriormente.

Após a organização dos testes e dos casos de teste da *Sprint* 01, iniciou-se o processo de planejamento da automação dos testes funcionais. Uma reunião com o *Scrum Master*, o testador e o gerente de desenvolvimento foi realizada, com o intuito de priorizar as funcionalidades e os cenários do sistema a serem automatizados. Em seguida, o testador relatou os cartões de teste, necessários para escrita dos testes, que foram alocados nas *Sprints* em aberto, conforme a sua prioridade.

Como alguns casos de teste já haviam sido cadastrados na *Sprint* 01, o planejamento dos testes da segunda *Sprint* ocorreu de forma mais rápida. Foi necessário cadastrar 7 casos de teste, além da atualização de 5 que já estavam cadastrados. A estimativa dos cartões de teste foi realizada de forma mais assertiva e tranquila do que a anterior, visto que os desenvolvedores se familiarizaram com o processo, o qual já havia sido executado na primeira *Sprint*.

Na *Sprint* 01, não foi possível realizar a automação de um novo cenário de teste, visto que a duração da *Sprint* é de 15 dias, e o planejamento e escrita dos casos de teste prolongou-se mais do que o esperado. Entretanto, tornou-se viável a atualização de duas funcionalidades que já estavam automatizadas, mas que foram alteradas. Na *Sprint* 02, realizou-se a automação de alguns cenários de uma das principais funcionalidades do sistema, devido a organização e planejamento feitos na *Sprint* anterior, também foi possível atualizar 2 cenários de teste já automatizados, que sofreram alterações na funcionalidade durante o andamento da *Sprint*.

A execução dos testes manuais se deu de forma tranquila nas duas *Sprints*, por consequência do planejamento realizado no início. A criação dos casos de teste previamente facilitou o trabalho do testador, que executou os testes de forma mais ágil, em comparação com as *Sprints* anteriores.

Ao fim de cada uma das *Sprints*, realizou-se o teste de regressão. Nesse momento, os testes automatizados já existentes, em conjunto com testes manuais para as principais funcionalidades do sistema que não estão automatizadas, foram executados. No decorrer do teste de regressão, foram encontrados 18 defeitos, que são decorrentes de versões antigas.

Nas duas *Sprints* tomadas como base para avaliação do processo melhorado, foram relatados 2 defeitos em produção, porém não foi necessário realizar versões de emergência, visto que os mesmos não foram considerados graves. Dessa forma, percebe-se que durante a utilização do processo proposto neste trabalho, houve uma diminuição nos relatos de defeitos em produção.

Devido ao custo espaço de tempo previsto para aplicação do processo de teste melhorado na empresa, não foi possível coletar os resultados dos indicadores de desempenho que foram definidos, porém realizou-se a alteração na ferramenta Mantis para relato dos defeitos, de forma que as informações necessárias sejam preenchidas.

6.4. Resultado da Avaliação

Para avaliar a aplicabilidade e aderência das melhorias propostas ao processo de teste utilizado pela empresa estudada, aplicou-se um questionário, que foi respondido pelos programadores, além do gerente de desenvolvimento da empresa. O testador da equipe não foi selecionado, visto que o mesmo realizou a pesquisa e é a autora do presente

trabalho. O questionário foi disponibilizado de forma *online*, através da ferramenta Google Forms⁵.

O questionário é composto por duas questões para identificação do perfil do respondente e uma para validar a importância do processo de teste na empresa. Também fazem parte duas perguntas para avaliar o processo de teste anterior, e quatro para avaliação do processo proposto nesse trabalho. Como citado anteriormente, as questões foram adaptadas da escala *Likert*. O questionário, bem como os seus resultados podem ser observados no APÊNDICE A.

Responderam ao questionário, dois programadores e um gerente, responsável pelo setor de desenvolvimento. Eles estão na empresa entre 3 a 6 anos, e mais de 6 anos, respectivamente. Os três respondentes consideram que os testes são muito importantes na organização, e todos eles avaliaram o processo utilizado pela empresa como sendo Regular.

A respeito da principal carência do processo de teste utilizado, as respostas divergem entre elas. O teste de regressão está presente em duas das respostas. Uma delas menciona que quando executado, o mesmo não previa os cenários dos clientes, o que ocasionava versões de emergência, para correção de defeitos encontrados em produção. A outra resposta, afirma que sentia falta do teste de regressão, além de que não era possível saber o real esforço despendido para os testes nas *Sprints*, porque os cartões referentes aos testes estavam sendo cadastrados apenas para algumas situações, e quando cadastrados, a pontuação não estava sendo realizada corretamente. Por fim, a falta de organização do que e de como testar, e também com as técnicas a serem utilizadas durante a execução dos testes, foram expostas na resposta de um dos colaboradores.

A avaliação do processo de teste melhorado foi positiva. Dois dos respondentes o julgaram como Bom, e um deles como Muito Bom. Constatou-se também, a partir da opinião dos envolvidos, que houve uma melhoria na execução dos testes após a adoção do processo proposto. Na Figura 10, é possível visualizar as respostas da avaliação do processo de teste proposto neste trabalho.

⁵ Informações sobre a ferramenta estão disponíveis em: <https://www.google.com/forms/about/>

Após avaliar as melhorias sugeridas, como você considera o processo de teste proposto?

(3 respostas)

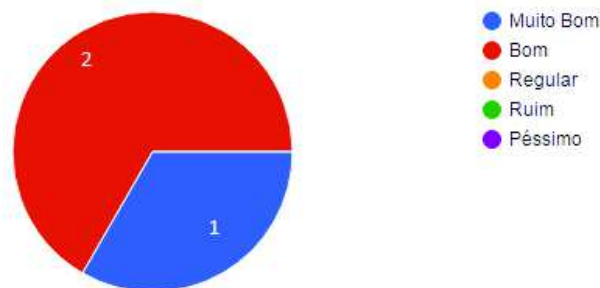


Figura 10. Avaliação do Processo de Teste Proposto

Fonte: Elaborado pela autora

Perguntou-se aos avaliadores do processo melhorado, quais as situações que ainda necessitam da aplicação de melhorias. Para dois deles, ampliar os testes automatizados, é uma delas. No ponto de vista dos envolvidos, a automação traz uma maior agilidade para o teste de regressão, que passou a ser utilizado no novo processo, além de possibilitar ao testador, a realização de testes manuais para os cenários de exceção. Outra sugestão, é que a gestão dos casos de testes deve ser aperfeiçoada, principalmente a manutenção dos mesmos. Aprimorar a utilização dos indicadores é a recomendação do gerente, para que seja possível aplicar melhorias no processo de forma frequente.

Por fim, as principais melhorias trazidas com a aplicação do processo de teste proposto, na opinião dos respondentes, são as seguintes:

- Através dos cartões de teste, e sua pontuação, é possível verificar, em formato mais acessível, o real esforço dos testes;
- Planejamento dos testes de forma mais adequada;
- Uma melhor organização na execução dos testes, a partir da definição do roteiro e dos cenários a serem seguidos;
- Os testes tornaram-se mais amplos, por consequência do planejamento e da organização citadas anteriormente, e;
- Os testes de regressão estão obtendo resultados mais satisfatórios, principalmente por ter os principais cenários dos clientes mapeados.

7. Conclusão

O teste de software torna-se essencial para assegurar a qualidade final no desenvolvimento do produto. Nesse contexto, o presente projeto teve como objetivo melhorar o processo de teste em uma empresa de desenvolvimento de software, de pequeno porte.

No final do projeto, pode-se concluir que foi possível entender, validar e propor melhorias no processo de teste da empresa estudada. A aplicação do processo melhorado, bem como a avaliação do mesmo junto aos envolvidos, também foi realizada durante o decorrer do trabalho.

Através do mapeamento do processo, chegou-se aos pontos positivos e negativos do mesmo. Elencaram-se melhorias a serem empregadas, com o intuito de atingir as situações consideradas negativas. Com base nessas melhorias, foi possível realizar a proposta de um novo processo de teste para a empresa.

O curto espaço de tempo para implantação e para avaliação do processo de teste melhorado, aliados ao tamanho da equipe de desenvolvimento e da empresa, foram as limitações encontradas durante o desenvolvimento deste trabalho.

Como forma de avaliar o processo melhorado, um questionário foi elaborado e aplicado aos desenvolvedores e ao gerente do setor de desenvolvimento da empresa. O questionário é composto por questões referentes ao processo utilizado anteriormente, e também perguntas relacionadas à utilização do novo processo de teste.

O uso do questionário permitiu entender qual a percepção dos envolvidos com relação aos testes executados anteriormente pela empresa, bem como, quais eram as principais situações que eles consideravam que precisavam de mudanças. Possibilitou também, que os avaliadores dessem o seu parecer sobre o processo proposto neste trabalho. As principais melhorias trazidas após a utilização do processo de teste melhorado na organização, e os aspectos que ainda podem ser aperfeiçoados, também foram abordadas nas questões.

No ponto de vista dos pesquisados, apesar de o processo anterior estar em uso já há algum tempo, pelo menos há três anos, algumas carências eram perceptíveis. Os mesmos reconheceram e elencaram quais os aspectos prováveis de correção, ou seja, os pontos negativos que precisavam ser melhorados.

Considerando que os resultados referentes à utilização do processo de teste proposto neste projeto foram satisfatórios, pois os avaliadores o julgaram superior ao utilizado anteriormente, pode-se concluir que o principal objetivo desse projeto foi alcançado. Isso indica que no ponto de vista dos colaboradores, a proposta trouxe algum benefício na execução dos testes.

Após a análise das respostas, pode-se constatar que as principais carências existentes nos testes realizados anteriormente foram corrigidas após a utilização do processo melhorado. Cabe ressaltar que algumas melhorias ainda podem ser aplicadas ao processo, visto que os pesquisados elencaram quais as situações, que na visão deles, ainda podem ser aperfeiçoadas.

Como trabalhos futuros, pretende-se ampliar a medição do processo de teste na empresa analisada, uma vez que foram aplicados apenas dois indicadores durante a

realização dessa pesquisa. Espera-se através disso, manter uma melhoria contínua no processo, bem como poder evidenciar as etapas que precisam de alterações.

8. Referências

- Alvarenga, T. H. P., Piekarski, C. M., Santos, B. S., Bittencourt, J. V. M., Matos, E. A. S. A. and Francisco, A. C. (2013) “Mapeamento de Processos: uma revisão” In: VIII Encontro de Engenharia e Tecnologia dos Campos Gerais, Ponta Grossa, PR.
- Anacleto, A. (2004) “Método de Avaliação para Melhoria de Processos de Software em Micro e Pequenas Empresas”, Dissertação de Mestrado, Universidade Federal de Santa Catarina, Florianópolis, SC.
- Andrade, M. de. (2010), Introdução à metodologia do trabalho científico: elaboração de trabalhos na graduação, São Paulo, Atlas, 10ª edição.
- Angrosino, M. (2011), Etnografia e observação participante, Porto Alegre, Bookman: ArtMed.
- Azevedo, D., Machado, L. and Silva, L. V. da. (2011) “Métodos e procedimentos de pesquisa: do projeto ao relatório final”, São Leopoldo, RS.
- Bissi, W. (2007) “Scrum – Metodologia de Desenvolvimento Ágil”, Vol. 2, Nº 1, Campo Mourão, PR.
- Camargo, K. G. (2012) “Elaboração de um Processo de Teste com Base em um Estudo de Caso Real”, Dissertação de Mestrado, Universidade Federal de São Carlos, São Carlos, SP.
- Campos, F. M. (2016) “Qualidade, Qualidade de Software e Garantia da Qualidade de Software são as mesmas coisas?”, <http://www.linhadecodigo.com.br/artigo/1712/qualidade-qualidade-de-software-e-garantia-da-qualidade-de-software-sao-as-mesmas-coisas.aspx>, Agosto.
- Costa, F. J. and Junior, S. D. S. (2014) “Mensuração e Escalas de Verificação: uma Análise Comparativa das Escalas de Likert e Phrase Completion”, Revista Brasileira de Pesquisas de Marketing, Opinião e Mídia, Vol. 15, São Paulo, http://www.revistapmkt.com.br/Portals/9/Volumes/15/1_Mensura%C3%A7%C3%A3o%20e%20Escalas%20de%20Verifica%C3%A7%C3%A3o%20uma%20An%C3%A1lise%20Comparativa%20das%20Escalas%20de%20Likert%20e%20Phrase%20Completion.pdf, Setembro.
- Crespo, A. N., Silva, O. J., Borges, C. A., Salviano, C. F., Junior, M. T. A. and Jino, M. (2004) “Uma Metodologia para Teste de Software no Contexto da Melhoria de Processo”, São Paulo, SP.
- Creswell, J. W. (2010), Projeto de Pesquisa: Métodos Qualitativo, Quantitativo e Misto, Porto Alegre, Bookman: Artmed, 3ª edição.
- Crispin, L. and Gregory, J. (2009), Agile Testing: A Practical Guide For Testers and Agile Teams, Boston, Person Education Inc, 1ª edição.
- Cunha, D. (2009) “Mais agilidade em suas estimativas com o Planning Poker”, <http://www.olharcritico.com/engenhariadesoftware/2009/12/mais-agilidade-em-suas-estimativas-com-o-planning-poker/>, Julho.

- Donner, M., Ellis, T. and Swift, K. (2001) “A technique for revealing and agreeing an agenda for process improvement”, *Journal of Materials Processing Technology*, Vol. 118.
- Duarte, K. C. and Falbo, R. A. (2000) “Uma Ontologia de Qualidade de Software”, *Dissertação de Mestrado, Universidade Federal do Espírito Santo, Vitória.*
- Engel, G. I. (2000) “Pesquisa-ação. *Revista Educar*. [on-line]”, Editora da UFPR, N.º 16, Curitiba, http://www.educaremrevista.ufpr.br/arquivos_16/irineu_engel.pdf, Setembro.
- Gil, A. C. (2010), *Como elaborar projetos de pesquisa*, São Paulo, Atlas, 5ª edição.
- Gouveia, C. C. (2004) “Teste de Integração para Sistemas Baseados em Componentes”, *Dissertação de Mestrado, Universidade Federal da Paraíba, Campina Grande.*
- Habib, E. (2016) “Testes Ágeis: Como os testes serão modificados com o advento das metodologias ágeis”, *Engenharia de Software Magazine*, Vol. 27, <http://www.garcia.pro.br/EngenhariadeSW/artigosMA/A4%20-%2027-7-%20Testes%20Ageis.pdf>, Julho.
- Hammer, M. (2002), *Process management and the future of Six Sigma*, Massachusetts, Sloan Management Review.
- Klein, A. Z., Silva, L. V., Machado, L. and Azevedo, D. (2015), *Metodologia de Pesquisa em Administração: Uma Abordagem Prática*, São Paulo, Atlas.
- Lüdke, M. and André, M. E. D. (2013), *Pesquisa em Educação - Abordagens Qualitativas*, Rio de Janeiro, E.P.U., 2ª edição.
- Mansar, S. L. and Reijers, H.A. (2007) “Best practices in business process redesign: use and impact”, *Business Process Management Journal*, Vol. 13, N.º 2.
- Mello, C. H. P. and Salgado, E. G. (2005) “Mapeamento dos processos em serviços: estudo de caso em duas pequenas empresas da área de saúde” In: *XXV Encontro Nac. de Eng. de Produção*, Porto Alegre, RS.
- Myers G. J. (2012), *The Art of Software Testing*, New York, Wiley.
- Neto, A. C. D. (2010) “Introdução a Teste de Software”, *Engenharia de Software Magazine*, Vol. 1, https://www.researchgate.net/profile/Arilo_Neto/publication/266356473_Introducao_a_Testes_de_Software/links/5554ee6408ae6fd2d821ba3a.pdf, Julho.
- Oliveira, E. S. (2003) “Uso de Metodologias Ágeis no Desenvolvimento de Software”, *Trabalho de Conclusão de Curso, Universidade Federal de Minas Gerais, Belo Horizonte, MG.*
- Paim, R., Cardoso, V., Caulliraux, H. and Clemente, R. (2011), *Gestão de Processos: Pensar, Agir e Aprender*, Porto Alegre, Bookman.
- Pfleeger, L. S. (2007), *Engenharia de Software: Teoria e Prática*, São Paulo, Pearson Education BR, 2ª edição.
- Pressman, R. S. (2011), *Engenharia de Software: Uma abordagem profissional*, AMGH Editora LTDA, 7ª edição.

- Prettz, J. B. and Bertolini, C. (2016) “Uma Proposta de um Processo de Teste em uma Empresa de Pequeno Porte”, <http://w3.ufsm.br/frederico/images/UmaPropostadeumProcessodeTesteEmpresaPequenoPorte.pdf>, Setembro.
- Sanches, C., Meireles, M. and Sordi, J. O. (2011) “Análise Qualitativa Por Meio da Lógica Paraconsistente: Método de Interpretação e Síntese de Informação obtida Por Escalas Likert” In: 3º Encontro de Ensino e Pesquisa em Administração e Contabilidade, João Pessoa, PB.
- Sartori, L. E. S. (2005) “Melhoria do Processo de Teste para Pequenas Empresas”, Dissertação de Mestrado UNIVEM - Fundação de Ensino “Eurípides Soares da Rocha”, Marília, SP.
- Scartezini, L. M. B. (2009) “Análise e Melhoria de Processos”, Goiânia, GO.
- Shaughnessy, J. J., Zechmeister, E. B. and Zechmeister, J. S. (2012) Metodologia de Pesquisa em Psicologia, Porto Alegre, AMGH, 9ª edição.
- Silva, D. D. (2015) “Melhoria do Processo de Teste para as Micro e Pequenas Empresas Brasileiras” Dissertação de Mestrado, Universidade Federal de Goiás, Goiânia, GO.
- Silva, J. R. S., Almeida, C. D. and Guindani, J. F. (2009) “Pesquisa documental: pistas teóricas e metodológicas”, Revista Brasileira de História & Ciências Sociais, Nº 1.
- Silveira, C. B. (2012) “Sistema Kanban: conceito e softwares”, <http://www.citisystems.com.br/kanban-conceito-sistema-o-que-e-on-line/>, Julho.
- Sommerville, I. (2011), Engenharia de Software, São Paulo, Pearson Education BR, 9ª edição.
- Tripp, D. (2005) “Pesquisa-ação: uma introdução metodológica. Educação e Pesquisa”, Vol. 31, São Paulo, <http://www.scielo.br/pdf/ep/v31n3/a09v31n3.pdf>, Setembro.
- Valle, R. and Oliveira, S. B. de. (2009), Análise e modelagem de processos de negócio: foco na notação BPMN (Business Process Modeling Notation), São Paulo, Atlas.
- Vaz, J. C. (2008) “Processos de trabalho no setor público: gestão e redesenho”, <http://josecarlosvaz.pbworks.com/w/page/8532009/Redesenho%20de%20Processos>, Junho.
- Vicente, A. A. (2010) “Definição e gerenciamento de métricas de teste no contexto de métodos ágeis”, Dissertação de Mestrado, USP – Universidade de São Paulo, São Paulo, SP.
- Villela, C. S. S. (2000) “Mapeamento de Processos como Ferramenta de Reestruturação e Aprendizado Organizacional”, Dissertação de Mestrado, Universidade Federal de Santa Catarina, Florianópolis, SC.

APÊNDICE A – Questionário e seus Resultados

Avaliação do Processo de Teste

Este questionário visa avaliar as melhorias que foram aplicadas ao processo de teste da empresa, bem como o processo anterior utilizado. O mesmo procura compreender a opinião dos colaboradores e envolvidos nesse processo.

*Obrigatório

Qual a sua função na empresa? *

- Analista de Sistema
- Programador
- Testador
- Outro:

Há quanto tempo você trabalha na empresa? *

- Menos de 1 ano
- De 1 a 3 anos
- De 3 a 6 anos
- Mais de 6 anos

Na sua opinião, qual a importância do processo de teste na empresa? *

- Muito Importante
- Importante
- Pouco Importante
- Não é Importante

Figura 1. Questionário aplicado aos avaliadores

Fonte: Elaborado pela autora

Como você considerava o processo de teste utilizado anteriormente? *

- Muito Bom
- Bom
- Regular
- Ruim
- Péssimo

Na sua opinião, qual era a principal carência desse processo? *

Sua resposta _____

Após avaliar as melhorias sugeridas, como você considera o processo de teste proposto? *

- Muito Bom
- Bom
- Regular
- Ruim
- Péssimo

Após a aplicação do processo proposto, você acredita que houve uma melhoria na execução dos testes? *

- Sim
- Não

Figura 2. Questionário aplicado aos avaliadores

Fonte: Elaborado pela autora

Se a sua resposta foi Não, porque consideras que as melhorias não tiveram resultado?

Sua resposta

Você reconhece algo que ainda deve ser melhorado? O que?

Sua resposta

Em sua opinião, qual a principal melhoria trazida com esse processo? *

Se achas que não houve melhorias com o processo, por favor descreva aqui.

Sua resposta

ENVIAR

Nunca envie senhas pelo Formulários Google

Este conteúdo não foi criado nem aprovado pelo Google. Denunciar abuso - Termos de Serviço - Termos Adicionais

Google Forms

Figura 3. Questionário aplicado aos avaliadores

Fonte: Elaborado pela autora

Tabela 1. Respostas do questionário

	Programador	Programador	Gerente de Desenvolvimento
Tempo de Empresa	Entre 3 a 6 anos	Entre 3 a 6 anos	Mais de 6 anos
Qual a importância do processo de teste na empresa?	Muito Importante	Muito Importante	Muito Importante
Como você considerava o processo de teste utilizado anteriormente?	Regular	Regular	Regular
Qual a principal carência desse processo?	Os testes de regressão, que quando executados, não estavam cobrindo os principais cenários dos clientes, aumentando as chances de uma "build" emergencial para corrigir defeitos encontrados em ambiente de produção	Organização do que testar e como testar. Somente as entregas pontuais? E como fazer isso. Quais técnicas utilizar.	Acredito que a principal carência é a falta do teste de regressão, como há poucos cenários automatizados, é muito custoso realizar a regressão de forma manual no sistema. Posso citar também que o esforço dos testes não são reais. Não temos como saber realmente quanto foi despendido em testes, pois não está sendo cadastrados cartões para todos os testes que são realizados, apenas para alguns deles. Também não está sendo realizado as devidas pontuações para esses cartões de testes, quando eles são cadastrados.
Após avaliar as melhorias sugeridas, como você considera o processo de teste proposto?	Muito Bom	Bom	Bom
Após a aplicação do processo	Sim	Sim	Sim

proposto, você considera que houve uma melhoria na execução dos testes?			
Se a sua resposta foi Não, porque consideras que as melhorias não tiveram resultado?	-	-	-
Você reconhece algo que ainda deve ser melhorado? O que?	Sim. Ampliar a cobertura dos testes automatizados, possibilitando assim, dedicar um tempo maior para os testes de exceção.	Gerenciamento dos casos de teste. Manutenção dos mesmos.	Acredito que devemos trabalhar para aumentar os testes automatizados das principais funcionalidade do sistema. Pois eles proporcionam uma agilidade durante a regressão, fazendo com que o testador não precise, em toda versão liberada testar sempre as mesmas coisas de forma manual. Temos que aprimorar a utilização de indicadores, para continuar aplicando melhorias no processo.
Em sua opinião, qual a principal melhoria trazida com esse processo?	Com os principais cenários dos clientes mapeados, os testes de regressão estão conseguindo obter resultados satisfatórios. Outro ponto a ser destacado é que com a criação dos cartões de testes, tornou-se mais fácil demonstrar o real esforço da equipe de testes.	Melhorias: Pontuação das tarefas de teste. Planejamento dos casos de teste. Organização dos testes e roteiros/cenários.	Saber o real esforço dos testes foi uma melhoria importante trazida com esse novo processo. A realização de testes de regressão, que antes não eram executados sempre, e levar em consideração o cenário dos clientes, também é uma vantagem desse processo.

Fonte: Elaborado pela autora