UNIVERSIDADE DE SÃO PAULO

FLAVIA MOREIRA MUCEDOLA

Estratégia de implantação de testes automatizados em ambiente ágil

São Paulo

2015

FLAVIA MOREIRA MUCEDOLA

Estratégia de implantação de testes automatizados em ambiente ágil

Monografia apresentada ao PECE – Programa de Educação Continuada em Engenharia da Escola Politécnica para obtenção do título de Especialista em Tecnologia da Informação

Orientador: Profª. MsC. Ana Claudia Rossi

São Paulo

2015

FLAVIA MOREIRA MUCEDOLA

Estratégia de implantação de testes automatizados em ambiente ágil

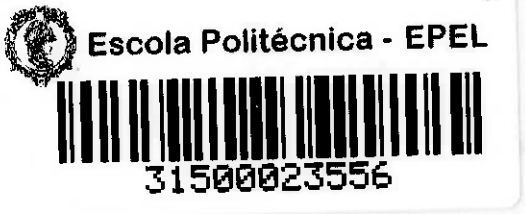
Monografia apresentada ao PECE – Programa de Educação Continuada em Engenharia da Escola Politécnica para obtenção do título de Especialista em Tecnologia da Informação

Área de Concentração: Tecnologia da Informação

Orientador: Profª. MsC. Ana Claudia Rossi

São Paulo

2015



FICHA CATALOGRÁFICA

Mucedola, Flavia Moreira

Estratégia de implantação de testes em ambiente ágil / F.M. Mucedola. --

São Paulo, 2015.

65 p.

Monografia (MBA em Tecnologia da Informação) - Escola Politécnica da Universidade de São Paulo. Programa de Educação Continuada em Engenharia.

DEDICATÓRIA

Dedico este trabalho à minha família,
e meus amigos, que me incentivaram e
me apoiaram para que eu não
desistisse e continuasse estudando.

AGRADECIMENTOS

Gostaria de agradecer a orientadora Ana Claudia Rossi, pela orientação e pela força desde o começo até o fim do trabalho.

Aos amigos do trabalho pelas discussões e experiências trocadas agregando ao trabalho realizado,

E a todos meus amigos que ajudaram incentivando e ajudando na execução deste trabalho.

RESUMO

O processo de desenvolvimento ágil tem se tornando um dos mais importantes recursos para melhorar o desenvolvimento dos softwares nas empresas. O processo de desenvolvimento ágil indica que o processo de testes tem que estar integrado junto ao processo de desenvolvimento e com esse fim possui vários modelos e metodologias para implantar o teste automatizado dentro do ambiente ágil, mas os trabalhos apresentam limitações e levam as organizações a falhas, uma vez que o teste continua sendo realizado no final do processo de desenvolvimento. Para a consecução do objetivo foi desenvolvida uma pesquisa bibliográfica, gerando a primeira versão de uma estratégia de implantação de testes automatizados em ambiente ágil, estratégia onde integra o processo de teste junto à fase de desenvolvimento e junto a técnicas ágeis mostra que a automação dos testes iniciada desde o inicio do método apresenta uma grande perspectiva de diminuição de falhas no final do processo e realiza a integração do time.

Palavras-Chave: Testes automatizados. Scrum. Técnicas ágeis. BDD. TDD. AMM. Método de instanciação de processos. TMM.

ABSTRACT

Agile development process is becoming one of the most important features to enhance the development of software in companies. Agile development process indicates that the testing process has to be integrated with the development process and to this end has several models and methodologies to deploy automated test within the agile environment, but jobs have limitations and failures lead to organizations once the test is still held at the end of the development process. To achieve the goal a literature was developed, creating the first version of a deployment strategy for automated testing in agile environment, strategy which integrates testing within the development phase and along the agile techniques shows that the automation of tests initiated from the start method has a wide disaster diminishing perspective the end of the process and performs the integration of the team.

Keywords: Automated Testing. Scrum. Agile techniques. BDD. TDD. AMM. Method instantiation processes. TMM.

LISTA DE ILUSTRAÇÕES

LISTA DE ABREVIATURA E SIGLAS

AMM	Agile Maturity Model
TMM	Test Maturity Model
BDD	Behavior Driven Development
TDD	Test Driven Development

SUMÁRIO

1. INTRODUÇÃO

Este capítulo discute o contexto e motivação do presente trabalho (seção 1.1), descreve o problema encontrado (seção 1.2), o objetivo desse trabalho (seção 1.3), a justificativa (seção 1.4) e a estrutura do trabalho (seção 1.5).

1.1 Contexto Inicial

Com a crescente evolução no desenvolvimento de software, a engenharia de software precisou evoluir e uma das evoluções em destaque são os métodos ágeis.

Com o uso de métodos ágeis, as fases do desenvolvimento precisam se adaptar e o teste do software que é uma das fases do desenvolvimento precisa acompanhar essa evolução. Com isso, uma das formas estudada para o processo de desenvolvimento ágil será a fase de testes.

A fase de testes de software garante a qualidade do produto e para garantir qualidade as empresas têm cada vez mais procurado formas de melhorar o processo de teste e garantir uma cultura de testes para a área de desenvolvimento dentro das empresas.

Os modelos criados para melhoria no processo de testes são voltados também para a melhoria do desenvolvimento de software e existem alguns modelos como o *Test Process Improvement (TPI)*, como o *Teste Maturity Model (TMM)* e *Test Capability Maturity Model (TCMM)*. O modelo utilizado e estudado é o *Test Maturity Model (2012)*.

O Modelo de maturidade de teste TMM segundo (Rios, Cristalli) é um modelo criado para melhoria no processo de teste e utilizaremos o mesmo para atingir uma maturidade no modelo de teste dentro do ambiente ágil.

Os testes no software podem ser feitos tanto manualmente quanto automaticamente. O teste automatizado quando comparado ao teste manual aumenta a rapidez na reprodução dos testes de algumas funcionalidades do sistema o qual se identificou a

viabilidade de automação, e com a automação os casos de teste são executados iterativamente, maximizam a cobertura do teste dentro do tempo disponível e aumentam a confiabilidade do teste e a qualidade do software.

Segundo Patel, os métodos ágeis de desenvolvimento de software introduziram melhores práticas em desenvolvimento de software, porém é preciso adotar e monitorar essas práticas continuamente para que seus benefícios sejam maximizados. O modelo de maturidade ágil AMM é um framework conceitual de práticas e princípios para desenvolver software mais rápido, de forma incremental, para deixar o cliente satisfeito e tem foco na adaptabilidade, adequação e na maturidade de um modelo de software.

Diversos métodos ágeis são sugeridos na literatura, como o Scrum, XP, Lean, porém todos eles adotam princípios do Manifesto ágil tais como, o desenvolvimento iterativo, entregas frequentes e software simples. De acordo com Manifesto Ágil (2001)

> *"A agilidade é a capacidade de criar e responder a mudanças, a fim de lucrar em um turbulento ambiente de negócios ".*
> *(MANIFESTO ÁGIL DE SOFTWARE)*

Como o processo de desenvolvimento ágil do software precisa mudar rapidamente as mudanças nos requisitos e priorizar o desenvolvimento das funcionalidades, obtendo assim resposta rápida as mudanças, é preciso de um método ágil que melhor se adeque ao processo de testes no desenvolvimento do software, e o método ágil Scrum será utilizado por ser o mais indicado e utilizado pelas empresas.

VersionOne (2009) realizou uma pesquisa onde o Scrum foi apresentado como o método mais utilizado pelas empresas e o que mais se destaca no mercado.

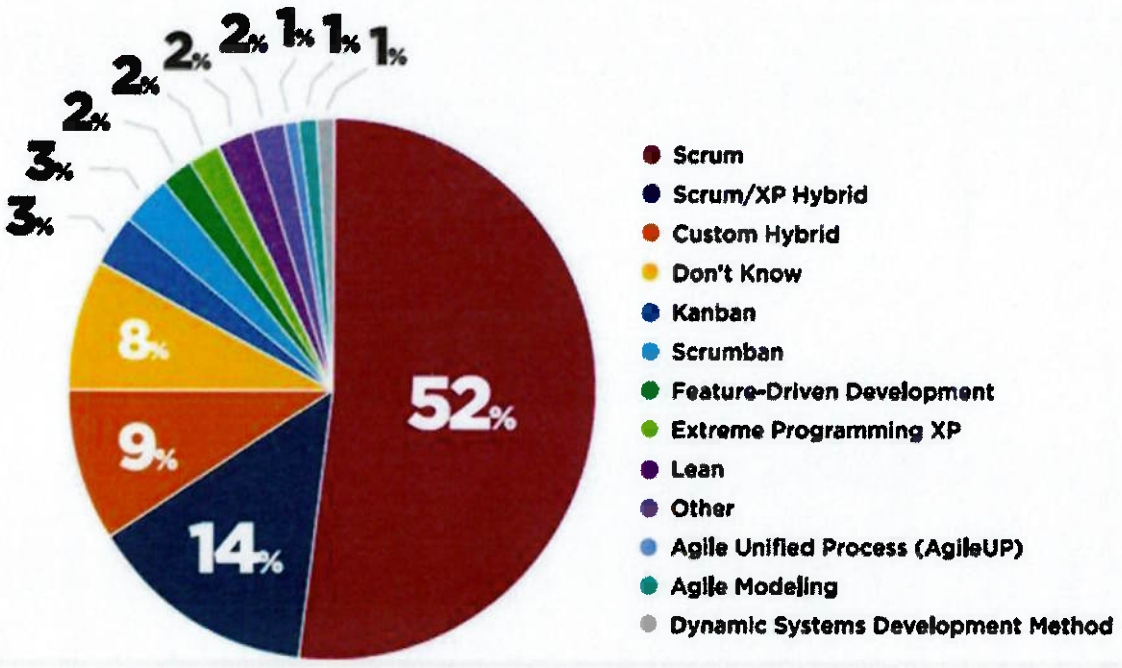


Figura 1 – Métodos mais utilizados nas empresas
Fonte: VersionOne (2009)

O Scrum foi criado pelos americanos Jeff Sutherland e Ken Schwaber e sua definição de acordo com a versão em português do Guia do Scrum 2011, é:

> *"Um framework dentro do qual pessoas podem tratar e resolver problemas complexos e adaptativos, enquanto produtiva e criativamente entregam produtos com o mais alto valor possível."*
> *(SCHWABER, SUTHERLAND, 2011, p. 3)*

De acordo com Schwaber e Sutherland, o Scrum pode resolver as dificuldades enfrentadas pelos métodos tradicionais e pode gerenciar e controlar o desenvolvimento do software. Como os processos ágeis são iterativos, as atividades de teste tem que seguir essas iterações e ser executado o mais rápido e de forma eficiente.

Para um teste ser executado dentro de um ambiente de desenvolvimento ágil é necessário que o teste esteja dentro do time de desenvolvimento. Uma boa prática para melhorar a velocidade de um time, ajudar na organização dos testes e unir a equipe de desenvolvimento é a automação dos testes.

Cohn (2009) criou uma maneira de visualizar como deve-se estruturar os testes automatizados como apresentado na figura 2, e todos esses níveis são essenciais para garantir que o software esteja funcionando e entregando o seu valor de negócio.

Figura 2 – Pirâmide de automação de testes
Fonte: Cohn (2009)

Segundo Cohn (2009) para seguir uma estratégia eficiente dos testes automatizados deve ser separado em três níveis. O primeiro nível de unidade na base da pirâmide é o teste de unidade que deve ser a base para uma boa estratégia dos testes por ser mais fácil de manter.

No topo possui os testes de interface, porém deve existir uma pequena quantidade de testes para evitar problemas de manutenção nos testes de interface e por ser mais caro de manter. Os testes de serviços ou testes de integração devem conter testes que preencham os testes que não foram realizados nos de unidade e de interface.

Por último no topo da pirâmide temos os testes manuais que como o nome próprio já diz, ele não será automatizado e explorará o sistema para tentar encontrar problemas que os demais testes não conseguiram encontrar.

1.2 Problema

O problema abordado é a implantação de um processo de teste automatizado em um processo ágil e será apresentado como se implanta teste automatizado em processos ágeis.

Com o fracasso no desenvolvimento de projetos de software, surgiu a necessidade da criação de um novo método, como uma alternativa aos métodos tradicionais. Essa nova abordagem para o desenvolvimento de software desperta cada vez mais interesse entre as organizações do mundo todo.

A tendência do desenvolvimento ágil de software é devido às mudanças constantes no desenvolvimento do software, as pressões por constantes inovações e as grandes mudanças nos ambientes de negócios. Assim, a expressão "Métodos Ágeis" está se tornando mais popular por ser simples e se adequar mais rapidamente as mudanças constantes.

Utilizando método ágil, as fases do desenvolvimento como a fase de teste, precisa se adaptar rapidamente as mudanças e a utilização da automação garante que a atividade de teste seja executada de forma mais rápida.

Segundo Collins (2012) muitas empresas estão investindo em testes automatizados para evitar defeitos no software e aumentar a eficácia dos testes durante o desenvolvimento. Porém testes automatizados aplicados em empresas que utilizam métodos ágeis precisam de colaboração de equipe e distribuição das tarefas de testes para conseguirem obter algum sucesso.

O método ágil Scrum que será o método utilizado não fala como é o processo de teste dentro de um desenvolvimento de software ágil, o modelo de maturidade ágil AMM, Patel (2009), será utilizado levando em consideração as áreas de processo chaves, para ajudar a criar um processo de testes automatizados em um ambiente ágil. Juntamente com o método de instanciação de processos, Dias (2010), para definir a visão de negócio da empresa e gerar uma arquitetura de processos a fim de identificar os principais gaps dentro da empresa.

O processo de testes definido deve possui uma maturidade a fim de garantir à empresa a qualidade no processo. Com o modelo de maturidade de testes é possível verificar qual nível a empresa atinge e garantir que o mesmo seja prioritário dentro da empresa.

1.3 Objetivo

O objetivo desse trabalho é propor uma estratégia de implantação de um processo de teste automatizado em um ambiente ágil utilizando o método Scrum.

Para elaboração da estratégia foram utilizados como modelo de referência:

- Modelo de Maturidade Ágil (AMM) o qual se avalia o nível de agilidade de uma empresa e se a mesma está apta para implantar processo ágil ou se já possui um processo ágil.

- SCRUM que é um dos métodos ágeis mais utilizados pelas empresas e é um framework que resolve problemas complexos e adaptativos entregando o mais alto valor possível. Conforme Schwaber; Sutherland (2011).

- O Modelo de maturidade de testes (TMM) que mostra o que é necessário para se atingir um nível de maturidade de testes nas empresas.

- Método de instanciação de processos proposto por Dias (2010) para definir a visão de negócio da empresa em termos da necessidade dela no processo ágil, gerando assim a arquitetura de processos.

Com a aplicação do método de instanciação de processos, utilizando como referência os modelos AMM e TMM será gerado uma metodologia onde será aplicada a estratégia de implantação de testes automatizados.

1.4 Justificativa

Com a competitividade do mercado cada vez maior, as empresas procuram cada vez mais práticas para se tornar mais eficiente, rápida e com mais qualidade. Dessa forma elas estão investindo cada vez mais em métodos de desenvolvimento ágil uma vez que o mesmo prega desenvolvimento mais rápido e com maior qualidade no processo de desenvolvimento.

Umas das técnicas do desenvolvimento ágil é o teste de software automatizado, e o mesmo possui solução para maximizar a cobertura dos testes dentro de um ambiente ágil, garantir que o teste faça parte de todo o desenvolvimento do projeto e que falhas sejam encontradas tardiamente.

Existem casos em que os testes automatizados foram aplicados nas empresas que possuíam o desenvolvimento ágil Scrum, como mostrou Collins no artigo Strategies for Agile Software Testing Automation, porém esses estudos mostram apenas técnicas de testes ágeis para serem incorporadas as empresas, porém não avaliam o processo atual da empresa e parte do pressuposto que o nível de agilidade é seguido a risco para que sua estratégia consiga ser implantada.

Dessa forma, nesse trabalho será apresentada uma estratégia de testes automatizados em ambiente ágil, utilizando de automação de testes ágeis uma vez que a automação é imprescindível para se atingir agilidade no desenvolvimento. Porém essa estratégia avalia os requisitos da empresa para avaliar o cenário atual, avalia o nível de maturidade ágil da empresa, pois dependendo do nível de maturidade da empresa a estratégia de implantação de testes automatizados não é viável e avalia também o nível de maturidade de testes para verificar o nível que a empresa está e avaliar o nível que ela chegará com a estratégia aplicada.

1.5 Estrutura

O trabalho será dividido segundo a estrutura a seguir:

O capítulo 1 presente, com o contexto inicial, o problema, o objetivo e a justificativa.

O capítulo 2 intitulado de "FUNDAMENTAÇÃO TEORICA" descreve o que é Scrum, as regras do Scrum, seus papéis, artefatos e eventos. Descreve também o que é teste, quais são os tipos de testes, o que é automação, a aplicação de testes automatizados em ambiente Scrum e apresenta o processo de automação de testes. Descreve o AMM, suas áreas de processos chave e o TMM com seus níveis detalhados.

O capítulo 3 intitulado "ESTRATÉGIA DE IMPLANTAÇÃO DE TESTES AUTOMATIZADOS EM AMBIENTE ÁGIL" relaciona os conceitos descritos nos capítulos anteriores e mostra a estratégia utilizada para implantar testes automatizados em ambiente ágil

O capítulo 4 intitulado "APLICAÇÃO" descreve como o método foi aplicado. Como a aplicação de testes automatizados no ambiente Scrum em uma organização e apresenta os resultados encontrados.

2 REVISÃO DA LITERATURA

Este capítulo apresenta os conceitos abordados neste trabalho. Apresenta a definição de teste de software, automação de testes e testes ágeis (seção 2.1), descreve o modelo de maturidade de testes TMM (seção 2.2), descreve o modelo de maturidade ágil AMM (seção 2.3), SCRUM (seção 1.4) e o método de instanciação de processos (seção 1.5).

2.1 TESTE DE SOFTWARE

2.1.1 Definição de teste de software

Muitos autores tentaram definir a atividade teste de software desde uma visão intuitiva de teste até uma visão formal como Myers (1979) e chegou a uma mesma definição que teste de software é o processo de executar o software de uma maneira controlada com o objetivo de avaliar se o mesmo se comporta conforme o especificado.

O teste determina se o software executado funcionou de acordo com suas especificações e seu objetivo é revelar falhas para que as mesmas sejam corrigidas antes da entrega final. A natureza da atividade de teste é destrutiva devido a ela aumentar a confiança do software expondo os problemas encontrados.

Conceitos como erro, defeito e falha precisam ser definidos para entendermos exatamente a diferença entre os mesmos na definição de teste de software. Segundo Muller e Fridenberg (2011) o erro é uma ação humana que produz um resultado incorreto, o defeito é o resultado de um erro encontrado quando o software ou o componente do software falha ao desempenhar a função esperada. E as falhas são as manifestações de um ou mais defeitos e afetam diretamente o usuário final do software.

2.1.2 Automação de testes

Segundo Collins (2012) no artigo Strategies for Agile Software Testing Automation: An Industrial Experience, automação de testes significa automatizar atividades de teste de software, incluindo o desenvolvimento dos roteiros de teste, a execução desses roteiros e a verificação dos resultados.

Os testes automatizados podem aumentar a eficiência ao reproduzir repetidamente diversas funcionalidades de um software. E para assegurar o custo da automação em um projeto é menor que o do teste manual, é necessário analisar os testes e definir uma estratégia de automação uma vez que requer esforço para conhecer as ferramentas de teste.

2.1.3 Testes ágeis

Crispin (2009) define teste ágil como testar um software com um plano para aprender sobre o software e deixar que as informações dos clientes orientem as atividades de teste utilizando os valores ágeis segundo o Manifesto ágil, como indivíduos e interações, software em funcionamento, colaboração com o cliente e responder as mudanças.

Para obtermos testes ágeis, a automação de testes se torna imprescindível, uma vez que a automação de teste é considerada o núcleo dos testes ágeis. A figura 3 mostra os quadrantes ágeis que mostra as diferenças entre os testes manuais e os testes automatizados.

Figura 3 – Quadrante de testes ágeis.

Fonte: adaptado CRISPIN (2009)

O quadrante um são os testes de tecnologia que dão respaldo a programação, como os testes unitários e os testes de componentes que auxiliam no entendimento do que o código deve realmente fazer. Os testes unitários são desenvolvidos geralmente em TDD (Test Driven Development), de focar na menor unidade do software que seja testável e na entrada e saída dos dados. Os testes de componentes devem tratar componentes isoladamente e verificar as interações entre as suas classes.

O quadrante dois são os testes de regras de negócio que dão respaldo a programação, como os testes de aceitação e testes funcionais. Esses testes devem possuir fácil entendimento aos envolvidos (stories) e sempre que possível devem ser automatizados. Geralmente são utilizados mapas mentais para escrever os cenários a serem testados e deve explicar as stories que fala o que é a funcionalidade, para que ela serve e coloca informações suficientes para o funcionamento do software e a criação dos testes de aceitação.

O quadrante 3 são os testes das regras de negócio que criticam o produto. Nesse quadrante ficam os testes exploratórios, de usabilidade e de aceitação do cliente. São os testes que complementam o quadrante um e o quadrante dois e são realizados após o desenvolvimento.

O quadrante quatro são os testes de tecnologia que criticam o produto, nesse quadrante ficam os testes de performance, de segurança, stress, etc e abrange todos os requisitos não funcionais.

2.2 MODELO DE MATURIDADE DE TESTES

Segundo o TMMI foundation, o modelo de maturidade de teste (TMM) é um modelo que foi criado para melhoria no processo de teste e possui cinco níveis como mostra na figura 4.

Nível	Objetivo
Inicial	Teste feito pela equipe de desenvolvimento
Fase de definição	-Desenvolver os objetivos do teste -Iniciar um processo de planejamento de teste -Institucionalizar técnicas e métodos básicos de teste
Integração	-Estabelecer uma organização de teste de software -Integrar o teste no ciclo de vida do software -Controlar e monitorar o processo de teste -Estabelecer um programa de treinamento
Gestão e medidas	-Estabelecer um programa amplo de revisão -Estabelecer um programa de medições de teste -Evoluir a qualidade do software
Otimização, prevenção de defeitos e controle de qualidade	-Aplicar processos de prevenção de defeitos -Controlar a qualidade -Otimizar o processo de teste

Figura 4 – Modelo de maturidade de testes
Fonte: Adaptado TMMI Foundation

O nível inicial ou nível um é o nível onde o processo é caótico, o desenvolvimento do software é feito com programação heroica e as pessoas não entendem a necessidade da qualidade e não compreendem o custo com a mesma.

O nível fase de definição ou nível dois, mais conhecido também como nível gerenciado já possui uma política e estratégia para os testes, possui um planejamento dos testes e um controle dos mesmos, existe a concepção e a execução dos testes e possui um ambiente separado apenas para a realização de testes.

O nível integração ou nível três, mais conhecido também como nível definido já possui uma organização dos testes, um programa de treinamento de testes aos

funcionários, o teste possui um ciclo de vida e integração, são realizados testes não funcionais e possui avaliação dos pares.

O nível gestão e medição ou nível quatro é onde ficam as medições dos testes e a evolução da qualidade de software constante.

O nível otimização ou nível cinco é de prevenção dos defeitos, com um processo de teste otimizado e um controle de qualidade mais maduro.

2.3 MODELO DE MATURIDADE ÁGIL

Será explicado o que é o AMM e quais são suas áreas de processo chave.

Método de desenvolvimento de software ágil é um conjunto de práticas e princípios para desenvolver software de maneira mais rápida, incrementalmente e garantindo a satisfação do cliente. Com isso surgiram vários métodos ágeis como Extreme Programming - XP e SCRUM que adotam os princípios de agilidade como definido no Manifesto Ágil como desenvolvimento frequente e iterativo, entregas continuas e antecipadas e simplicidade.

O modelo de maturidade ágil AMM foi criado e é utilizado para melhorar o desenvolvimento de software ágil e levar os princípios de software ágil como menor custo, satisfação do cliente e qualidade de software. O AMM possui cinco níveis de maturidade e cada nível tem o objetivo de ajudar a organização a melhorar suas atividades. A figura 5 apresenta os níveis do AMM.

Nível	KPA´s
Inicial	Não possui KPA por não possui processo de desenvolvimento ágil
Explorado	-Planejamento de projeto - Desenvolvimento guiado por cartões de história - Cliente disponível no mesmo ambiente -Introdução a TDD
Definido	-Gestão de relacionamento com o cliente - Software/produto entregue frequentemente -Interação mútua -Desenvolvimento guiado por testes - Interação e implementação -Padrões de código
Melhorado	-Gerenciamento de projetos -Ritmo sustentável -Time auto organizado -Ritmo sustentável -Planejamento de otimização de código
Maduro	-planejamento de projeto -desenvolvimento guiado por cartões de história

Figura 5 – Níveis do AMM

Fonte: adaptado de Patel (2009)

O nível inicial ou nível um é o nível inicial onde a empresa não possui um processo de desenvolvimento de software ágil definido, onde possui vários problemas como, por exemplo, cronogramas, comunicação, qualidade de software e custo alto no desenvolvimento.

O nível explorado ou nível dois já possui práticas de desenvolvimento mais estruturadas do que o nível um, e esse nível possui metas como melhorar o planejamento dos projetos, melhorar a definição dos requisitos ágeis, aumentar a colaboração, melhorar as práticas de desenvolvimento e têm como objetivo ajudar a equipe de desenvolvimento e o cliente a identificar e melhorar os problemas de planejamento, e aprender com o sucesso dos projetos anteriores e com seus fracassos. Este nível é atingido por uma avaliação do atual processo da empresa onde identifica fraquezas que ajudarão a equipe de desenvolvimento a resolver problemas de planejamento e entender os requisitos associados aos projetos.

As áreas de processos chaves para o nível dois são: planejamento dos projetos, desenvolvimento orientado a story cards (cartões de histórias), estar junto ao local do cliente e introdução ao TDD (desenvolvimento guiado por testes).

O nível definido ou nível três possui metas como a satisfação do cliente, a melhoria na comunicação, a qualidade do software e o aperfeiçoamento nas práticas de codificação. O objetivo desse nível é ajudar os desenvolvedores a identificar e melhorar os problemas relacionados ao cliente, problemas de codificação, testes, de lançamentos frequentes e padrões de codificação. Para atingir o nível três é feito um processo de avaliação do processo atual da empresa para identificar onde existem as fraquezas.
As áreas de processos chaves para o nível três são: gestão de relacionamento com o cliente, entregar software frequentemente, programação em par, interação mútua, TDD, implementação e interação e padrões de código.

O nível melhorado ou nível quatro é onde as empresas que estão nesse nível de maturidade já possuem uma posição para coletar medida detalhada durante o processo de desenvolvimento de software. O objetivo desse nível é fortalecer as equipes, gerenciar os projetos, avaliar os riscos e evitar desperdícios.
As áreas de processos chaves para o nível quatro são: gerenciamento de projeto, ritmo sustentável, auto-organização do time, avaliação dos riscos e planejamento do código.

O nível mantido ou nível cinco é onde as empresas melhoram continuamente seus processos através de feedback de ideias inovadoras para o processo, novas formas de testes e novas tecnologias. Empresas que passam do nível quatro para o nível cinco devem possuir uma riqueza nos dados e nas métricas para gerenciar melhor o processo. Os objetivos desse nível é a melhoria no contexto, o gerenciamento da incerteza e a prevenção dos defeitos.
As áreas de processos chaves para o nível cinco são: planejamento do projeto e o desenvolvimento guiado por story cards.

2.4 SCRUM

O Scrum, criado pelos americanos Jeff Sutherland e Ken Schwaber possui a seguinte definição de acordo com a versão em português do Guia do Scrum 2011.

> *"Um framework dentro do qual pessoas podem tratar e resolver problemas complexos e adaptativos, enquanto produtiva e criativamente entregam produtos com o mais alto valor possível."*
> *(SCHWABER, SUTHERLAND, 2011, p. 3)*

O método Scrum evoluiu ao longo dos anos tornando-se menos prescritivo, mais simples e focados nos valores ágeis como definido no Manifesto ágil de sofware. O método Scrum consiste em equipes associadas a papéis, eventos e artefatos como apresentado na tabela 1.

Papéis	Eventos	Artefatos
Product Owner (PO)	Planejamento da Sprint	Backlog do Produto
Equipe de Desenvolvimento	Daily Scrum	Backlog da Sprint
Scrum Master (SM)	Revisão da Sprint	Incremento
	Retrospectiva da Sprint	

Tabela 1 – Papéis, eventos e artefatos do SCRUM.
Fonte: Feito pelo autor

A figura 6 mostra a relação entre papéis, eventos e artefatos. Cada iteração no Scrum é chamada de Sprint e dura entre uma a quatro semanas. Antes de cada Sprint ser iniciada, o Product Owner transforma sua necessidade no Backlog do produto, que é uma lista priorizada pelos itens a serem desenvolvidos. O backlog é discutido pela equipe na reunião de planejamento da Sprint que sempre são divididas em duas partes. Na primeira parte a equipe de Desenvolvimento estima quantos itens do Backlog eles conseguem entregar e criam o backlog da Sprint e o objetivo da mesma junto ao Product Owner. Na segunda parte a equipe transforma o objetivo da Sprint em tarefas e cria um plano para alcançar o objetivo da Sprint. Esse plano é revisado pela equipe todos os dias na Diária (Daily Scrum) que é uma

reunião de horário fixo que dura no máximo 15 minutos para acompanhar o desenvolvimento da Sprint.

Ao final da Sprint, possui a reunião de Revisão da Sprint onde a equipe de desenvolvimento mostra a nova funcionalidade (Incremento) e discutem os próximos passos. Após a revisão da Sprint é realizada a Retrospectiva com todo o time Scrum para analisar o trabalho do time e identificar melhorias. O ciclo se repete a cada Sprint.

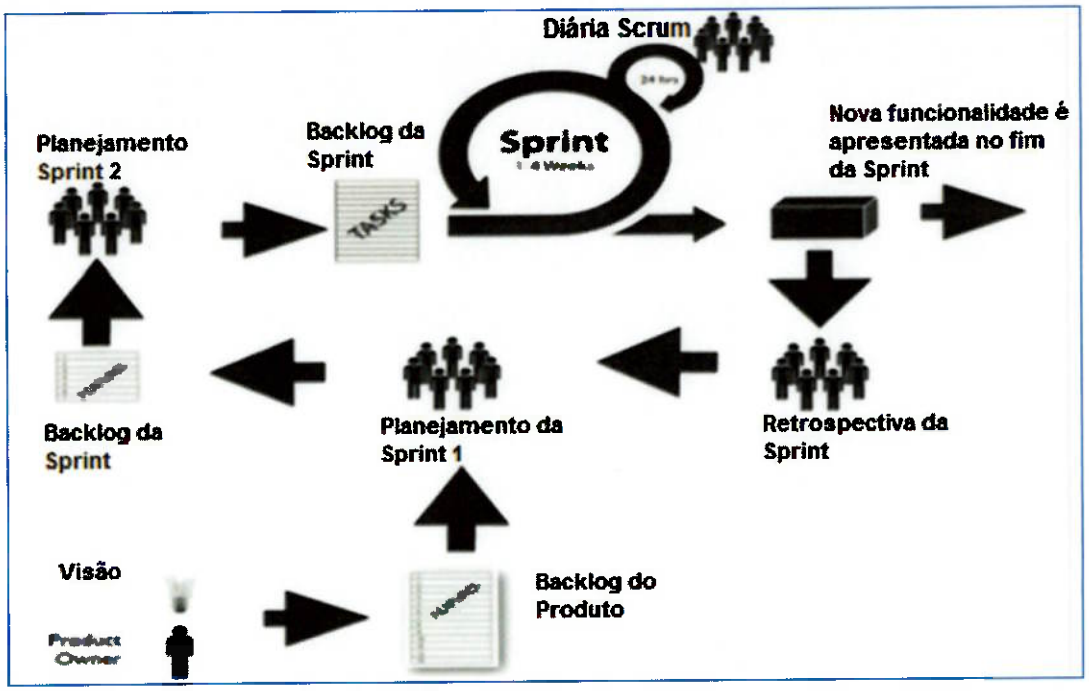


Figura 6 – Relação entre papéis, eventos e artefatos do Scrum

Fonte: Sutherland (2011)

2.4.1 Papéis do Scrum

A equipe Scrum possui três papéis: o Product Owner, o Scrum Master e a equipe de desenvolvimento.

O Product Owner coonhecido como PO é o responsável por manter o Backlog do Produto atualizado além de garantir o valor do trabalho realizado pela equipe de desenvolvimento. Para o PO obter sucesso é necessário que toda organização respeite suas decisões e ninguém da empresa tem permissão para falar com a equipe de desenvolvimento sobre prioridades.

A equipe de desenvolvimento é auto-organizável e ninguém, nem mesmo o Scrum Master diz como ela deve transformar o Backlog do produto em Incrementos de funcionalidade. A equipe deve possuir no mínimo três pessoas e no máximo nove, pois ter menos de três desenvolvedores na equipe reduz as interações entre os indivíduos e resulta em ganhos menores de produtividade e mais de nove desenvolvedores requer muita coordenação para um processo empírico como o Scrum.

O Scrum Master é responsável por garantir que o Scrum seja compreendido e executado, fazendo com que toda equipe Scrum atenda as regras do Scrum. O Scrum Master ajuda a encontrar técnicas para a gestão eficiente do Backlog do Prdoduto, comunica a visão, os objetivos e os itens do backlog, ensina a equipe de desenvolvimento a criar itens claros do backlog, pratica a agilidade e facilita os eventos do Scrum.

2.5 MÉTODO DE INSTANCIAÇÃO DE PROCESSOS

O método de instanciação de processos apresentado por Dias (2010), foi um método que adaptou processos de desenvolvimento para fábricas de software a partir de uma arquitetura de processos de referência, que é um conjunto de visões que são representadas por modelos conceituais de processo segundo Borsoi (2008), e essa adaptação foi denominada instanciação, uma vez que o processo foi considerado um objeto processo e toda vez que esse objeto processo é utilizado em um determinado ambiente ele se instancia.

O objeto processo proposto por Dias (2010) apresenta os elementos básicos de um modelo de arquitetura de uma fábrica de software e o objeto processo possui as mesmas informações que um objeto computacional como identidade, atributos e comportamentos.

Segundo Dias (2010), a instanciação é a parte do processo em que são detalhados e especializados os elementos dos objetos processos de acordo com as

características da empresa e do projeto que serão chamadas de requisitos de negócio.

Existem dois tipos de instanciação de processos, a instanciação tipo I e a instanciação tipo II. A instanciação tipo I é a instanciação de uma arquitetura de referência para uma arquitetura operacional e ela analisa os requisitos de negócio da empresa. A instanciação tipo II é de uma arquitetura operacional para uma arquitetura de projeto se software e é necessário analisar os requisitos do projeto de software em questão.

Neste trabalho falaremos apenas da instanciação do tipo I já que identificaremos os requisitos de negócio da empresa e não de um projeto de software específico.

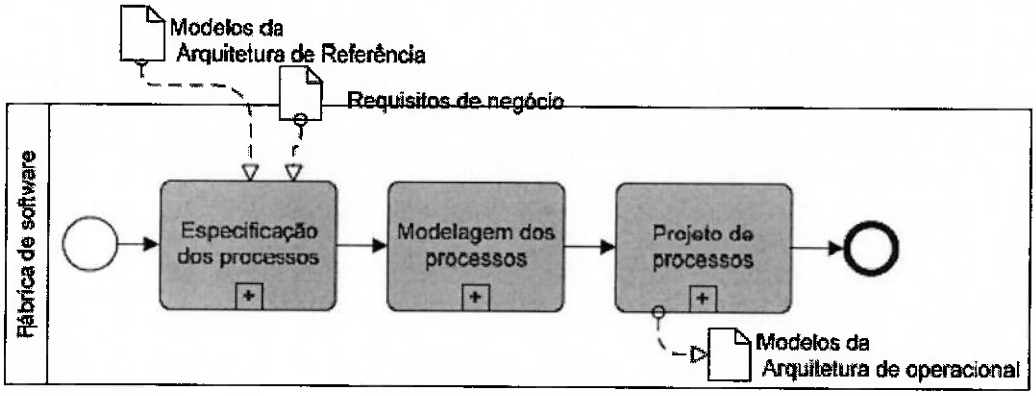


Figura 7 – Instanciação do tipo I.
Adaptado: Dias (2010)

A Figura 7 mostra as etapas do método de instanciação do tipo I. Nesta figura existem três etapas, a especificação dos processos que possui a entrada dos modelos de arquitetura de referência e dos requisitos de negócio da empresa, a modelagem dos processos e o projeto de processos que gera os modelos para a arquitetura operacional.

Na etapa de especificação dos processos são levantados todos os requisitos de negócio da empresa como as legislações, tecnologias, tipos de sistemas e as metas da empresa a nível estratégico. Para essa atividade de levantamento dos requisitos possuem questionários a serem preenchidos e com a resposta desses questionários

é possível identificar os candidatos a requisitos de negócio. Após a identificação dos requisitos devemos classificar de acordo com a Tabela 2 que possui visões com categorias para ajudar a classificar os requisitos e os candidatos a requisitos são exemplos.

Visões	Categorias	Candidatos a Requisitos
Empresa	Políticas	Candidato a Requisito ID1, ID2, ID9
	Procedimentos	
	Unidades organizacionais	
	Características da Empresa	
	Papéis corporativos	
	Agencias regulamentadoras	
	Stakeholders	
	Fornecedores	
	Mercado	
Engenharia	Tipos de software desenvolvidos	Candidato a requisito 4
	Recursos utilizados	
	Processos	
Informação	Padrões	
	Produtos de software	
Tecnologia	Tecnologias utilizadas	
Computação	Ferramentas	

Tabela 2 - Tabela de categorização de requisitos de negócio da fábrica por visões.

Fonte: Dias (2010)

No final da especificação de processos são gerados os requisitos de negócio da empresa. A atividade seguinte é o Mapeamento dos requisitos de negócio versus arquitetura de referência e com essa arquitetura é possível identificar as alterações nos elementos de processos de acordo com os requisitos de negócio.

Id dos Requisitos de negócio do projeto	Processos impactados	Elementos do processo impactados	Ação a ser executada

Tabela 3 - Matriz de requisitos de negócio e impacto nos objetos processos.
Fonte: Dias (2010)

Na tabela 3 os requisitos de negócio identificados devem ser inseridos na primeira coluna, na segunda coluna devem ser colocados os processos da arquitetura de referência, na terceira coluna devem ser identificados os processos que os requisitos de negócio impactam e na quarta coluna a ação a ser executada para adequar o objeto processo da arquitetura de referência ao requisito de negócio. As ações, segundo Dias (2010) são:

• Adicionados – quando devido ao requisito de negócio é necessário criar um elemento do objeto processo na arquitetura de processos de operacional, pois este não existia na arquitetura de processos referência;

• Removidos - quando devido ao requisito de negócio é necessário retirar um elemento do objeto processo que existia na arquitetura de processos de referência;

• Reduzidos - quando devido ao requisito de negócio é necessário diminuir escopo dos elementos de um objeto processo da arquitetura de processos de referência;

• Expandidos - quando devido ao requisito de negócio é necessário aumentar o escopo dos elementos de um objeto processos da arquitetura referência;

• Substituídos - quando devido ao requisito de negócio é necessário substituir algum elemento do objeto processo por outro já existente, por exemplo, a substituição de um documento da arquitetura de referência por um documento com o conteúdo exigido pelo cliente;

• Redefinidos - quando devido ao requisito de negócio é necessário redefinir os

elementos de um objeto processo.

Após a etapa de Mapeamento dos requisitos de negócio versus arquitetura de referência, vem à etapa de elaboração do documento de especificação dos processos. As atividades, os papéis e os artefatos estão nas figuras 8,9 e 10.

Nome da atividade:	**Nome da atividade que será descrita**
Ator:	Papel responsável pela execução da atividade
Tipo:	Tipo serve para organizar as atividades em grupos
Descrição:	Detalhamento sobre a definição atividade
Meta:	Meta a ser atingida durante a execução da atividade
Pré-condições:	Condições necessárias para início da atividade
Artefato entrada:	Artefato utilizado durante a execução da atividade
Tarefas:	Detalhamento passo a passo da atividade
Recursos:	Recursos necessários para execução da atividade
Habilidades:	Habilidades necessárias ao ator para executar a atividade
Artefato saída:	Artefato resultante após a execução das tarefas
Controle:	Restrição à realização da atividade, por exemplo, referentes a políticas ou legislação.
Métricas:	Medições que serão realizadas para avaliar o desempenho da atividade

Figura 8 - Diagrama descritivo da atividade.

Fonte: Dias(2010)

Nome	**Nome do papel**
Descrição	Detalhamento sobre o papel

Figura 9 - Diagrama descritivo do papel.

Fonte: Dias(2010)

Nome	**Nome do artefato**
Descrição	Detalhamento sobre o artefato
Tipo	Tipo que classifica os artefatos em grupos
Versão	Versão atual do artefato
Proprietário	Pessoa responsável pela criação do artefato
Política	As políticas que o artefato está sujeito
Modelo	*Template,* orientações ou critérios para produzir o artefato
Composto	Indicações de dos artefatos que compões o artefato descrito
Derivado	Indicação do artefato que origina o artefato descrito

Figura 10 - Diagrama descritivo do artefato.

Fonte: Dias (2010)

Após ter os requisitos de negócio especificados, os processos são modelados e a arquitetura operacional é gerada com as atividades análise e elaboração dos modelos estruturais, a análise e elaboração dos modelos comportamentais e a validação com o cliente.

3 MÉTODO DE TESTE AUTOMATIZADO EM AMBIENTE ÁGIL

Este capítulo está dividido em quatro seções. A seção 3.1 apresenta as considerações iniciais que motivaram a proposta do trabalho proposto na monografia: teste automatizado em ambiente ágil. A seção 3.2 descreve como o método de teste automatizado em ambiente ágil foi proposto, em qual referência ele foi baseado e o passo a passo do método. A seção 3.3 é a aplicação do método no cenário da empresa, para isso, apresenta o contexto atual da empresa. E também, é apresentada a proposta da estratégia de implantação de processo de teste automatizado no contexto ágil através da aplicação do método. E a seção 3.4 mostra os resultados da estratégia de implantação.

3.1 CONSIDERAÇÕES INICIAIS

Aplicar teste automatizado nas empresas de desenvolvimento que trabalham com agilidade é imprescindível, porém passa a ser um grande desafio quando não possui um processo de teste definido e integrado ao processo de desenvolvimento. Para isso, é necessário entender qual o nível de agilidade da empresa e identificar seus processos de negócio para aplicar uma estratégia de teste automatizado.

Com o objetivo de demonstrar um caso de estudo de uma aplicação da estratégia de implantação de teste automatizado em ambiente ágil, nesse capitulo será abordado o método de testes automatizado criado, utilizando o método de instanciação de processos com o modelo de maturidade AMM.

A metodologia proposta para a estratégia de implantação de teste automatizado em ambiente ágil é:

- Utilizar o método de instanciação de processos para a identificação dos requisitos de negócio da empresa, como os procedimentos, a organização, os papéis e as expectativas.
- Utilizar o modelo de referência AMM, modelo de maturidade ágil, para avaliar o nível de agilidade da empresa.
- Identificar o nível de maturidade de testes da empresa utilizando o TMM.
- Utilização das técnicas de testes automatizados sugeridas por Crispin (2009).

3.2 PROPOSTA PARA A ESTRATÉGIA DE IMPLANTAÇÃO DE TESTE AUTOMATIZADO EM AMBIENTE ÁGIL

A proposta para a criação de uma estratégia de implantação de teste automatizado em ambiente ágil foi baseada no método de instanciação de processos de Dias (2010) e no modelo de maturidade ágil, AMM, de Patel (2009).

3.2.1 Utilização do método de instanciação de processos para identificação dos requisitos de negócio

Como apresentado na Figura 11, para aplicar o Método de instanciação de processos para definir o processo de teste automatizado em ambiente ágil especificamente SCRUM, é necessário definir as necessidades da empresa extraindo os requisitos de negócios, que são as necessidades que a empresa identifica como o negócio principal para desenvolvimento de software, utilizando o método de instanciação de processos proposto por Dias (2010).

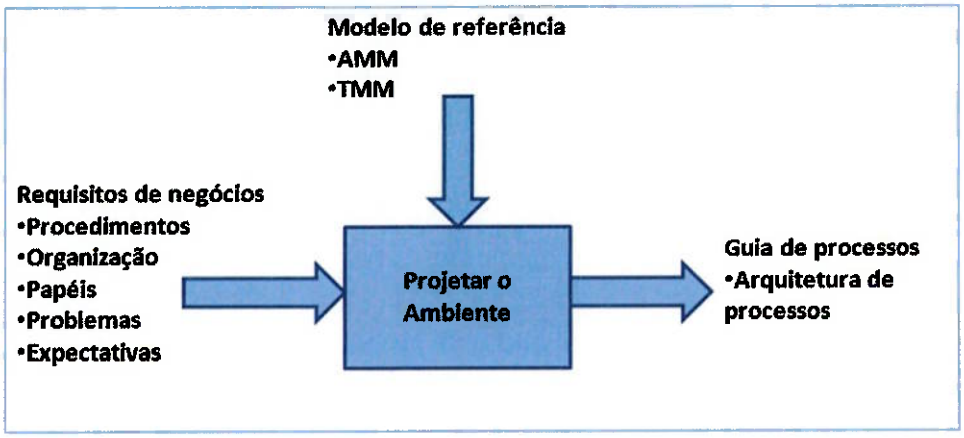


Figura 11 - Método de instanciação para gerar uma arquitetura de processos.
Fonte: Adaptado Dias (2010)

Com os requisitos de negócios gerados a partir da identificação de procedimentos, organização, papéis, políticas, problemas e expectativas, juntamente com o modelo de referência AMM com SCRUM o ambiente da empresa é projetado gerando assim uma arquitetura de processos e após a mesma pode ser implantada e avaliada.

De acordo com Dias (2010), para identificar os requisitos de negócio da empresa é necessário identificar as visões da empresa, identificando os procedimentos utilizados na empresa, ter uma visão geral da organização, quais são os papéis, os problemas existentes e a expectativa. A instanciação é a parte do processo onde são gerados os objetos processos de acordo com as características identificadas na empresa e do projeto e conforme Dias (2010) sugerem esses são os requisitos de negócio.

Os objetos processos identificados em cada empresa e projeto possuem atributos e operações. Porém para o objeto processo no método de instanciação, os artefatos são os atributos e as atividades são as operações. O artefato produz, altera ou utiliza uma atividade e é responsável por um papel, a atividade é realizada por um papel. E conforme sugerido por Dias (2010), fazer um diagrama estrutural do processo e da atividade é importante para verificar a consistência do objeto processo.

Após a identificação dos requisitos de negócio da empresa, é possível saber quais os problemas da empresa, qual a expectativa, como a empresa funciona, é possível ter uma visão geral dos papéis e dos seus procedimentos. Dessa forma, o modelo de referência AMM deve ser utilizado como complemento para projetar o ambiente desse processo e verificar se a empresa possui ou não agilidade para aplicar a estratégia de implantação de testes automatizados.

Com a identificação dos requisitos de negócio da empresa, é avaliado o nível de agilidade da mesma para verificar se existe um nível de agilidade e assim dar continuidade a metodologia criada.

3.2.2 Utilização do modelo de maturidade ágil AMM para identificação do nível de agilidade

Patel (2009) definiu um roteiro para identificação e melhoria do processo de desenvolvimento ágil de acordo com a figura 12. As principais características desse processo são a avaliação da adaptabilidade e adequação que é realizada pelos membros da equipe ágil e é útil durante o processo de implantação do AMM. O

objetivo desse processo é garantir ou identificar se uma organização possui ou não um desenvolvimento ágil. Caso a empresa não possua um desenvolvimento ágil, então a adaptabilidade e adequação recomenda o que precisa ser feito para ser uma organização de desenvolvimento ágil. No inicio da utilização do AMM os objetivos de negócio devem ser definidos pela equipe ágil e esses objetivos devem conduzir as atividades como a seleção das KPAs (área de processo chave), identificação do nível de maturidade e priorização das áreas de melhoria.

Após uma versão de avaliação AMM é realizada pela equipe ágil para identificar á área de melhoria e o nível de maturidade do processo de software. O plano de melhoria é identificado com base nas entradas fornecidas pelos questionários de avaliação de cada nível de maturidade das áreas de processo KPAs. Após a identificação das KPAs para cada nível de maturidade é feito um mapeamento dos melhores conhecimentos baseados em praticas ágeis.

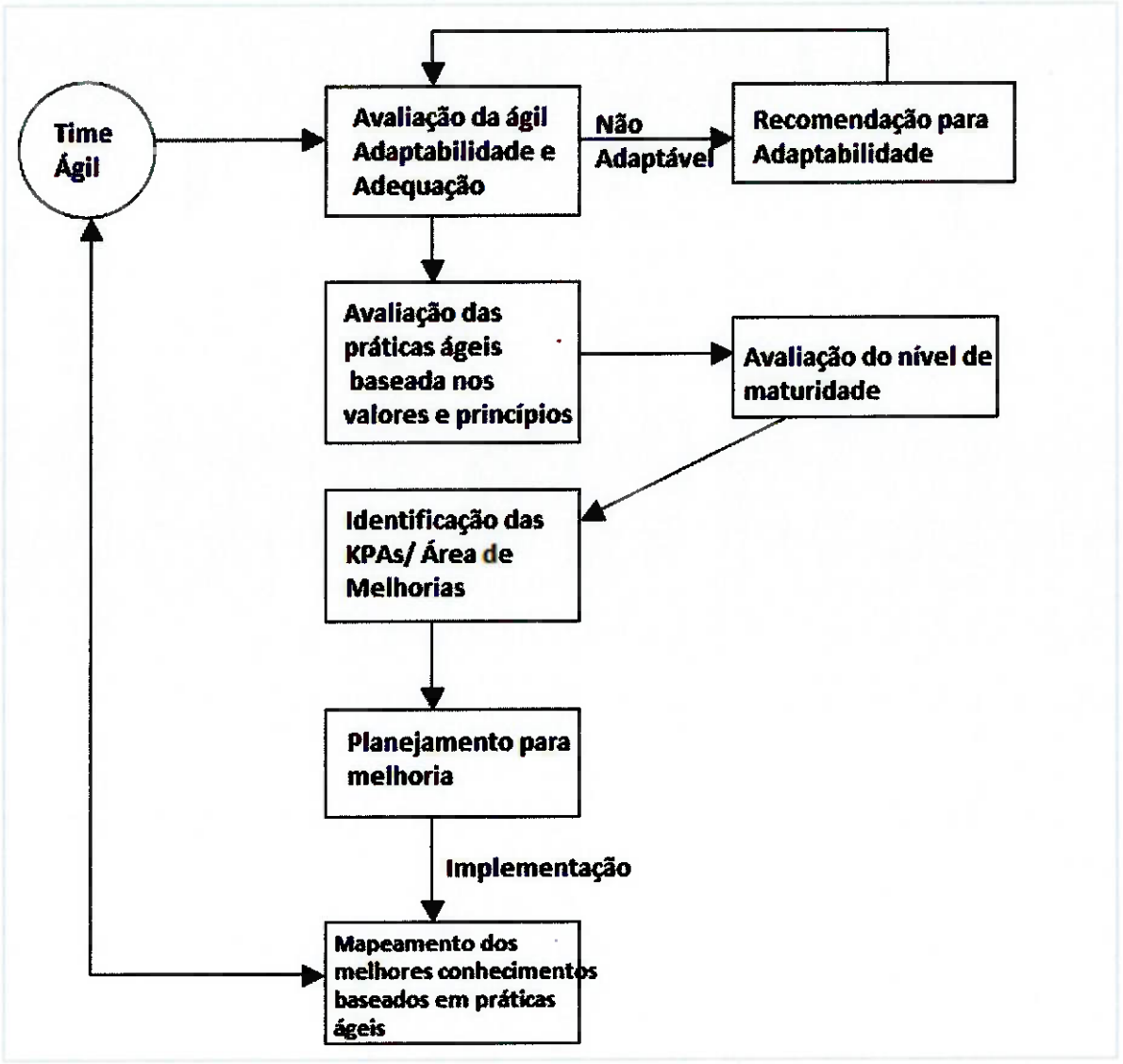


Figura 12 - Roteiro de identificação e melhoria do processo de desenvolvimento ágil

Fonte: Adaptado de Patel (2009)

A avaliação de adaptabilidade e adequação é baseada em um questionário que avalia os principais problemas no processo de desenvolvimento de software. O questionário apresentado na figura 9 é divido em cinco seções, são elas: Métodos de desenvolvimento de software utilizado ou que pretende utilizar, identificação dos problemas durante o desenvolvimento de software e a solução adotada para resolução dos problemas, relacionamento e disponibilidade do cliente, conhecimento sobre agilidade e trabalho em grupo dos gerentes e desenvolvedores e a avaliação do tamanho do projeto.

Questões Adaptabilidade	Possíveis Opções
Qual deles é o mais difícil problema de desenvolvimento de software que você teve ou está tentando resolver?	disponibilidade do cliente relacionamento com o cliente Entregar software em tempo real com todos os recursos requisitos variáveis mudança requisições deslizes no cronograma projeto cancelado Sistema com problema Taxas de defeito Característica falsa rica Rotatividade de pessoal Falta de pessoal qualificado Documentação excessiva de requisitos Alta rotação de empregados
Qual ciclo de vida de desenvolvimento (processo) é usada ou pretende usar no projeto piloto?	Desenvolvimento de software incremental Processo sequencial (Modo de queda de água) Prototipagem método de desenvolvimento de software processo evolutivo Processo de reutilização
Qual fator você deseja otimizar durante o projeto piloto?	produtividade orçamento mínimo Taxa de queima eficiente Entrega no prazo Batendo a estimativa Seguindo o plano Trabalhando com conhecimento incompleto Manipulação requisitos emergentes
Quantas vezes você tem cliente disponível no local do projeto?	No local do cliente contrato fixo Visite uma vez por semana contrato de preço Visite quando necessário Não está disponível em todos
Qual é o principal problema que você cara com o cliente ou que problemas que você está tentando resolver?	disponibilidade Requisitos variáveis (vez diferente) Pedir para entregar o produto rapidamente
Categoria do Cliente no sentido de domínio	especialista de domínio especialização em negócios analista de negócios noviço
Qual o método utilizado para apresentar requisitos ou vai usar no projeto piloto?	A documentação detalhada História de usuário (cartões de história) caso de uso outro
A estimativa é feita por?	revelador gestor de projeto rastreador projeto equipe de testes Equipe de garantia de qualidade
Técnica de estimação utilizada ou planejando usar?	estimativa passado pontos de função modelo COCOMO outro
O que você considera sobre relacionamento com o cliente?	não satisfeito satisfeito realmente impressionado
Você faz ou pretende fazer upfront concepção do projeto	Sim ou Não
O que é desenvolvedores mais importantes qualidade de trabalho	Capacidade de trabalhar em grupo (programação em pares) Alta capacidade individual (programação Solo)
Faça a sua equipa de gestão considere fazer coisa mais simples que poderia funcionar?	Sim ou Não
Gerente oferece ritmo sustentável (40 horas de trabalho)	Sim ou Não
Qual é a opinião ou atitude do gerente para a responsabilidade	Responsabilidade é atribuído ou dado para a equipe Responsabilidade aceita pela equipe
Como muitas vezes os gerentes fazem reunião ou pronto para fazer?	Diário levantar reunião reunião semanal Do quando necessário De modo nenhum
Qual é o tamanho do projeto-piloto?	Pequeno Médio Grande

Figura 13 - Roteiro de identificação e melhoria do processo de desenvolvimento ágil

Fonte: Adaptado Patel (2009)

A avaliação traz três resultados de acordo com as respostas fornecidas no modelo de adaptabilidade. O primeiro resultado recomenda adotar método ágil no seu

projeto piloto, o segundo resultado diz que está pronto para adotar um método ágil e o terceiro resultado diz que o projeto piloto não se adéqua a um método ágil, porém após o conhecimento de desenvolvimento de software ágil é possível aplicar a agilidade.

Após essa avaliação é feita uma avaliação das práticas ágeis e após a identificação das áreas de processos-chave KPAs para melhoria. O objetivo do método de avaliação é avaliar as práticas ágeis de software no desenvolvimento atual e identificar a área-chave do processo como oportunidade de melhoria a essa abordagem se obtêm através de questionários de avaliação como apresentado na Figura 9 e no AMM as KPAs identificam questões que devem ser abordadas para se atingir um nível de maturidade. O método AMM de avaliação é flexível e não envolve KPAs desnecessárias ou questionários. Uma das formas de realizar a avaliação do processo de software é a auto-avaliação, pois além de ser popular, possui um custo baixo boa acessibilidade e a apropriação do resultado. Dessa forma, como sugerido por Patel (2009), será utilizado o sistema de auto-avaliação para avaliar os processos de software, mas a avaliação automática também será considerada.

As respostas do questionário de avaliação são: Sim, parcialmente, Não e Não aplicável (N/A). A resposta parcialmente permite entender que a parte do processo de trabalho pode ter sido executada ou se aplicada ela não foi totalmente resolvida. A resposta N/A é aplicada quando não é possível implementar a pratica. A resposta Sim quer dizer que a prática é totalmente implementada e bem abordada no projeto e a resposta Não quer dizer que a prática não é dirigida para todos.

Com a utilização desses critérios, o percentual para cada KPA é calculado com a fórmula apresentada na figura 14. Onde o Yn é o número de respostas Sim, Pn é o número de respostas parcialmente, o Tn é o número total de questões e o NAn é o número de respostas N/A

$$\frac{\sum (Y_n) + \frac{1}{2} \sum (P_n) * 100}{\sum (T_n) - \sum (NA_n)}$$

Figura 14 – Fórmula para calcular o percentual de cada KPA.
Fonte: Patel (2009)

A partir do cálculo se obtêm a porcentagem e a partir da porcentagem é verificado o nível da KPA avaliada. A interpretação é feita de acordo com a figura 15 com a classificação do nível de capacidade da KPA avaliada, a porcentagem para a classificação e a descrição da classificação.

Classificação	Percentual	Descrição
Totalmente atingida	86% a 100%	Existe evidências de uma completa e sistemática abordagem para a realização plena e das práticas-chave definida na KPA avaliada. Não há deficiências significativas existentes na unidade da organização definida.
Largamente alcançado	51% a 85%	Existe evidências da abordagem e conquista significativa das práticas-chave definidas na KPA avaliada. Desempenho das práticas-chave pode variar em algumas áreas.
Parcialmente atingidos	16% a 50%	Existe evidências da abordagem e realização das práticas-chave definidas na KPA avaliada. Alguns aspectos da conquista pode ser imprevisível.
Não Realizado	51% a 85%	Existe pouca ou nenhuma evidência de realização das práticas-chave definidos na KPA avaliada.

Figura 15 – Classificação do nível da KPA avaliada.
Fonte: Adaptado de Patel (2009).

Uma vez que já analisamos e identificamos o nível da KPA dentro da área de desenvolvimento, para o método de automação de testes em ambiente ágil, será necessário que a empresa esteja no mínimo na classificação de "parcialmente atingido", pois nessa classificação já possui evidências de agilidade na área de desenvolvimento.

Com os requisitos de negócios levantados e o modelo de referência AMM utilizado para identificar o nível de agilidade da empresa, caso a empresa possua um nível de agilidade, é possível partir para o próximo passo onde é avaliado o nível de maturidade de testes da empresa.

3.2.3. Identificação do nível de maturidade de testes utilizando o TMM

Com a arquitetura de processos gerada e a identificação dos objetos processos definidos, o TMM, nível de maturidade de testes, será utilizado para verificar se a empresa possui um processo de teste, se possui artefatos no processo e assim atingir um nível de maturidade em um ambiente ágil.

O TMM possui cinco níveis de maturidade, porém para integrarmos a estratégia de implantação de testes automatizados é necessário avaliar os conceitos de cada nível na empresa e verificar se a mesma se encaixa em algum dos níveis conforme apresentado na figura 4 que apresenta o modelo de maturidade de testes.

Independente do nível de maturidade de testes da empresa é possível dar continuidade na estratégia e partir para o próximo passo. Pois o nível de teste avaliado na empresa, influencia diretamente nas ações de teste que ela precisa melhorar para possuir testes automatizados dentro do seu ambiente ágil.

3.2.4. Utilização das técnicas de testes automatizados

O último passo da estratégia é a utilização das técnicas de testes automatizados. Conforme sugerido por Crispin (2009) o teste está presente em todo o desenvolvimento e utilizando técnicas e ferramentas, é possível facilitar o desenvolvimento e automatizar o processo.

Para os testes de unidade existe a técnica TDD que visa o desenvolvimento do software guiado por testes de unidade, ou seja, cada trecho de código deve possui um teste. O TDD possui ferramentas no mercado como o PHPUnit.

Para os testes de regras de negócio ou de aceitação existe a técnica BDD que é o desenvolvimento guiado pelo comportamento, onde antes de começar o desenvolvimento do software, toda a equipe se reúne para montar os cenários de testes, entender todas as regras de negócio e após iniciar o desenvolvimento. O BDD possui ferramentas no mercado como o Cucumber.

Os testes exploratórios são realizados sem técnicas ou ferramentas e são feitos no final do processo para explorar o sistema de maneira geral e garantir que nenhuma falha seja encontrada em outras funcionalidades.

3.3 APLICAÇÃO DO MÉTODO

Para aplicar a estratégia de implantação de teste automatizado em ambiente ágil dentro de uma empresa, é necessário seguir os passos da estratégia apresentado na seção 3.2.

A aplicação foi realizada em uma organização que desenvolve produtos de Internet e é líder de mercado em sua área. A implantação do Scrum começou em 2009 em um projeto piloto e cresceu para abranger todas as equipes de desenvolvimento.

3.3.1 Aplicando o método de instanciação de processos para identificação dos requisitos de negócio

Para aplicarmos o método de instanciação de processo proposto por Dias (2010), a identificação dos requisitos de negócio da empresa é o primeiro passo para a estratégia de implantação de teste automatizado.

De acordo com a Tabela 4, os requisitos de negócio da empresa para a área de desenvolvimento foram identificados:

Requisitos de Negócios	Aplicação na empresa
Segmento da corporação	Internet
Objetivo da corporação	Ajudar profissionais a conquistar melhores oportunidades para sua carreira.
Como a corporação ganha dinheiro	Anunciando currículos de candidatos
Unidade organizacional da corporação	Nível estratégico: Diretor de desenvolvimento de software Nível gerencial : Gerência de produtos de desenvolvimento (PO's) Nível operacional: Departamento de desenvolvimento, Scrum Master e time de desenvolvimento
Papéis corporativos	Product Owner Scrum Master 1 analista de sistemas 2 desenvolvedores back-end 1 desenvolvedor front-end 1 analista de qualidade de software por time

Tabela 4 – Requisitos de negócio da área de desenvolvimento da empresa aplicada.

Fonte: Feito pelo autor.

Para ter uma visão geral da área de desenvolvimento, um organograma da área foi gerado conforme Figura 16 para se entender os papéis e os níveis da empresa avaliada e as descrições sobre a visão empresa para a área de desenvolvimento conforme Figura 17.

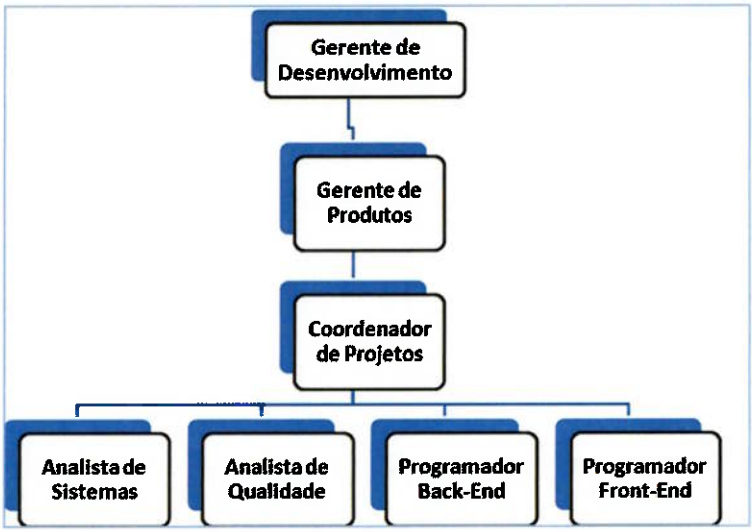


Figura 16 – Organograma da empresa avaliada.

Fonte: Feito pelo autor.

Área de desenvolvimento	Descrição
Stakeholders	Os stakeholders são os PO´s (gerente de produtos) da área de desenvolvimento, onde eles priorizam os objetivos e distribuem entre os times. C26
Produtos e serviços oferecidos pela fábrica de desenvolvimento	Todos os produtos para os candidatos anunciarem seus currículos nas vagas desejadas e todo fluxo para as empresas visualizarem e conseguirem contratar o candidato.
Quanto demora para desenvolver o serviço	Cada serviço é separado em requisitos de negócio, que se transforam em histórias e essas histórias são pontuadas. Demora média de 7 dias a Sprint para a entrega de um objetivo fechado durante a reunião de planejamento.
Priorização	Priorização é feito pelos PO´s e passada ao time de desenvolvimento para definição do prazo total do projeto.
Método utilizado	É utilizado o método SCRUM para acompanhar o desenvolvimento. São feitas reuniões diárias para levantar impedimentos e saber o andamento das tarefas de todos e após a finalização da sprint são feitos os eventos de review, retrospectiva e após são feitos os planejamentos de release.

Figura 17 – Visão área de desenvolvimento.

Fonte: Feito pelo autor.

Aplicando o Dias (2010) na empresa, é possível obter as regras de negócio e continuar a aplicação da estratégia.

3.3.2 Aplicando o AMM para avaliar o nível de maturidade ágil da empresa

A empresa utiliza o método Scrum como referência, porém foi aplicado o modelo de. referência AMM para identificar o nível de maturidade ágil da empresa.

Seguindo o roteiro de identificação e melhoria do processo de desenvolvimento ágil de Patel (2009), foi realizada a avaliação da ágil adaptabilidade e adequação, onde utilizando a figura 9 como referência, com perguntas sobre manter o cliente junto ao time dentro dos projetos, e por a empresa já possuir o método Scrum aplicado, pode dizer que a empresa está pronta para adotar um método ágil.

Após essa avaliação, é feita a avaliação das práticas ágeis e a identificação das áreas de processos-chave para melhoria. Foi realizada uma auto-avaliação utilizando como base as KPA´s, área de processo chave de teste, do AMM para

verificar se a empresa possui agilidade e se possui as KPA´s de teste, conforme apresentada na Tabela 5, e a própria equipe respondeu ao questionário.

KPA´s	Respostas (Sim, parcialmente, Não, não aplicável)
Existe planejamento do projeto	Sim
O desenvolvimento é feito com cartões de histórias	Não
O cliente fica disponivel no local	Sim
Possui TDD	Parcialmente
Possui gestão de relacionamento com o cliente	Sim
Produtos são entregues frequentemente	Sim
Possui Programação em Par	Parcialmente
Possui interação mútua	Parcialmente
Possui padrões de código	Parcialmente
Possui gerenciamento de projeto	Não aplicável
Possui ritmo sustentável	Sim
O time é auto organizado	Sim
Possui avaliação de risco	Sim
Possui planejamento de otimização do código	Sim

Tabela 5 – Auto-avaliação das práticas ágeis de acordo com as KPA´s.

Fonte: Feito pelo autor.

De acordo com essa avaliação feita pela equipe respondendo as questões da tabela 4, é realizado o cálculo para a classificação do nível da KPA identificada. O cálculo é realizado de acordo com a fórmula da figura 11 onde percebemos que a quantidade de respostas sim e parciais dividido pelo número total de questões, chega um uma classificação de largamente alcançado como apresentado na Figura 18.

Respostas	Numero de respostas	total questões do questionário (Tabela 4)	Total de respostas N/A	Classificação
Sim	8	14	1	85%
Parcialmente	4			
Não	1			
N/A	1			

Figura 18 – Avaliação das respostas da auto-avaliação.

Fonte: Feito pelo autor.

Foi verificado que a empresa possui um nível de agilidade porém não possui muitas áreas de processo de teste implementada, dessa forma é necessário implantar as KPA´s ao processo de teste automatizado para as KPA´s que foram respondidas na tabela 4 como "Não" e "parcialmente".

3.3.3 Aplicando o TMM para avaliar o nível de maturidade de testes da empresa

O processo atual da empresa já possui um nível de maturidade ágil, mas não possui um processo de teste definido. Como apresentado na figura 19, existe um processo onde o PO trabalha sozinho na elaboração do backlog do produto e só após o mesmo estar pronto que a equipe de desenvolvimento é envolvida.

A partir desse processo, o time de desenvolvimento tem todo acompanhamento do PO e realiza todos os eventos e papéis do Scrum, mas não possui um processo de testes mapeado.

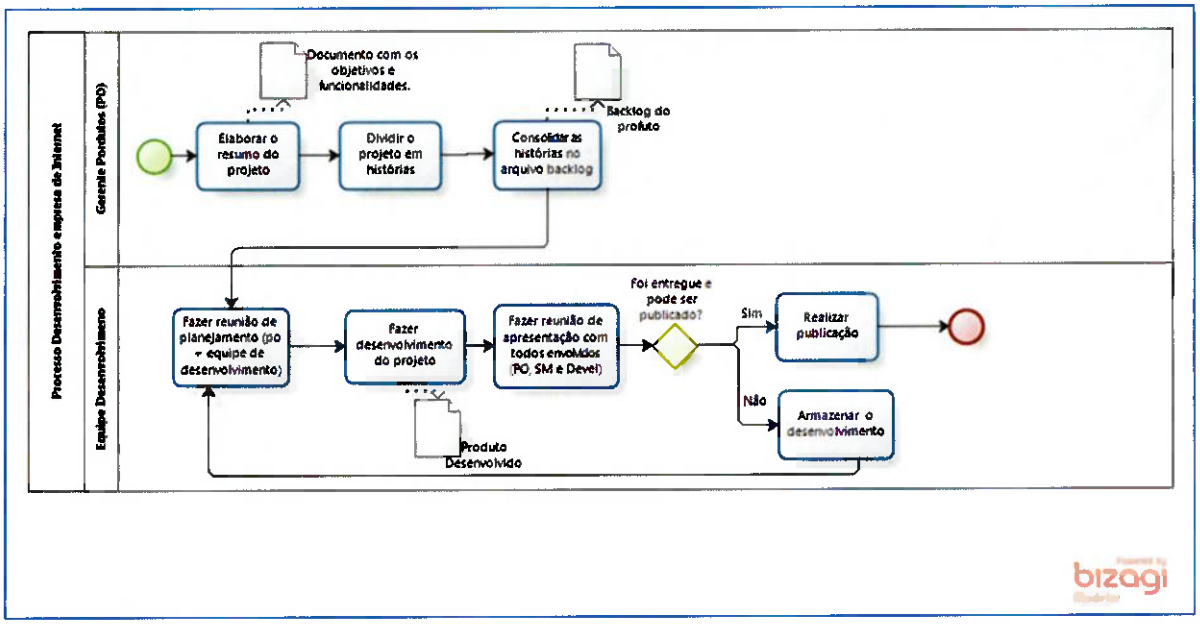


Figura 19 – Processo atual da empresa.

Fonte: Feito pelo autor.

No processo atual da empresa, não era possível identificar a fase de testes os artefatos de testes gerados e dessa forma podemos dizer que a empresa se

encontrava no nível inicial do TMM, onde a atividade de teste é um processo caótico, sem ferramentas e sem equipe treinada.

3.3.4 Aplicando a estratégia de implantação de testes automatizados utilizando técnicas de teste

Após a identificação dos requisitos de negócio da empresa, da identificação do nível de agilidade para descobrir as KPA´s a serem implementadas no novo processo de teste automatizado, descobrir que a empresa não possui um nível de maturidade de teste e nem ferramentas, é possível criar uma nova estratégia de implantação no processo da empresa.

Essa nova estratégia deve utilizar as KPA´s do processo de teste que antes não eram utilizadas. São elas:

- O desenvolvimento deve ser feito utilizando cartões de histórias;
- Deve utilizar a técnica TDD;
- Deve possuir programação em par;
- Deve possuir interação mútua;
- Deve possuir padrões de código.

Como apresentado na figura 19, as fases de requisitos e implementação são as fases do processo de desenvolvimento atual da empresa e analisando esse processo atual, é possível verificar que já existe uma fase de definição dos requisitos na qual a equipe de desenvolvimento não é envolvida, apenas o Product Owner juntamente a área de produtos e a fase de desenvolvimento onde o Product Owner juntamente a equipe de desenvolvimento definem o que precisa ser desenvolvido e realiza.

A fase de implementação atual não possui um processo de teste definido e a fase "fazer o desenvolvimento do projeto" não possui artefatos de entrada e nem artefatos de saída de testes.

Aplicando a estratégia de implantação de testes automatizados em ambiente ágil, após identificar as necessidades da empresa, é possível identificar os requisitos de negócio da empresa como já realizado, identificar o nível de agilidade da empresa e aplicar o TMM para a empresa atingir um nível de maturidade de testes na estratégia implantada, onde artefatos de entrada e saídas serão gerados a fim de melhorar o processo e atingir maior qualidade.

Com a aplicação do método de instanciação, juntamente com o processo atual da empresa, foram identificadas duas fases para o processo de desenvolvimento da empresa. Conforme apresentado na Figura 20, a fase requisitos foi mantida a fim dos papéis e a estrutura da empresa não ser modificada e a fase de implementação onde serão aplicadas as técnicas de testes, onde serão gerados artefatos de entrada e saída para assim atingir o nível de maturidade de testes e garantir uma qualidade maior ao processo.

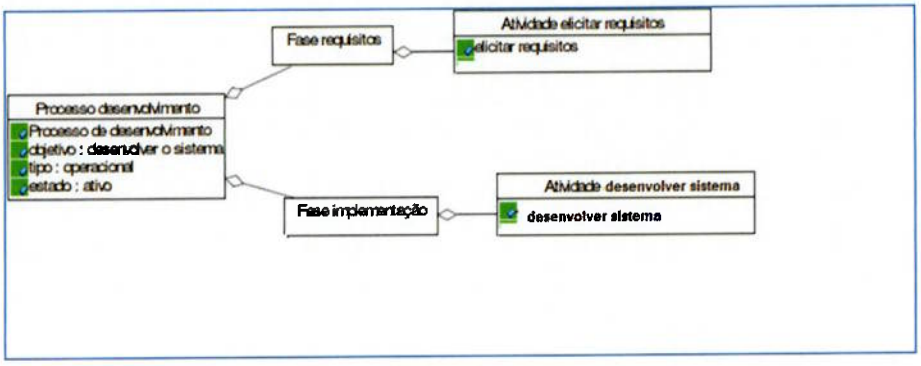


Figura 20 – Diagrama estrutural do processo e da atividade.
Fonte: Adaptado Dias (2009).

Elicitar requisitos é a atividade onde se obtêm as informações de um sistema ou de um produto e para essa atividade, como apresentado na figura 21, têm-se os papéis de PO e time de desenvolvimento trabalhando juntos nas tarefas de escrever as histórias do usuário e definir os critérios de aceite de cada história. Esses passos são de acordo com o processo AMM juntamente com o Scrum.

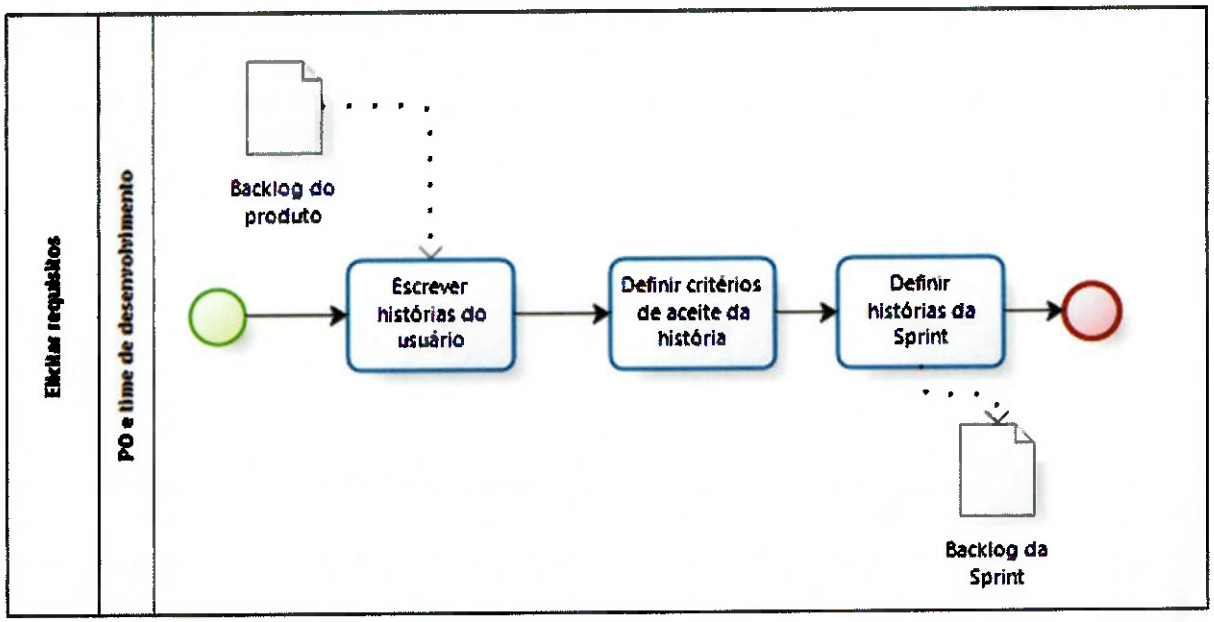


Figura 21 – Diagrama BPMN do detalhamento da atividade elicitar requisitos.
Fonte: Feito pelo autor

O processo de elicitar requisitos se inicia quando existe uma demanda para desenvolvimento de um produto e o cliente escreve um documento junto ao PO com as informações básicas ou características do que o produto precisa ter e quais os requisitos para atender sua necessidade, chamado de Backlog do Produto.

Após, o PO, que é o representante do produto, se reúne junto ao time de desenvolvimento para escrever as histórias do usuário e complementar o backlog do produto. Nessa fase e é importante o time estar presente, pois desde o começo já começam os questionamentos e a qualquer momento o cliente pode ser envolvido para esclarecer as dúvidas geradas caso o PO não consiga responder.

Após as histórias do usuário serem escritas, os critérios de aceite de cada história deve ser definido para entender os requisitos de cada história e pode entender sua complexidade. Quando as histórias e os critérios de aceite já estão escritos, então é estimado e definido o que entrará na Sprint para desenvolvimento e sai um documento chamado de Backlog da Sprint, que possui os itens a serem desenvolvidos junto com o objetivo da Sprint que foi definido juntamente entre PO e time e o tempo da Sprint que varia entre cinco e quinze dias.

Os itens do Backlog da Sprint são inseridos em um quadro de tarefas da Sprint, como apresentado na figura 22 e acompanhados na próxima atividade de desenvolver o sistema.

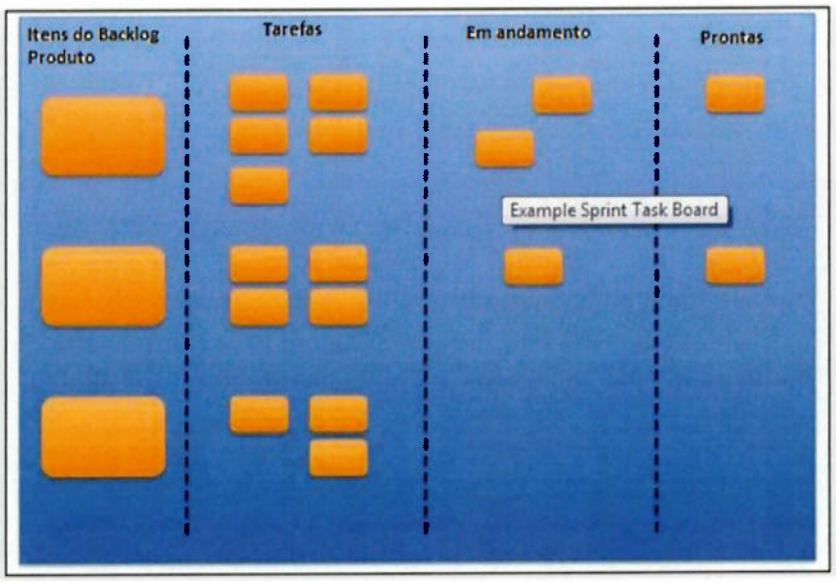


Figura 22: Exemplo quadro de tarefas da Sprint
Fonte: Adaptado de Scrum Institute

A atividade de desenvolver o sistema da fase de implementação é a fase onde se inicia a Sprint com o Backlog da Sprint que possui as histórias e o objetivo, e possui um tempo de Sprint para atingir aquele objetivo no tempo definido.

A figura 23 apresenta as tarefas dentro da atividade de desenvolvimento focada nos testes. Para que o processo de teste automatizado dentro de um ambiente ágil siga o processo de agilidade, o teste não é uma fase separada da fase de desenvolvimento e sim uma fase que inicia o processo de desenvolvimento, se mantém durante todo o processo e finaliza o mesmo conforme Collins (2012) e Crispin (2009)

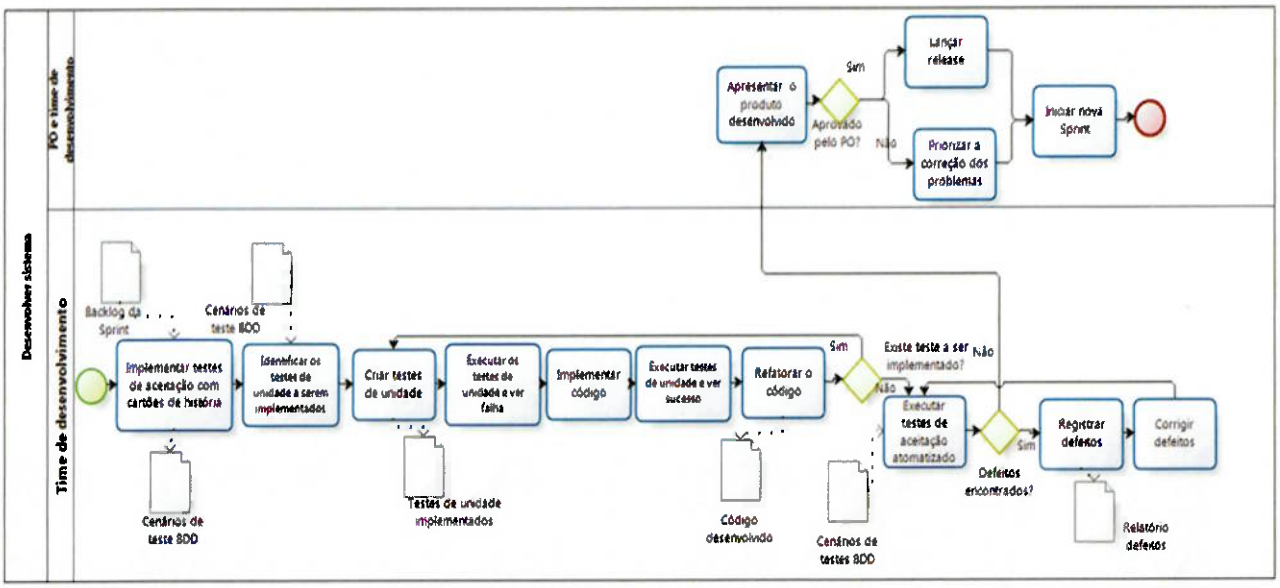


Figura 23 – Diagrama BPMN do detalhamento da atividade desenvolver sistema.
Fonte: Feito pelo autor.

A atividade de desenvolver sistema se inicia com o time de desenvolvimento possuindo o artefato Backlog da Sprint como entrada que foi definido na atividade anterior elicitar requisitos e gerado o quadro com as tarefas da Sprint.

A primeira tarefa da atividade de desenvolver o sistema é "Implementar os testes de aceitação com cartões de história" como sugerido por Patel (2009), essa tarefa é feita por todos integrantes do time de desenvolvimento para entender exatamente o que precisa ser desenvolvido e listar os cenários de teste em conjunto. Para aplicar essa tarefa, como recomendado por Crispin (2009) será utilizada a técnica BDD – Desenvolvimento guiado pelo comportamento.

Na técnica BDD, como apresentado na figura 24, a funcionalidade é a história definida e a partir da história junto as características da história, são descritos os cenários de teste com as condições necessárias de validação de acordo com os critérios de aceite definidos no Backlog do Produto.

Após escrever os casos de teste no formato de BDD, o mesmo pode ser aplicado numa ferramenta BDD, nesse contexto a ferramenta mais conhecida é o Cucumber que é uma ferramenta onde se descreve os cenários de teste em um formato de texto, o próximo passo é gerar os passos do teste escrito na linguagem de

programação definida para implementação e após ver o teste falha já que existe apenas a classe implementada e não os métodos e funções. O teste será escrito no formato da linguagem selecionada e o teste de aceitação só passará quando os testes forem implementados numa próxima fase.

```
#language: pt-br

Funcionalidade: Adição
    Para evitar enganos
    Enquanto alguém com dificuldades em matemática
    Eu gostaria de facilitar a soma de dois números

Cenário: Adicionar dois números
    Dado que informei "50" para a calculadora
    E que informei "70" para a calculadora
    Quando Eu pressionar "Add"
    Então o resultado deverá ser "120".

Cenário: Adicionar dois números negativos
    Dado que informei "-27" para a calculadora
    E que informei "-48" para a calculadora
    Quando Eu pressionar "Add"
    Então o resultado deverá ser "-75".
```

Figura 24 – Exemplo de cenário escrito na técnica BDD.

Fonte: Elemar (2012)

Após a tarefa "implementar testes de aceitação com cartões de história" é gerado o artefato Cenários de teste em BDD e o mesmo é utilizado como entrada para a próxima tarefa que é "identificar os testes de unidade a serem implementados". Essa tarefa analisa o documento de cenários gerado e verifica quais situações são passíveis de testes de unidade, uma vez que o teste de unidade valida apenas à unidade de código e não dependências de classes e métodos.

Com os testes definidos e listados, inicia-se a tarefa "criar testes de unidade" que será implementada utilizando a técnica TDD – Desenvolvimento guiado por testes, também recomendado por Crispin (2009) como técnica de agilidade de testes.

Com a tarefa de "criar testes de unidade" é gerado um artefato "testes de unidade implementados" onde todos os testes já foram escritos, mas não executados uma vez que o código não foi implementado. Com os testes de unidade criado e sem o código implementado, a técnica TDD recomenda que se execute o teste a fim de ver

programação definida para implementação e após ver o teste falha já que existe apenas a classe implementada e não os métodos e funções. O teste será escrito no formato da linguagem selecionada e o teste de aceitação só passará quando os testes forem implementados numa próxima fase.

```
#language: pt-br

Funcionalidade: Adição
    Para evitar enganos
    Enquanto alguém com dificuldades em matemática
    Eu gostaria de facilitar a soma de dois números

Cenário: Adicionar dois números
    Dado que informei "50" para a calculadora
    E que informei "70" para a calculadora
    Quando Eu pressionar "Add"
    Então o resultado deverá ser "120".

Cenário: Adicionar dois números negativos
    Dado que informei "-27" para a calculadora
    E que informei "-48" para a calculadora
    Quando Eu pressionar "Add"
    Então o resultado deverá ser "-75".
```

Figura 24 – Exemplo de cenário escrito na técnica BDD.
Fonte: Elemar (2012)

Após a tarefa "implementar testes de aceitação com cartões de história" é gerado o artefato Cenários de teste em BDD e o mesmo é utilizado como entrada para a próxima tarefa que é "identificar os testes de unidade a serem implementados". Essa tarefa analisa o documento de cenários gerado e verifica quais situações são passíveis de testes de unidade, uma vez que o teste de unidade valida apenas à unidade de código e não dependências de classes e métodos.

Com os testes definidos e listados, inicia-se a tarefa "criar testes de unidade" que será implementada utilizando a técnica TDD – Desenvolvimento guiado por testes, também recomendado por Crispin (2009) como técnica de agilidade de testes.

Com a tarefa de "criar testes de unidade" é gerado um artefato "testes de unidade implementados" onde todos os testes já foram escritos, mas não executados uma vez que o código não foi implementado. Com os testes de unidade criado e sem o código implementado, a técnica TDD recomenda que se execute o teste a fim de ver

o mesmo falhar, como apresentado na tarefa "Executar os testes de unidade e ver falha" e após partir para a tarefa de "implementar o código".

Na tarefa de "implementar do código" será implementado o código do sistema para fazer os testes de unidade passarem. Continuando com a técnica TDD o próximo passo será "executar os testes de unidade e ver sucesso", onde o código já foi implementado a fim de fazer o teste de unidade passar.

Com o sucesso na execução do teste de unidade, continuando a técnica de TDD, deve ter a tarefa "refatorar o código", onde o código deve ser refatorado para acabar com duplicidade de código e melhorar nomenclaturas ou até mesmo iterações e com essa tarefa sai o artefato "Código desenvolvido" com os testes de unidade executados.

Com a finalização do desenvolvimento utilizando a técnica TDD, existe um fluxo de decisão no método de automação de testes, onde é questionado se possui mais algum teste de unidade a ser implementado para garantir uma melhor segurança do código implementado. Se a resposta para esse fluxo de decisão for sim, volta-se para a tarefa "criar testes de unidade" e inicia-se o processo TDD novamente. Uma vez que a resposta para i fluxo de decisão foi "Não", entra o artefato "cenários de teste BDD" para a execução da tarefa "executar testes de aceitação automatizado".

A tarefa "executar testes de aceitação automatizado" executa automaticamente os cenários definidos no inicio da atividade de desenvolvimento e gerado na ferramenta Cucumber que foi a escolhida para a implementação. Com os testes executados e os critérios de aceite da história validados, possui outro fluxo de decisão que validará se defeitos foram ou não encontrados.

Caso defeitos tenham sido encontrados no sistema, inicia-se a tarefa de "registrar defeitos" a fim de ficar visível a todos os problemas encontrado gerando então o artefato "relatório de defeitos" para ser possível reproduzir os mesmos após a correção e possuir dados para métricas de testes que mostram o quão efetivo está sendo o método de testes, uma vez que os testes ágeis se iniciam no inicio do processo de desenvolvimento com técnicas de testes nesse processo.

Com os defeitos registrados, a tarefa "corrigir defeitos" é iniciada e após volta novamente para a tarefa "executar testes de aceitação automatizados" onde todos os testes no formato BDD são executados e validados se os critérios de aceite foram atendidos. Caso não existam mais defeitos encontrados, a tarefa "apresentar o produto desenvolvido" é realizada.

A tarefa "apresentar o produto desenvolvido" é realizada pelo PO e time de desenvolvimento, uma vez que a história só é declarada como feita com a validação final do PO. Caso o PO aprove as histórias, a release do produto é lançada e uma nova Sprint é iniciada. Caso o PO não aprove as histórias, será realizada a atividade de "priorizar a correção dos problemas" para após iniciar a atividade "iniciar nova Sprint" e finalizar a fase de desenvolvimento.

3.4 RESULTADOS

Com a estratégia proposta de implantar automação de testes em um ambiente ágil, o teste foi incorporado ao processo de desenvolvimento, as KPA´s do processo de teste foram implementadas e um nível de maturidade de testes foi atingido por possui um processo organizado e possui artefatos.

A estratégia fez com que a fase de testes se iniciasse desde o começo do processo de desenvolvimento e devido a possuir uma estratégia e uma organização dos testes, os testes possuem um ciclo de vida e uma integração, são realizados testes tantos automatizados quanto manuais e existe artefatos de entrada e saída como por exemplo, a saída dos cenários de teste e o relatório de defeitos.

Dessa forma podemos dizer que a empresa atingiu um nível de maturidade de testes passando do nível inicial ou nível um, para o nível fase de definição ou nível dois, onde possui um processo de planejamento de testes e utiliza técnicas de teste.

Possuindo um processo de teste definido dentro do ambiente ágil, especificamente o Scrum que já era utilizado na empresa, o envolvimento de todos os papéis são realizados desde o inicio do método e os eventos do Scrum como reunião de

planejamento, diárias, reviews e retrospectivas são mais claras para todos os participantes, pois não existem separações de papeis dentro do time de desenvolvimento e o PO está envolvido em todas as fases do processo.

Utilizando a estratégia de automação de testes em ambiente ágil foi possível observar uma maior multidisciplinaridade dentro da equipe de desenvolvimento, uma vez que se inicia o processo escrevendo testes de aceitação para depois começar o desenvolvimento e isso mostrou uma aproximação e envolvimento da equipe. E a importância dos testes passou a ser enxergada por todo o time desde o inicio do processo.

A Tabela 6 apresenta o cenário da empresa anteriormente e após a aplicação da estratégia de automação de testes utilizando o método de instanciação juntamente com o modelo de referência AMM que identificou as KPA´s a serem implantadas e o TMM para se atingir um nível de maturidade que não existia e possuir artefatos.

Cenário Anterior	Cenário Atual
Existe planejamento do projeto	Continua o planejamento
O desenvolvimento não era feito com cartões de histórias	Tornou-se a primeira fase do processo de desenvolvimento
O cliente fica disponivel no local	Cliente continua disponivel no local
Possui TDD parcialmente	TDD é a técnica incorporada ao processo de desenvolvimento e passou a ser utilizado sempre
Possui gestão de relacionamento com o cliente	Continua com a gestão
Produtos são entregues frequentemente	Continua entregando produtos frequentemente conforme os valores de agilidade
Possui Programação em Par parcialmente	Todo processo é realizado em par a fim de compartilhar os conhecimentos, facilitar e agilizar o desenvolvimento e todos estarem interados de tudo o que acontece no processo.
Possui interação mútua parcialmente	Possui interação mútua, a comunicação melhorou por todos estarem participando desde o inicio do processo
Possui padrões de código parcialmente	Possui padrões de código, principalmente pela programação em par onde um inspeciona o código do outro ao mesmo tempo e evita retrabalho ou desperdicio no fim do processo
Possui ritmo sustentável	Continua com ritmo sustentável que a agilidade prega
O time é auto organizado	Continua com time auto organizado
Possui avaliação de risco	Continua com a avaliação de risco pela equipe de desenvolvimento juntamente ao PO e SM
Possui planejamento de otimização do código	Continua

Tabela 6 – Resultados da aplicação da estratégia de implantação de teste automatizado.

Fonte: Feito pelo autor.

3.5 CONSIDERAÇÕES DO CAPÍTULO

Neste capítulo foi apresentada a estratégia de automação de testes em um ambiente ágil. O conhecimento dos processos da empresa e do processo de desenvolvimento é importante para a utilização da estratégia, pois possui muitas variáveis a serem avaliadas e muitas decisões a serem tomadas e estas precisam ser embasadas em modelos já existentes e estudados.

O modelo proposto abordou as perspectivas tratadas pelos outros modelos além de oferecer como diferencial as seguintes características:

- Associa KPA´s do AMM ou as áreas de processo de teste que mostram o que uma empresa precisa para ser considera ágil e o método considera que ela deve estar classificada no mínimo como "parcialmente atingido" pois nesse nível já possui uma evidência clara de agilidade.
- Com o AMM foi possível identificar KPA´s a serem implantadas no processo de teste.
- Uma vez que a empresa possui um nível de agilidade, podemos inserir métodos como o Scrum para ajudar no processo de desenvolvimento e separar papéis e incorporar eventos dentro do método.
- Utilizando o método de instanciação de processos foi possível identificar a necessidade da empresa, identificar duas fases existentes como as fases requisitos e implementação e as atividades elicitar requisitos e desenvolver o sistema, sendo que as atividades possuem entradas e saídas de artefatos.
- Utilizando o TMM para avaliar que a empresa não possuía nível de maturidade e aplicando essa estratégia foi possível atingir um nível de maturidade de teste e ter um processo existente.

A estratégia de implantação de teste automatizado foi elaborado e fundamentado no método de instanciação de processos juntamente ao AMM, utilizou o TMM para gerar artefatos e trazer um nível de maturidade ao processo de teste da empresa e o

Scrum como método ágil dentro do AMM e as técnicas de testes ágeis, desta forma, a estratégia criada pode ser aplicada e instanciada em empresas que possuem um nível de maturidade ágil definida, que consigam identificar seu objeto processo e aplicar o método inserindo as técnicas de testes ágeis, independente do tamanho do projeto ou do tamanho da empresa.

4 CONSIDERAÇÕES FINAIS

Neste capítulo serão apresentadas as conclusões da aplicação do método proposto para a definição de uma estratégia de implantação de processo de teste automatizado em ambiente ágil, desenvolvida com esse trabalho e, também a proposta de trabalhos futuros que poderão ser realizados a partir desse estudo.

4.1 CONCLUSÃO

Com as metodologias de pesquisas como o AMM, TMM, Scrum, método de instanciação de processos e técnicas de teste ágil foi identificado que o teste estava presente no processo de desenvolvimento, porém não possuía uma metodologia aplicada e não estava incorporado em todo o processo de desenvolvimento conforme a agilidade define.

A identificação das necessidades que a empresa identifica como negócio principal para o seu desenvolvimento de software e o método de instanciação de processos foi utilizado gerando um objeto processo, com suas fases e atividades.

Após a identificação dos requisitos de negócio, a identificação do nível de maturidade ágil de uma empresa é importante para verificar se a mesma está apta a incorporar a estratégia criada, e com o estudo foi possível verificar que sem um nível de agilidade não é possível aplicar a estratégia implantada.

O método proposto que defini uma estratégia de implantação de automação de testes em ambiente ágil possui técnicas baseadas no quadrante ágil que prega testes desde o começo do desenvolvimento e para isso ser possível é necessário implantar essa cultura dentro do time, como a técnica BDD que inicia o processo de desenvolvimento e une todo o time a fim de entender e listar os cenários de teste para desenvolver de uma forma mais clara e ágil.

Durante o desenvolvimento do sistema, outras técnicas de testes são utilizadas como o TDD que inicia com o teste de unidade antes do desenvolvimento, incluindo uma cultura de testes e da importância dos testes dentro do time Scrum.

A estratégia desenvolvida, baseado em várias literaturas e levando em consideração os autores estudados, é concluído que o time incorpora a cultura de testes e prevê que testes desde o inicio do desenvolvimento evita falhas futuras no sistema uma vez que falhas encontradas tardiamente geram mais tempo para correção e é mais custoso.

Uma vez que os testes estão automatizados, já que os mesmos são realizados e implementados desde o começo, todo do time Scrum está focado no desenvolvimento, não existindo separação de papéis dentro de um time e o teste passa a ser importante desde o começo para o time e não segrega times onde o time de testes só encontra defeitos em um sistema e não ajuda no desenvolvimento, caracterizando a agilidade.

4.2 TRABALHOS FUTUROS

Com este trabalho podem ser realizados os seguintes trabalhos futuros:

- Aplicar a estratégia de implantar automação de testes em um ambiente ágil e extrair resultados como a diminuição de defeitos no final do processo de desenvolvimento a partir de relatórios de defeitos gerados como artefatos da estratégia.

- Atingir um melhor nível de maturidade de testes (TMM), como o nível quatro que é o nível de gestão e medição, onde é necessário incorporar ao processo artefatos que gerem uma medição melhor dos testes.

REFERÊNCIAS

About International Scrum Institute - How can we help you?. Disponível em <http://www.scrum-institute.org/Scrum_Master_Accredited_Certification_Program.php>. Acessado em novembro de 2014.

BDD na prática – parte 2 – Reaproveitando código | Elemar DEV. Disponível em <http://elemarjr.net/2012/04/11/bdd-na-prtica-parte-2-reaproveitando-cdigo/>. Acessado em novembro de 2014.

BECK, K.; GRENNING, J. At all. **Manifesto para Desenvolvimento Ágil de Software**. Disponível em: <http://www.agilemanifesto.org/iso/ptbr/> Acessado em agosto de 2014.

COHN, M. Succeding with Agile. **Software Development using Scrum.** Addison-Wesley, 2009. 465 p.

COLLINS, E., DIAS-NETO, A, LUCENA JR, V. **Strategies for Agile Software Testing Automation: An Industrial Experience**. IEEE, 2012.

COLLINS, E., LUCENA JR, V. **Software Testing Automation Practices in Agile Environment: And Industry Experience Report**. IEEE, 2012.

CRESPO, A; SILVA, O; BORGES, C. At all. **Uma metodologia para teste de software no contexto da melhoria de processo**. Unicamp, 2004.

CRISPIN, L; GREGORY, J. **A pratical guide for testers and agile teams**. Addison-Wesley, 2009. 573 p.

Cucumber - Making BDD fun. Disponível em < http://cukes.info/> . Acessado em novembro de 2014.

DIAS, LEONARDO. **Método de instanciação de uma arquitetura de**

processos aplicado em fábrica de software. Dissertação de Mestrado à Escola Politécnica da Universidade de São Paulo, 2010.

MULLER, T; FRIEDENBERG, D. **Certified Tester Foundation Level Syllabus**. ISTQB, 2011.

MYERS, GLENDFORD J. **The art of Software Testing.** New York, 1979.

PATEL, C., RAMACHANDRAN, M. **Agile Maturity Model (AMM): A Software Process Improvement framework for Agile Software Development Practices**. Int.J. of Software Engineering, IJSE Vol.2 No.1, 2009.

RIOS, EMERSON., CRISTALLI, RICARDO. **Introdução ao TMM – Test Maturity Model**.Disponível em <http://www.itreinamento.com.br/LinkClick.aspx?fileticket=qqVal9psbb4%3D&tabid=249&mid=440 >. Acessado em dezembro de 2014.

SCHWABER, K.; SUTHERLAND, J.; **Scrum Guide**; Scrum.org, 2011.

State of Agile Survey. Disponível em < http://www.versionone.com/pdf/2009_State_of_Agile_Development_Survey_Results.pdf >. Acessado em dezembro de 2014.

Welcome to the TMMi Foundation | TMMi Foundation. Disponível em < http://www.tmmi.org/ >. Acessado em dezembro de 2014.