



UNIVERSIDADE FEDERAL DE PERNAMBUCO

CENTRO DE INFORMÁTICA

SISTEMAS DE INFORMAÇÃO

MARCELO AGUIAR COELHO DE MOURA FILHO

**Definição de um conjunto de diretrizes para implementar o TMMi no contexto de
desenvolvimento ágil**

Recife

Maio de 2022

MARCELO AGUIAR COELHO DE MOURA FILHO

Definição de um conjunto de diretrizes para implementar o TMMi no contexto de desenvolvimento ágil

Trabalho apresentado ao Programa de Graduação em Sistemas de Informação do Centro de Informática da Universidade Federal de Pernambuco como requisito parcial para obtenção do grau de Bacharel em Sistemas de Informação.

Orientador: Alexandre Marcos Lins de Vasconcelos

Recife

Maio de 2022

MARCELO AGUIAR COELHO DE MOURA FILHO

Definição de um conjunto de diretrizes para implementar o TMMi no contexto de desenvolvimento ágil

Trabalho apresentado ao Programa de Graduação em Sistemas de Informação do Centro de Informática da Universidade Federal de Pernambuco como requisito parcial para obtenção do grau de Bacharel em Sistemas de Informação.

Recife, 16 de Maio de 2022

BANCA EXAMINADORA

Prof. Alexandre Marcos Lins de Vasconcelos (Orientador)

Centro de Informática - UFPE

Prof. Juliano Manabu Iyoda (2º membro da banca)

Centro de Informática - UFPE

AGRADECIMENTOS

Muito obrigado a minha família que me deu suporte para que eu pudesse seguir no curso com êxito. Meus agradecimentos ao meu professor orientador Alexandre Vasconcelos pela orientação na realização deste trabalho.

"A ausência da evidência não significa evidência da ausência." (Carl Sagan)

RESUMO

Garantir a qualidade no contexto de desenvolvimento de software é muito importante para que as principais necessidades dos clientes sejam supridas. Este trabalho tem como objetivo propor e criar diretrizes que possam viabilizar níveis mais adequados de qualidade ao processo de testes para as pessoas envolvidas com esta atividade no processo de desenvolvimento de software. As diretrizes se baseiam nos requisitos para atingir os níveis de maturidade do TMMi, um modelo de maturidade para testes de software. Este modelo possui cinco níveis de maturidade e o trabalho irá propor diretrizes que, se aplicadas corretamente, podem viabilizar que a organização alcance o segundo nível, o qual funciona com um estágio inicial para que a empresa deixe de ter um processo de testes *ad-hoc*. Ao final do trabalho, será avaliado se os conjuntos de diretrizes e práticas são elegíveis para serem utilizados na prática.

Palavras-chave: Diretriz, qualidade e TMMi.

ABSTRACT

Ensuring quality in the context of software development is very important so that the main needs of customers are met. This work aims to propose and create guidelines that can enable more adequate levels of quality in the testing process for people involved with this activity in the software development process. The guidelines are based on the requirements for reaching the TMMi maturity levels, a maturity model for software testing. This model has five maturity levels and this work will propose guidelines that, if applied correctly, can enable the organization to reach the second level, which works with an initial stage so that the company no longer has an ad-hoc testing process. At the end of this work, it will be evaluated whether the sets of guidelines and practices are eligible to be used in practice.

Keywords: Guideline, quality and TMMi.

LISTA DE FIGURAS

Figura 1 — BPMN do processo de elaboração das diretrizes	13
Figura 2 — Níveis de maturidade do TMMi	19
Figura 3 — Resultados da pesquisa	37
Figura 4 — Resultados da pesquisa	38
Figura 5 — Resultados da pesquisa	39

LISTA DE ABREVIATURAS E SIGLAS

BPMN	Business Process Model and Notation
CTAL	Certified Tester Advanced Level
CTAL-TM	Certified Tester Advanced Level - Test Manager
CTFL	Certified Tester Foundation Level
TMMi	Test Maturity Model integration
SG	Specific Goal (objetivo específico)
SP	Specific Practice (prática específica)
GG	Generic Goal (objetivo genérico)
GP	Generic Practice (prática genérica)

SUMÁRIO

1.	INTRODUÇÃO.....	11
1.1.	PERGUNTA DE PESQUISA	12
1.2.	OBJETIVOS.....	12
1.3.	METODOLOGIA DE PESQUISA.....	12
1.4.	ESTRUTURA DE TRABALHO	13
2.	REFERENCIAL TEÓRICO.....	14
2.1.	PROCESSO DE TESTES	14
2.2.	TESTES DE SOFTWARE NO CONTEXTO ÁGIL.....	14
2.3.	NÍVEIS DE TESTE	15
2.4.	TIPOS DE TESTE	16
2.5.	O TMMi	16
2.5.1.	NÍVEL INICIAL	18
2.5.2.	NÍVEL GERENCIADO	18
2.5.3.	NÍVEL DEFINIDO	19
2.5.4.	NÍVEL MEDIDO	20
2.5.5.	NÍVEL OTIMIZADO	21
2.6.	O TMMi NO CONTEXTO ÁGIL	22
2.7.	DIRETRIZES	22
2.8.	CONSIDERAÇÕES FINAIS DO CAPÍTULO	23
3.	DIRETRIZES PARA ALCANÇAR O NÍVEL 2.....	24
3.1.	A EMPRESA POSSUI UMA POLÍTICA DE TESTES?.....	24
3.2.	O TIME DE QUALIDADE DE SOFTWARE DESENVOLVE ESTRATÉGIAS DE TESTE?.....	25
3.3.	SÃO DESENVOLVIDOS PLANOS DE TESTE?.....	27
3.4.	HÁ MONITORAMENTO E CONTROLE DOS TESTES?	29

3.5. EXISTE UM PROCESSO BEM ESTRUTURADO PARA MODELAGEM E EXECUÇÃO DOS TESTES?	30
3.6. O AMBIENTE DE TESTE PERMITE A EXECUÇÃO DE TESTES GERENCIÁVEIS E REPETÍVEIS?	32
3.7. CONSIDERAÇÕES FINAIS DO CAPÍTULO	33
4. AVALIAÇÃO DAS DIRETRIZES E RESULTADO DE PESQUISA.....	34
4.1. POLÍTICA E ESTRATÉGIA DE TESTES.....	34
4.2. PLANEJAMENTO E MONITORAMENTO DE TESTES	35
4.3. MODELAGEM E AMBIENTE DE TESTES	36
4.4. CONSIDERAÇÕES FINAIS DO CAPÍTULO	37
5. CONCLUSÃO.....	38
6. REFERÊNCIAS	38
APÊNDICE A - POLÍTICA DE TESTES.....	40
APÊNDICE B - ESTRATÉGIA DE TESTES	41
APÊNDICE C - PLANO DE TESTES	62
APÊNDICE D - MONITORAMENTO E CONTROLE DOS TESTES	65
APÊNDICE E - MODELAGEM E EXECUÇÃO DE TESTES	66
APÊNDICE F - QUESTIONÁRIO ACERCA DAS ÁREAS DE PROCESSO DO TMMi	67

1. INTRODUÇÃO

Cada vez mais, a demanda pelo uso de software tem aumentado e, conseqüentemente, tem aumentado também a necessidade de melhores níveis de qualidade no seu processo de desenvolvimento. Segundo Exploring principles of user-centered ágil software development [1], na última década, o desenvolvimento de software foi caracterizado por duas principais abordagens. A abordagem tradicional se caracteriza por possuir um modelo sequencial em que uma tarefa não pode ser iniciada até a anterior ser concluída. As metodologias tradicionais não oferecem muita flexibilidade perante as possíveis mudanças que possam surgir no ciclo de vida do projeto. Já na metodologia ágil, o software é construído por pequenas equipes usando os princípios do design contínuo, melhoria e testes baseados em feedback rápido e mudanças, buscando aumentar a flexibilidade e a velocidade no desenvolvimento.

No contexto do desenvolvimento ágil, uma das atividades que se destaca são os testes de software. Segundo o Syllabus do BSTQB [2], o processo de teste de software envolve verificar se o sistema atende aos requisitos especificados, bem como se atenderá às necessidades dos stakeholders em seu ambiente operacional.

Para garantir melhores níveis de qualidade ao processo de testes, foram criados modelos de referência como o Test Maturity Model Integration (TMMi) [4]. O modelo TMMi utiliza terminologias provenientes do ISTQB [3] e é formado por um conjunto de áreas de processo que contêm práticas bem estruturadas associadas ao processo de testes, distribuídas em cinco níveis de maturidade. Os níveis de maturidade são sequenciais e cumulativos, ou seja, para implementar um deles, o anterior precisa ter sido finalizado. Além disso, o TMMi [4], agrega muitas características do desenvolvimento e execução dos testes no contexto ágil, como citado no documento “TMMi no mundo ágil” [5] que mostra que, quando o TMMi é integrado ao desenvolvimento ágil, pode resultar em benefícios substanciais. Por esta razão, este modelo foi escolhido para servir de alicerce para a definição de diretrizes a serem criadas com o objetivo de mostrar como aplicar na prática, ao contexto do desenvolvimento ágil, alguns conceitos que são descritos nas áreas de processo do TMMi.

Devido ao pouco tempo disponível para a realização do trabalho, o conjunto de diretrizes propostas irá focar apenas em áreas de processo voltadas para o nível 2 de maturidade, que é o primeiro nível de melhoria de processo, visto que o nível 1 (inicial) é onde as empresas se encontram antes de implementarem o TMMi.

1.1. PERGUNTA DE PESQUISA

É possível criar diretrizes que estejam de acordo com as áreas de processos do TMMi [4] de modo a ajudar os projetos ágeis a alcançarem níveis mais adequados de qualidade?

1.2. OBJETIVOS

O trabalho possui, como objetivo a criação de diretrizes que englobam as práticas descritas nas áreas de processos do TMMi [4], instanciadas para o contexto do desenvolvimento ágil de software.

Após a criação das diretrizes, será realizada uma pesquisa com engenheiros de qualidade e áreas correlatas de quatro organizações diferentes, com a finalidade de verificar a aplicabilidade das diretrizes criadas ao contexto das organizações. Devido à falta de tempo para rodar o conjunto de diretrizes em pelo menos uma iteração, a sua eficácia não será avaliada neste trabalho, apenas a sua aplicabilidade.

1.3. METODOLOGIA DE PESQUISA

A metodologia utilizada para a execução deste trabalho foi dividida nas seguintes etapas: pesquisa na literatura sobre processos de teste, leitura e entendimento dos componentes do TMMi, estruturação e criação das diretrizes, além de um questionário referente às áreas de processo associadas a cada diretriz (Apêndice F). Após criado o conjunto de diretrizes, estas foram apresentadas para engenheiros de qualidade de software para que pudessem verificar se as áreas de processo referentes às perguntas utilizadas para definir as diretrizes estão presentes em suas respectivas organizações, com a finalidade de avaliar a aplicabilidade das mesmas. O processo se deu início através da busca na literatura por evidências que mostrem a importância dos processos de testes. Em seguida, foi feito um estudo do TMMi [4] e do TMMi no mundo ágil [5] de modo a entender a estrutura do TMMi, com finalidade de criar diretrizes que pudessem contribuir com os processos de teste em organizações ágeis. Com o conjunto de diretrizes criado, foi feita uma pesquisa com pessoas da área de testes de software, com pelo menos dois anos de experiência, para que pudessem verificar se as áreas de processo relacionadas às diretrizes estão de acordo com os processos já conhecidos por elas.

A Figura 1 apresenta um diagrama BPMN que descreve o processo de concepção das diretrizes.

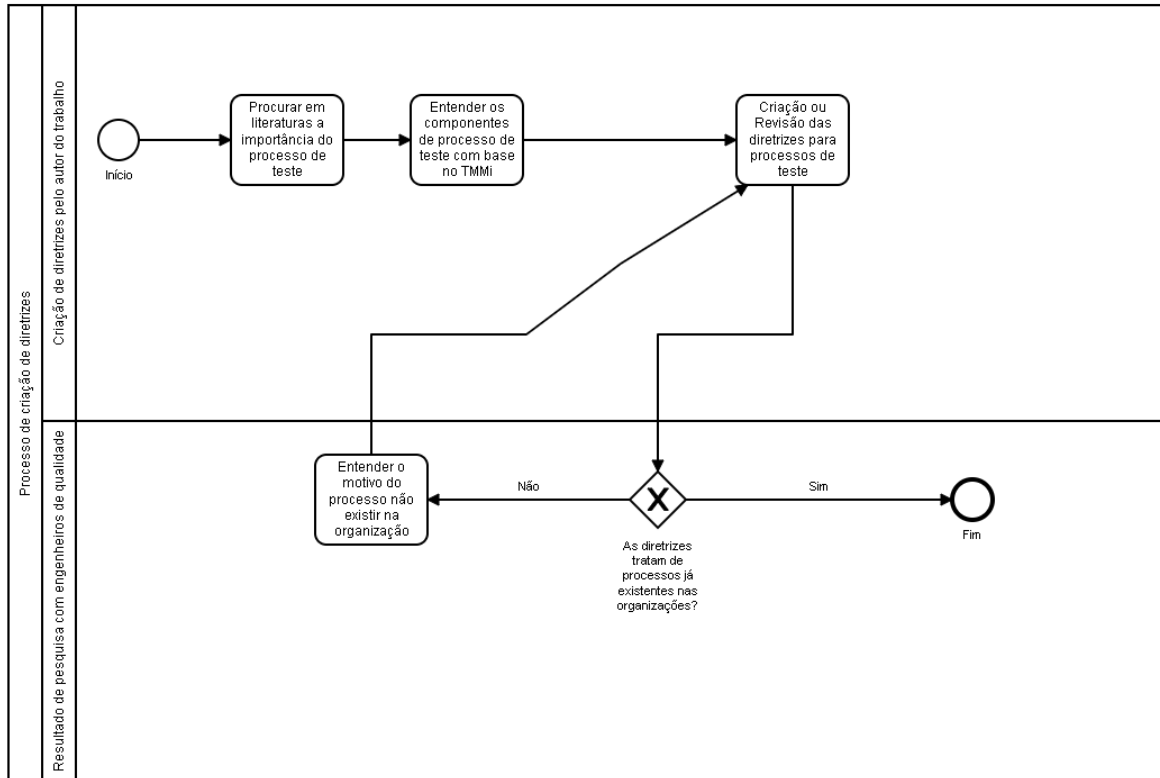


Figura 1

Fonte: elaborado pelo autor

1.4. ESTRUTURA DE TRABALHO

Além deste capítulo introdutório que explica os principais objetivos do trabalho e a metodologia que será utilizada na pesquisa, o trabalho é formado por mais 5 capítulos. O capítulo 2 apresenta os principais conceitos que serão utilizados ao longo do documento. O capítulo 3 apresenta as diretrizes criadas a partir das áreas de processo do TMMi [4], incluindo exemplos práticos de como implementá-las no contexto do desenvolvimento ágil. O capítulo 4 descreve as avaliações das diretrizes e resultado de pesquisa e o que se pode concluir a partir delas. O capítulo 5 apresenta as conclusões do trabalho. O trabalho também é composto por apêndices que auxiliam no entendimento para a aplicação das diretrizes.

2. REFERENCIAL TEÓRICO

Este capítulo apresenta o referencial teórico sobre os conceitos utilizados ao longo deste trabalho. O processo de testes é explicado, dando ênfase à importância de que o mesmo atinja níveis mais adequados de qualidade. Será explicado também a utilização dos testes no contexto de desenvolvimento ágil. Após isso, será descrito o TMMi [4], modelo de maturidade de testes utilizado neste trabalho, juntamente com uma explicação do que é o TMMi no mundo ágil [5]. Para finalizar o capítulo, será descrito o conceito de diretrizes a ser utilizado no trabalho.

2.1. PROCESSO DE TESTES

Segundo o BSTQB [2], apesar de não existir um processo universal para os testes de software, existem conjuntos de atividades que, sem elas, os testes possuem menos chances de atingir seus objetivos e garantir melhores níveis de qualidade. Estes conjuntos em questão são denominados de processos de teste. Nesse contexto, é importante pontuar os principais objetivos dos testes de software, segundo o Syllabus do BSTQB [2], que podem ser comuns para todos os projetos: mitigar a chance de defeitos serem inseridos no produto, verificar se o produto está de acordo com todos os requisitos estipulados com os principais stakeholders, verificar se os testes fornecem um nível adequado de confiança em relação ao objeto testado, bem como encontrar falhas e defeitos antes do produto ser disponibilizado para o usuário final.

2.2. TESTES DE SOFTWARE NO CONTEXTO ÁGIL

No contexto de desenvolvimento ágil, os ciclos de desenvolvimento são mais curtos em comparação às metodologias tradicionais. Na metodologia Scrum, por exemplo, no início de uma iteração, chamada Sprint, há a cerimônia de planejamento, na qual as histórias de usuários que serão desenvolvidas durante a Sprint são selecionadas. Estas histórias, posteriormente, são divididas em tarefas que descrevem o que precisa ser implementado para que cada história possa ser considerada finalizada. Cada atividade recebe pontuações referentes ao tempo e esforço que precisará ser investido para a sua finalização. Durante o desenvolvimento de uma Sprint, os testes são fundamentais para garantir que o produto seja desenvolvido com qualidade. Para o desenvolvimento das diretrizes propostas serão utilizados alguns termos específicos do Scrum, como por exemplo, retrospectiva, planejamento, backlog

do produto e “definição de feito”. Estes termos servirão para que se possa falar dos processos de testes dentro de um contexto de desenvolvimento ágil.

Os testes de software são necessários para verificar se os requisitos funcionais e não funcionais estão sendo atendidos com êxito e para buscar possíveis falhas que possam ter sido inseridas durante o desenvolvimento do produto. Segundo o livro "Beautiful Testing: Leading Professionals Reveal How They Improve Software" [8], a melhor forma de considerar que uma funcionalidade foi implementada com sucesso é testá-la integrada com as demais. No contexto do desenvolvimento ágil isso se torna possível devido às iterações curtas.

2.3. NÍVEIS DE TESTE

Segundo o CTFL [3], os níveis de teste podem ser divididos em:

- Teste de componentes:

Neste nível, os testes são executados em componentes individuais de forma isolada em relação ao restante do sistema. Um teste de componente pode ser executado através da virtualização de algum serviço específico ou mock de dados com a finalidade de verificar apenas um componente em específico. É comum que em contextos ágeis, a criação de testes automáticos para componentes possa preceder a criação do código do componente em si.

- Teste de integração:

Já os testes de integração focam em verificar se as integrações entre os componentes e o sistema estão funcionando corretamente. No processo de desenvolvimento contínuo e incremental, é comum que os testes presentes nesse nível sejam automatizados e façam parte do processo de integração contínua.

- Teste do sistema:

Os testes de sistema, por sua vez, são concentrados no comportamento do sistema como um todo. Geralmente são considerados nesse nível os testes de ponta a ponta e testes não funcionais. O envolvimento precoce de testadores no refinamento da história do usuário ou em atividades de testes estáticos, como revisões, corroboram para que os testes de sistema sejam executados com mais acurácia.

- Teste de aceite:

O teste de aceite, similar ao teste de sistema, também é focado no funcionamento do produto como um todo. Nesse nível de teste, são produzidas informações que podem avaliar o sistema para implantação e utilização pelo usuário

final. Segundo o CTFL [3]: “O objetivo principal é desenvolver a confiança de que os usuários podem usar o sistema para atender às suas necessidades”.

2.4. TIPOS DE TESTE

Segundo o CTFL [3], os tipos de testes são divididos em :

- Teste funcional:

Os testes funcionais devem ser realizados em todos os níveis de teste. Esse tipo de teste envolve verificar se as funções executadas pelo sistema estão funcionando de acordo com o requisito estipulado.

- Teste não funcional:

Já os testes não funcionais, eles avaliam as características do produto, como segurança, usabilidade e performance. Os testes não funcionais podem ser realizados em todos os níveis de teste. O risco de descobrir defeitos não funcionais muito tardiamente no ciclo de vida do produto pode acarretar o fracasso de um projeto.

- Teste caixa-branca:

Os testes de caixa-branca são realizados com base na implementação do sistema. A criação de testes do tipo caixa-branca pode envolver habilidades ou conhecimentos especiais, como a maneira como o código é construído, como os dados são armazenados e até mesmo o conhecimento para se implantar ferramentas de cobertura de código.

- Teste relacionado à mudança:

Quando ocorre uma mudança no sistema, para adicionar novas funcionalidades ou até mesmo corrigir um defeito que tenha sido inserido no produto, é necessário checar se a mudança não impactou em funcionalidades que já estavam funcionando corretamente. Um teste relacionado à mudança pode ser o de confirmação, que é quando um defeito é corrigido e o objetivo do teste é checar se o defeito foi realmente corrigido e se os testes relacionados ao defeito estão passando. Também há o teste de regressão que está relacionado a alterações no código que podem afetar, acidentalmente, outras partes do código que já funcionava antes. Os testes de regressão tem como objetivo a execução de testes para detectar defeitos colaterais indesejados.

2.5. O TMMi

O modelo de maturidade do TMMi [4] é composto por cinco níveis, os quais são apresentados na figura 2 e descritos nas subseções a seguir. Cada nível, a partir do 2, possui objetivos genéricos e específicos, os quais possuem práticas que precisam ser implementadas para que a organização chegue a um determinado nível de maturidade. Segundo o TMMi [4], um objetivo específico descreve: “a característica única que deve estar presente para satisfazer a área do processo”, já os objetivos genéricos “são metas genéricas aparecem perto do final de uma área de processo e são chamadas de "genéricas" porque a mesma declaração de meta aparece em todas as áreas de processos”. Um exemplo de um objetivo genérico é o “GG 2 Institucionalizar um Processo Gerenciado”. Um processo gerenciado, segundo o TMMi [4], envolve os stakeholders, é um processo controlado e gerenciável. O processo gerenciado tem como objetivo produzir produtos de trabalho e a institucionalização do mesmo garante que ele seja mantido, mesmo em momentos de estresse. Uma das práticas genéricas associadas a este objetivo é “GP 2.4 Atribuir responsabilidades”. Como exemplo de um objetivo específico pode ser citado o “SG 1 Estabelecer uma política de teste“, o qual solicita que uma organização defina uma política de teste, um documento que descreve o que os testes significam para a empresa. Uma das práticas específicas associadas a este objetivo é “SP 1.1 Definir metas de teste”, que tem como objetivo entender as principais necessidades e objetivos para que as metas estejam alinhadas ao negócio do produto. Este trabalho irá focar apenas nos objetivos específicos do TMMi, pois as diretrizes são construídas baseadas nos objetivos específicos de cada área do processo, com a finalidade da diretriz por si só já auxiliar a implementar a área do processo. Segundo o TMMi [4], os objetivos e práticas genéricos aparecem ao final da implementação de um nível. Por exemplo: “ O objetivo genérico 3 é aplicável somente ao atingir o nível de maturidade 3 ou superior”. Por este motivo, as práticas e objetivos genéricos não serão levados em consideração na criação das diretrizes, pois as diretrizes terão como objetivo auxiliar a implementação do nível 2 de maturidade.

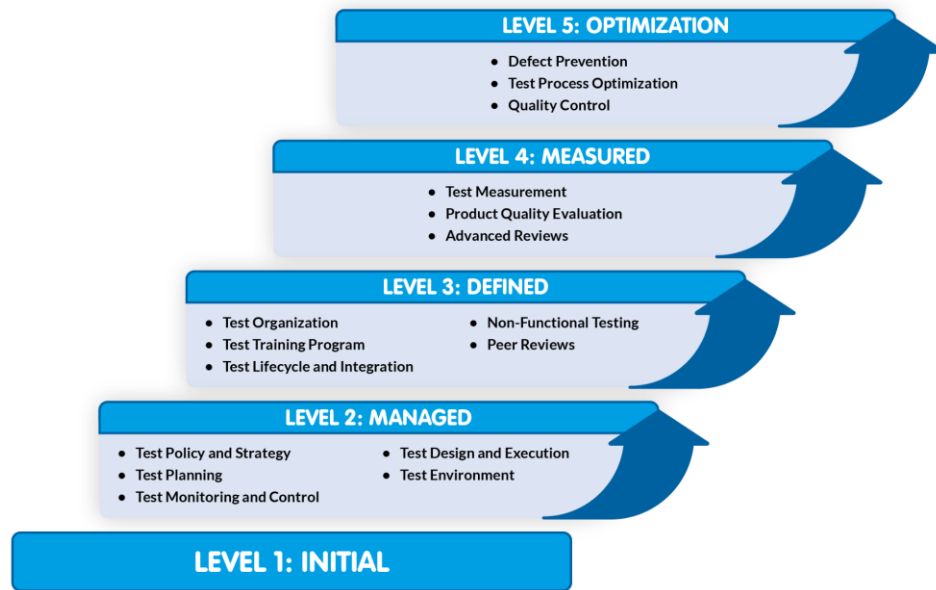


Figura 2 - Fonte: <https://www.tmmi.org/tmmi-model>

2.5.1. NÍVEL INICIAL

O nível 1 é o nível inicial do TMMi [4], onde os processos são caóticos e sem um entendimento do custo que a qualidade do produto possui. Times que se encontram no nível inicial geralmente não conseguem entregar os produtos com níveis de qualidade adequados e não suprem as expectativas dos clientes. Os testes são majoritariamente *ad-hoc* e desenvolvidos após a codificação do produto. Geralmente os testes desenvolvidos e executados no nível inicial possuem como objetivo mostrar que o produto funciona sem problemas que possam comprometer os principais fluxos da aplicação. Geralmente os lançamentos dos produtos ocorrem sem uma visibilidade a respeito da qualidade e dos riscos.

2.5.2. NÍVEL GERENCIADO

No nível 2, gerenciado, os objetivos descritos nas diretrizes do TMMi [4] começam a ficar mais rigorosos em relação ao nível inicial e demandam atividades e processos que ajudam a garantir níveis de qualidade mais adequados. As áreas de processo desse nível são:

- Política e Estratégia de Teste.

As políticas de testes costumam ser documentos de alto nível que refletem o que é esperado dos testes e o que significam dentro da organização, enquanto a estratégia de testes, apesar de também ser um documento de alto nível, estão mais relacionados com a execução, escopo e ambiente de testes.

- Planejamento de Testes.

O plano de testes é preparado em nível de projeto, além disso, ele também define como o produto deverá ser testado e como será feita a distribuição dos testes para os testadores executá-los. Também inclui a descrição dos ambientes de testes, ferramentas que serão utilizadas e os tipos de testes que serão executados.

- Monitoramento e Controle de Testes.

Abrange o monitoramento do progresso do teste e a qualidade do produto em relação às estimativas documentadas, os compromissos, os planos e as expectativas, os relatórios sobre o progresso do teste e a qualidade do produto para os stakeholders.

- Projeto e Execução de Testes.

Essa área aborda a fase de preparação do teste, incluindo a aplicação de técnicas de projeto de teste para derivar e selecionar condições de teste e casos de teste. Também aborda a criação de dados de teste específicos, a execução dos testes usando procedimentos de teste documentados e o gerenciamento de incidentes.

- Ambiente de teste.

Nesta área de processo são tratados pontos como a disponibilidade dos ambientes onde os testes serão executados, bem como o que é necessário para configurá-los.

2.5.3. NÍVEL DEFINIDO

O nível 3, definido, pode ser trabalhado pela organização após terem sido cumpridos os objetivos específicos do nível gerenciado. Isso ocorre pelo fato de que muitas habilidades do nível anterior são requeridas para a implantação dos objetivos específicos do nível definido. Há algumas diferenças entre o nível definido e o gerenciado. No nível gerenciado, as descrições de processo e procedimentos podem mudar bastante de projeto para projeto em uma organização. Já no nível definido, estas descrições são padronizadas, o que acaba permitindo que a circulação de colaboradores entre diferentes projetos seja feita com mais facilidade. No entanto, neste nível, é possível que estas descrições sejam adaptadas para que se adequem a um projeto ou unidade organizacional. Outra diferença deste nível em relação

ao anterior é que os processos são descritos com mais rigor. As áreas de processo deste nível são:

- Organização de teste.
Define o funcionamento (tarefas, responsabilidades, estrutura de relatório) de um grupo independente de teste na organização, bem como o valor agregado deste grupo para os produtos que estão sendo desenvolvidos.
- Programa de treinamento de teste.
Aborda o estabelecimento de um plano organizacional de treinamento em teste e para deixar os testadores aptos a executarem os processos de testes que existirão durante o ciclo de vida do desenvolvimento do produto.
- Ciclo de vida e integração de teste.
Aborda todas as práticas para estabelecer e manter um conjunto utilizável de ativos de processo de teste organizacional.
- Teste não funcional.
Envolve a realização de uma avaliação de risco não funcional do produto e a definição de uma abordagem de teste com base nos riscos identificados. Os testes não funcionais envolvem verificações relacionadas a performance, disponibilidade e segurança, por exemplo. Em iterações curtas e entregas contínuas, pode ser difícil executar uma grande bateria de testes relacionada a um requisito não funcional, sendo assim, é importante elencar quais os principais riscos referentes aos requisitos não funcionais que não serão cobertos pelos testes.
- Revisões por pares.
Abrange as práticas para executar revisões por pares em produtos de trabalho. Por exemplo, testadores que revisam uma especificação de requisitos para verificar a sua testabilidade.

2.5.4. NÍVEL MEDIDO

Após implementados os dois níveis anteriores, o nível quatro pode ser implementado. Este nível é caracterizado por possuir processos de testes mensuráveis. Segundo o TMMi [4], a existência de um programa de medição viabiliza que uma organização possa implementar um processo de avaliação da qualidade do produto, definindo as necessidades de qualidade, atributos de qualidade e métricas de qualidade. As áreas de processo deste nível são:

- Medição do Teste.

As atividades de medição de testes precisam estar alinhadas com a necessidade do projeto. Essas necessidades podem ser estipuladas com os principais stakeholders do projeto. Para implementar essa área de processo, pode ser criada uma matriz de rastreabilidade com os principais objetivos da medição dos testes.

- Avaliação da Qualidade do Produto.

Esta área de processo engloba as práticas que ajudam a desenvolver um entendimento quantitativo do produto que está sendo desenvolvido para que se possam alcançar metas mensuráveis em relação à sua qualidade. Ou seja, essa área de processo visa a identificação e priorização das necessidades referentes à qualidade do produto.

- Revisões Avançadas.

Esta área de processo se baseia na prática de revisão por pares. Na revisão avançada, são estabelecidas práticas que corroboram para que as abordagens de teste sejam mais coordenadas nas análises por pares e execução de testes dinâmicos para checar se o comportamento do sistema está como o esperado.

2.5.5. NÍVEL OTIMIZADO

Ao alcançar o nível 5 de maturidade, uma organização é capaz de melhorar continuamente seus processos com base em um entendimento quantitativo de processos estatisticamente controlados. As áreas de processo deste nível 5 são:

- Prevenção de Defeitos.

Tem como objetivo analisar as causas comuns de defeitos que são inseridos no produto ao longo do ciclo de vida do desenvolvimento para que possam ser definidas ações que impeçam que erros semelhantes sejam inseridos no futuro.

- Controle de Qualidade.

Tem como objetivo gerenciar e controlar estatisticamente o processo de teste. Segundo o TMMi [4], nesse nível de maturidade, os testes são escolhidos com base em amostras estatisticamente representativas com finalidade de tornar o processo de teste mais eficiente e manter a qualidade do produto

- Otimização do Processo de Teste.

Nesta área de processo, são identificadas, de maneira proativa e sistemática, o que pode ser continuamente melhorado no processo de teste. Nesta área de processo são geradas propostas de melhorias com base nos testes analisados.

2.6. O TMMi NO CONTEXTO ÁGIL

É fundamental entender que o TMMi [4] foi criado, inicialmente, para ser um modelo de maturidade que ajudasse as organizações a atingir níveis de maturidade mais altos no que diz respeito aos processos de testes de software. Com o advento das metodologias ágeis, viu-se necessária a criação de um documento que ajudasse as empresas a aplicar este modelo no contexto do desenvolvimento ágil de software.

Segundo a versão 1.3 do documento TMMi no mundo ágil [5], o TMMi aplicado ao contexto de desenvolvimento ágil tem como objetivo identificar as áreas de testes específicas onde as mudanças podem fornecer valor levando em conta os objetivos de negócio. O uso do modelo TMMi em um contexto ágil viabiliza práticas críticas de teste que geralmente são "esquecidas", mostrando que é possível que sejam aplicados princípios que viabilizem que o desenvolvimento ágil ocorra em conjunto com as práticas do TMMi.

2.7. DIRETRIZES

Segundo o Dicionário Priberam [7], diretriz significa norma, indicação ou instrução que serve de orientação. Neste trabalho, portanto, as diretrizes são estabelecidas com o intuito de servir como guia para que engenheiros de qualidade ou profissionais de áreas correlatas possam aplicá-las e alcançar níveis de qualidade mais adequados no processo de testes.

Cada diretriz está associada a uma pergunta referente a uma área de processo do TMMi. Esta pergunta deverá ser respondida de forma objetiva com “Sim”, “Não” ou “Não se aplica”. Em seguida, haverá uma breve explicação sobre o que se trata a área do processo, bem como uma sugestão de como implementar, em um contexto ágil, a área de processo associada à pergunta. A sugestão de implementação possuirá documentos de apoio ou diretivas para que a área de processo seja aplicada com sucesso.

As diretrizes propostas neste trabalho diferem do que está presente no documento do TMMi no mundo ágil [5], pelo fato de possuírem exemplos práticos e dicas que auxiliam o leitor no entendimento de como implementar as áreas de processo no contexto do desenvolvimento ágil.

2.8. CONSIDERAÇÕES FINAIS DO CAPÍTULO

Neste capítulo foi explicado o que é o processo de teste, o TMMi e seus respectivos níveis de maturidade e áreas de processo, bem como o TMMi no contexto ágil. Com base neste referencial teórico, o próximo capítulo apresenta um conjunto de diretrizes criadas para auxiliar na implantação do nível 2 de maturidade do TMMi no contexto do desenvolvimento ágil de software.

3. DIRETRIZES PARA ALCANÇAR O NÍVEL 2

Como visto no capítulo anterior, algumas áreas de processo devem ser implementadas, através de práticas, para que um nível de maturidade do TMMi [4] possa ser considerado implementado. Neste capítulo, serão apresentadas diretrizes para que o nível 2 possa ser atingido. Devido a restrições de tempo, os demais níveis do TMMi não serão abordados.

No nível 2, os testes são executados muito tardiamente no ciclo de desenvolvimento, ou seja, as organizações que estão neste nível de maturidade não enxergaram o processo de teste de software como algo presente durante todo ciclo de vida de desenvolvimento do produto. Além do mais, não existem programas formais de revisão, o que pode gerar problemas, tais como a propagação de defeitos desde os requisitos até o código. No entanto, o nível dois ainda é considerado melhor que o nível inicial, no qual os processos são caóticos.

Nas seções a seguir, será apresentada uma diretriz associada a cada área de processo do nível 2 do TMMi. Para cada diretriz, haverá um exemplo mostrando a sua aplicabilidade.

3.1. A EMPRESA POSSUI UMA POLÍTICA DE TESTES?

Uma política de testes é um documento que especifica as expectativas e os principais objetivos dos testes no projeto em questão, deixando claro o valor do teste no ambiente e o produto com o qual o time de qualidade está envolvido. Uma boa política de testes pode proporcionar níveis adequados de testes e, conseqüentemente, de qualidade. No geral, a política pode conter os frameworks utilizados, a definição do processo de teste e o objetivo para cada nível. A política também pode fornecer formas de medir a eficácia dos testes e futuros pontos de melhoria. A política de testes, segundo o BSTQB [2], influencia diretamente a estratégia de teste da organização. Para medir o desempenho dos testes abrangidos pelas políticas, segundo o TMMi [5], são introduzidos indicadores de desempenho de teste na política. Segundo o glossário do ISTQB [3], uma política de testes deve ser de alto nível, conter os principais objetivos e princípios da organização em relação aos testes. Para criar uma política de testes, segundo o CTAL-TM, Syllabus do ISTQB [3], alguns pontos importantes precisam ser abrangidos, são eles:

- Resumir o valor que a organização obtém dos testes;
- Descrever como avaliar a eficácia e eficiência dos testes para atender a esses objetivos;

- Descrever o processo de teste típico, tendo os processos descritos no ISTQB [3] como base;
- Deixar explícito como a organização irá melhorar seus processos de teste;

Área de Processo Associada:

Esta diretriz está associada à área de processo Política e Estratégia de Teste e possui a seguinte meta e práticas específicas associadas:

SG 1 Estabelecer uma política de teste.

SP 1.1 Definir metas de teste.

SP 1.2 Definir política de teste.

SP 1.3 Distribuir a política de teste para os stakeholders.

Implementação no contexto ágil:

Uma política de testes tem como objetivo descrever o que os testes significam para a organização, bem como as principais abordagens de teste utilizadas. O que diferencia uma política de testes desenvolvida para uma organização que utiliza uma metodologia ágil em comparação a uma desenvolvida para uma organização que utiliza uma metodologia tradicional, é o fato de que a documentação é mais enxuta e, algumas vezes, até engloba o conteúdo de outros documentos como a estratégia e planos de testes, como citado no próprio TMMi no mundo ágil [5].

Um exemplo de política de testes voltada ao contexto ágil é apresentado no apêndice A deste trabalho. Como pode-se perceber, a política de testes está escrita de uma forma simples e direta, de forma a servir de base para a criação das estratégias e planos de testes.

3.2. O TIME DE QUALIDADE DE SOFTWARE DESENVOLVE ESTRATÉGIAS DE TESTE?

Segundo o ISTQB [3], a estratégia de testes é um documento que está alinhado com a política de testes e deve detalhar como os testes devem ser realizados dentro de uma organização. A estratégia de teste é para toda a empresa ou para todo o programa. Um programa, segundo o PMBOK [10], é um grupo de projetos relacionados gerenciados de modo coordenado para a obtenção de benefícios estratégicos e controle que não estariam disponíveis se eles fossem gerenciados individualmente, contendo processos e abordagens para os testes que serão realizados nos softwares com um certo nível de detalhamento. A estratégia de teste

deverá possuir uma avaliação de risco e uma descrição dos tipos de teste que serão executados: teste de unidade, aceitação, desempenho, regressão, entre vários outros que possam ser incluídos. Também é necessário explicar os principais objetivos dos testes.

Área de processo associada:

Esta diretriz também está associada à área de processo Política e Estratégia de Teste e possui a seguinte meta e práticas específicas associadas:

SG 2 Estabelecer uma estratégia de teste.

SP 2.1 Realizar uma avaliação de risco de produto genérico.

SP 2.2 Definir estratégia de teste.

SP 2.3 Distribuir a estratégia de teste para os stakeholders.

SG 3 Estabelecer indicadores de desempenho de teste.

SP 3.1 Definir indicador de desempenho de testes.

SP 3.2 Implantar indicadores de desempenho de teste.

Implementação no contexto ágil:

Segundo o TMMi no mundo ágil [5]: “Uma estratégia de teste é um documento vital dentro de um ambiente ágil. Ele define em alto nível o teste a ser realizado pelas equipes ágil”. Para criar uma estratégia de teste é necessário possuir um documento com algumas informações, tais como quem irá revisar e aprovar o documento e qual será o cronograma dos testes executados. Outro ponto importante é a abordagem de teste que será utilizada, como será executado o processo de teste e qual a responsabilidade de cada membro da equipe durante o cronograma. É importante também definir os tipos dos testes que serão executados, como por exemplo, teste de carga, segurança, performance, funcionais, entre outros. Também é necessário descrever as ferramentas que serão utilizadas e se é viável e válido automatizar os testes. O documento também deverá conter os ambientes que serão utilizados no teste e como deixá-lo preparado para os testes. Por fim, a estratégia de testes deverá conter os riscos que podem ser estimados e os planos de contingência definidos. Um exemplo de template de estratégia de teste está presente no Apêndice B deste trabalho. No entanto, nem todos os capítulos e seções presentes no documento precisam ser utilizados. O documento pode variar conforme a necessidade de cada projeto.

No contexto ágil, é comum que as organizações tentem abordar estratégias de testes que consigam fornecer feedbacks de maneira mais rápida a respeito dos possíveis defeitos que

possam ter sido inseridos no produto durante o ciclo de desenvolvimento. Uma estratégia válida de testes, por exemplo, poderia ser: “80% dos testes criados serem de unidade com finalidade de receber feedbacks mais rápidos e minimizar perda de tempo”.

Em relação aos indicadores de desempenho no processo de teste, eles fornecem insumos para que a qualidade do software possa ser medida. Com base nos resultados destes indicadores, são tomadas decisões para que o produto possa ser construído com melhor qualidade. Como exemplo de indicadores de desempenho podem ser citados: Total de Defeitos Detectados, Total de Defeitos Encontrados pelo Cliente, Total de Defeitos Removidos, entre outros. Após estabelecidos quais os critérios e como monitorá-los, é necessário que os resultados dos testes sejam registrados para que os indicadores de qualidade possam indicar se o produto está sendo construído com os índices de qualidades esperados. No contexto ágil, onde as entregas são contínuas, pode acontecer que alguns defeitos não sejam detectados durante a iteração, se este número de defeitos for maior que o valor determinado no indicador de desempenho, pode indicar que alguns processos de teste precisam ser melhorados. Outro exemplo, ainda no contexto ágil, é o nível de satisfação do cliente com a entrega feita ao final de uma iteração.

3.3. SÃO DESENVOLVIDOS PLANOS DE TESTE?

Um plano de teste é um documento que é atualizado regularmente e define o escopo dos testes. Este documento contém os casos de teste a serem executados, os critérios de aceitação dos testes, a agenda de execução dos testes, bem como a distribuição das responsabilidades desempenhadas no processo de testes.

Área de processo associada:

Esta diretriz está associada à área de processo Planejamento de Teste e possui as seguintes metas e práticas específicas associadas:

SG 1 Executar uma Avaliação de Risco do Produto.

SP 1.1 Definir as categorias e os parâmetros de risco do produto.

SP 1.2 Identificar os riscos do produto.

SP 1.3 Analisar os riscos do produto.

SG 2 Estabelecer uma Abordagem de Teste.

SP 2.1 Identificar os itens e recursos a serem testados.

SP 2.2 Definir a abordagem de teste.

SP 2.3 Definir os critérios de entrada.

SP 2.4 Definir os critérios de saída.

SP 2.5 Definir os critérios de suspensão e retomada .

SG 3 Estabelecer as Estimativas do Teste.

SP 3.1 Estabelecer uma estrutura analítica de trabalho de nível superior

SP 3.2 Definir o ciclo de vida do teste.

SP 3.3 Determinar as estimativas para o esforço e custo do teste.

SG 4 Desenvolver um Plano de Teste.

SP 4.1 Estabelecer o cronograma de testes .

SP 4.2 Gerar um plano para o pessoal de teste.

SP 4.3 Planejar o envolvimento dos stakeholders.

SP 4.4 Identificar os riscos do projeto de teste.

SP 4.5 Estabelecer o plano de teste.

SG 5 Obter o Compromisso com o Plano de Teste.

SP 5.1 Revisar o plano de teste.

SP 5.2 Reconciliar os níveis de trabalho e recursos.

SP 5.3 Obter os compromissos do plano de teste.

Implementação no contexto ágil:

No Scrum, durante o planejamento de cada Sprint, a equipe de desenvolvimento seleciona as histórias disponíveis no backlog que estão com prioridades altas. Os testadores, no contexto da criação do planejamento de teste devem agregar valor, segundo o ISTQB [3], nas seguintes atividades: “Participar da análise detalhada de risco das histórias de usuários, determinar a testabilidade das histórias de usuários, criar testes de aceitação para histórias de usuários, dividir as histórias de usuários em tarefas (particularmente tarefas de teste), estimar o esforço de teste para todas as tarefas de teste, identificar os aspectos funcionais e não funcionais do sistema a ser testado e apoiar e participar na automação de testes nos vários níveis de testes.” Vale também que o plano de testes é fortemente influenciado pela política e pela estratégia de testes adotada pela organização. Um exemplo de plano de testes preenchido pode ser visto no Apêndice C.

Segundo o CTFL Foundation [3], os principais pontos presentes neste documento são:

- Definição do escopo, objetivos e riscos do teste.
- Definição da abordagem geral de teste.

- Definição das atividades de teste que serão integradas nas atividades do ciclo de vida de desenvolvimento do software.
- Definição sobre o que testar, pessoas e outros recursos necessários para realizar as atividades de teste e como elas serão executadas.
- Agendamento das atividades de análise, projeto, implementação, execução e avaliação de testes, seja em datas específicas ou no contexto de cada iteração.
- Seleção das métricas para monitoramento e controle de teste.
- Definição do custo para a realização das atividades de teste.
- Definição da estrutura de como os testes irão ser detalhados e documentados.

3.4. HÁ MONITORAMENTO E CONTROLE DOS TESTES?

O monitoramento e controle dos testes, segundo as diretrizes do TMMi [4], tem como objetivo proporcionar um entendimento da qualidade dos testes realizados no produto ao longo das iterações. Esse entendimento é de suma importância, a partir do momento em que medidas podem ser tomadas caso os testes estejam se desviando muito do planejamento. Os processos e resultados de testes devem ser constantemente revisados para que a partir dessa revisão as abordagens e o planejamento possam ser modificados conforme a necessidade e objetivo da organização.

Área de processo associada:

Esta diretriz está associada à área de processo Monitoramento e Controle de Teste e possui as seguintes metas e práticas específicas associadas:

SG 1 Monitorar o Progresso do Teste em Relação ao Plano.

SP 1.1 Monitorar os parâmetros de planejamento de teste.

SP 1.2 Monitorar os recursos do ambiente de teste fornecidos e usados.

SP 1.3 Monitorar os compromissos de teste .

SP 1.4 Monitorar os riscos do projeto de teste.

SP 1.5 Monitorar o envolvimento dos stakeholders.

SP 1.6 Realizar revisões de progresso do teste.

SP 1.7 Realizar revisões de marcos de progresso de testes.

SG 2 Monitorar a Qualidade do Produto em Relação ao Plano e às Expectativas.

SP 2.1 Verificar os critérios de entrada.

SP 2.2 Monitorar os defeitos.

SP 2.3 Monitorar os riscos do produto.

SP 2.4 Monitorar os critérios de saída.

SP 2.5 Monitorar os critérios de suspensão e reinício.

SP 2.6 Realizar as revisões de qualidade do produto.

SP 2.7 Realizar as revisões de marcos de qualidade de produto.

SG 3 Gerenciar Ações Corretivas para Encerramento.

SP 3.1 Analisar os problemas.

SP 3.2 Tomar ações corretivas.

SP 3.3 Gerenciar as ações corretivas.

Implementação no contexto ágil:

Tendo em mente que, no contexto ágil, as iterações são constantes e as mudanças de requisitos e necessidades dos stakeholders podem ser atualizadas, gerando mudança nos requisitos e nas especificações do produto final, faz-se necessário que o monitoramento e o controle dos testes sejam feitos regularmente para que o plano de testes possa ser atualizado de acordo com os novos requisitos e metas estipulados. Sendo assim, segundo o guia do browserstack [9], os processos de monitoramento e controle de testes devem:

- Fornecer feedback à equipe de QA e outras partes interessadas sobre o progresso dos ciclos de teste da iteração.
- Transmitir os resultados dos testes alcançados até o momento a todas as partes relevantes.
- Identificar e definir métricas de teste relevantes.
- Determinar o curso de ação futuro, com base nas métricas que estão sendo coletadas.

Um exemplo de plano de monitoramento pode ser encontrado no Apêndice D, o qual especifica do que se trata o projeto, quais são os principais itens que precisam ser monitorados e o histórico de revisão do documento. O que diferencia o monitoramento e controle no desenvolvimento ágil em relação a outras metodologias tradicionais é o fato de manterem-se atualizados de forma contínua.

3.5. EXISTE UM PROCESSO BEM ESTRUTURADO PARA MODELAGEM E EXECUÇÃO DOS TESTES?

A modelagem de teste está relacionada com as técnicas que são utilizadas para a concepção dos testes de software. O processo de modelagem e execução dos testes é

responsável pela preparação dos testes e definição das condições para a execução dos mesmos. Segundo o TMMi [4], nesse processo também é abordada a criação de dados para testes (descrição dos valores de entrada, pré-condições de execução, resultados esperados e condições de pós-execução) e, após estipulados esses dados, é criado um procedimento de teste que poderá ser executado. Ainda segundo o TMMi, a modelagem e a execução dos testes permitem melhorar a capacidade do processo de teste que fornece melhores práticas de planejamento e gerenciamento de desenvolvimento, assim como a manutenção do produto. Possibilitando, dessa maneira, que níveis mais adequados de qualidade possam ser alcançados. Um exemplo de modelagem e execução de teste está presente no Apêndice E.

Área de processo associada:

Esta diretriz está associada à área de processo Modelagem e Execução de Testes e possui as seguintes metas e práticas específicas associadas:

SG 1 Realizar a Análise e a Modelagem do Teste usando Técnicas de Projeto de Teste.

SP 1.1 Identificar e priorizar as condições de teste.

SP 1.2 Identificar e priorizar os casos de teste.

SP 1.3 Identificar os dados específicos de teste necessários.

SP 1.4 Manter a rastreabilidade horizontal com requisitos.

SG 2 Implementar o Teste.

SP 2.1 Desenvolver e priorizar os procedimentos do teste.

SP 2.2 Criar dados específicos de teste.

SP 2.3 Especificar o procedimento para realização do teste de admissão. Tem como objetivo verificar se o ambiente de testes está preparado para a execução dos testes especificados no plano de testes.

SP 2.4 Desenvolver o cronograma de execução de teste.

SG 3 Executar os Testes.

SP 3.1 Realizar os testes de admissão.

SP 3.2 Executar os casos de teste.

SP 3.3 Registrar os incidentes de teste.

SP 3.4 Escrever a log de teste.

SG 4 Gerenciar os Incidentes de Teste para o Encerramento.

SP 4.1 Decidir a disposição dos incidentes de teste no painel de controle de configuração.

SP 4.2 Executar a ação apropriada para corrigir os incidentes do teste.

SP 4.3 Acompanhar o status dos incidentes de teste.

Implementação no contexto ágil:

No contexto ágil, a modelagem dos testes geralmente começa em paralelo ao desenvolvimento do código. Segundo o TMMi no mundo ágil [4]: “Em um ambiente ágil, procedimentos detalhados de teste não são uma prática comum. Trabalhando em uma equipe experiente, os testes provavelmente serão documentados em um nível de abstração muito mais alto. Isso será suficiente para aqueles que executam os testes, pois é esperado que eles tenham o nível necessário de domínio e conhecimento técnico para realizar os testes.”. Dessa forma, fica evidente que, em um time de desenvolvimento ágil, é natural que a modelagem dos testes seja feita em um nível mais alto de abstração.

3.6. O AMBIENTE DE TESTE PERMITE A EXECUÇÃO DE TESTES GERENCIÁVEIS E REPETÍVEIS?

Segundo o glossário do ISTQB [6], um ambiente de teste é formado por hardware, instrumentação, simuladores, ferramentas de software e outros elementos de suporte necessários para realizar um teste. Dessa forma, um ambiente de testes controlado e gerenciado se torna fundamental para a execução de testes. Segundo o TMMi, a especificação do ambiente de teste é realizada no início do projeto. É preciso entender qual a demanda dos tipos de teste que serão executados para saber qual ambiente será disponibilizado, como os níveis e tipos do teste serão distribuídos pelos ambientes, bem como se o ambiente será compartilhado por mais de um time de teste (nesse caso, é necessário ter um controle rigoroso para que os testes possam continuar repetíveis e independentes). É necessário, também, que os ambientes estejam disponíveis no momento da execução dos testes.

Área de processo associada:

Esta diretriz está associada à área de processo Ambiente de Teste e possui as seguintes metas e práticas específicas associadas:

SG 1 Desenvolver os Requisitos do Ambiente de Teste.

SP 1.1 Descobrir a necessidade do ambiente de teste.

SP 1.2 Desenvolver os requisitos do ambiente de teste.

SP 1.3 Analisar os requisitos do ambiente de teste.

SG 2 Executar a Implementação do Ambiente de Teste.

SP 2.1 Implementar o ambiente de teste.

SP 2.2 Criar dados genéricos de teste.

SP 2.3 Especificar o procedimento de teste de entrada do ambiente de teste.

SP 2.4 Executar o teste de ingestão de ambiente de teste.

SG 3 Gerenciar e Controlar Ambientes de Teste

SP 3.1 Executar o gerenciamento de sistemas.

SP 3.2 Realizar o gerenciamento de dados de teste.

SP 3.3 Coordenar a disponibilidade e o uso dos ambientes de teste.

SP 3.4 Relatar e gerenciar incidentes do ambiente de teste.

Implementação no contexto ágil:

No contexto de desenvolvimento ágil, é comum que as empresas possuam mais de um ambiente para a execução de testes, permitindo que vários testes de tipos diferentes possam ser executados simultaneamente e de forma independente. Por exemplo, se a organização providencia uma infraestrutura para que os testes de performance e um outro para os testes de segurança. Isso viabiliza que os testes de um tipo específico não interfiram na execução dos outros testes. Levando em consideração que o ambiente de teste será configurado em uma organização ágil, então as atividades para preparar os ambientes são enxergadas como tarefas do processo ágil, sendo abordadas em cerimônias de planejamentos e reuniões diárias.

3.7. CONSIDERAÇÕES FINAIS DO CAPÍTULO

Neste capítulo foram definidas diretrizes para dar apoio à implementação do nível 2 do TMMi. No próximo capítulo, as diretrizes definidas serão avaliadas quanto à sua aplicabilidade, através de uma sondagem com engenheiros de testes de quatro organizações distintas.

4. AVALIAÇÃO DAS DIRETRIZES E RESULTADO DE PESQUISA

Através das diretrizes criadas será possível direcionar a implementação das áreas de processo do nível 2 do TMMi ao contexto de desenvolvimento ágil. Assim, com base nas diretrizes, foi gerado um formulário para que alguns engenheiros de qualidade ou de profissões correlatas pudessem avaliar se as áreas de processo do nível 2 são aplicadas ou não na sua organização. Para o caso em que a pessoa não souber se a área do processo se aplica ou não, foi adicionada a opção “Não se aplica”. Este formulário está presente no Apêndice C.

O formulário foi respondido por 5 engenheiros de qualidade de quatro empresas diferentes. Todos possuem mais de dois anos de experiência no ramo de qualidade de software e a média aritmética de experiência das pessoas que responderam é de oito anos e meio. Quatro das cinco pessoas que responderam o formulário não conheciam o TMMi ou não possuíam experiência com este modelo. Uma das pessoas disse conhecer o TMMi, mas indicou que possui pouca experiência no mesmo. Os resultados da pesquisa indicam que, mesmo sem ter familiaridade com o modelo de maturidade em questão, a maioria das pessoas trabalha em projetos que possuem processos semelhantes aos do TMMi. Uma síntese do resultado das respostas ao formulário pode ser observada nas subseções a seguir.

Caso alguma área de processo não exista na organização, a diretriz não poderia ser aplicada de imediato, precisando existir uma avaliação do projeto em que a pessoa está inserida para que possa ser entendido o que falta para a diretriz poder ser aplicada.

4.1. POLÍTICA E ESTRATÉGIA DE TESTES

Como pode ser observado na Figura 3, a maioria das pessoas que responderam o formulário sabem do que se trata uma política de testes e 100% das respostas indicam que as suas respectivas empresas possuem estratégias de teste. Com base nisso, é possível perceber que a existência de estratégia é algo comum em empresas com cultura ágil, sendo viável, portanto, a aplicação das diretrizes criadas relacionadas à área de processo de Política e Estratégia de Teste.



Figura 3 Política e Estratégia de Testes - Elaborado pelo autor

4.2. PLANEJAMENTO E MONITORAMENTO DE TESTES

Para ambas as perguntas presentes na Figura 4, todas as respostas indicaram que há planos de testes e monitoramento e controle de testes na organização. Pode-se concluir, portanto, que ambas as diretrizes relacionadas às áreas de processo de Planejamento de Testes e Monitoramento e Controle de Teste podem ser aplicadas nas organizações.

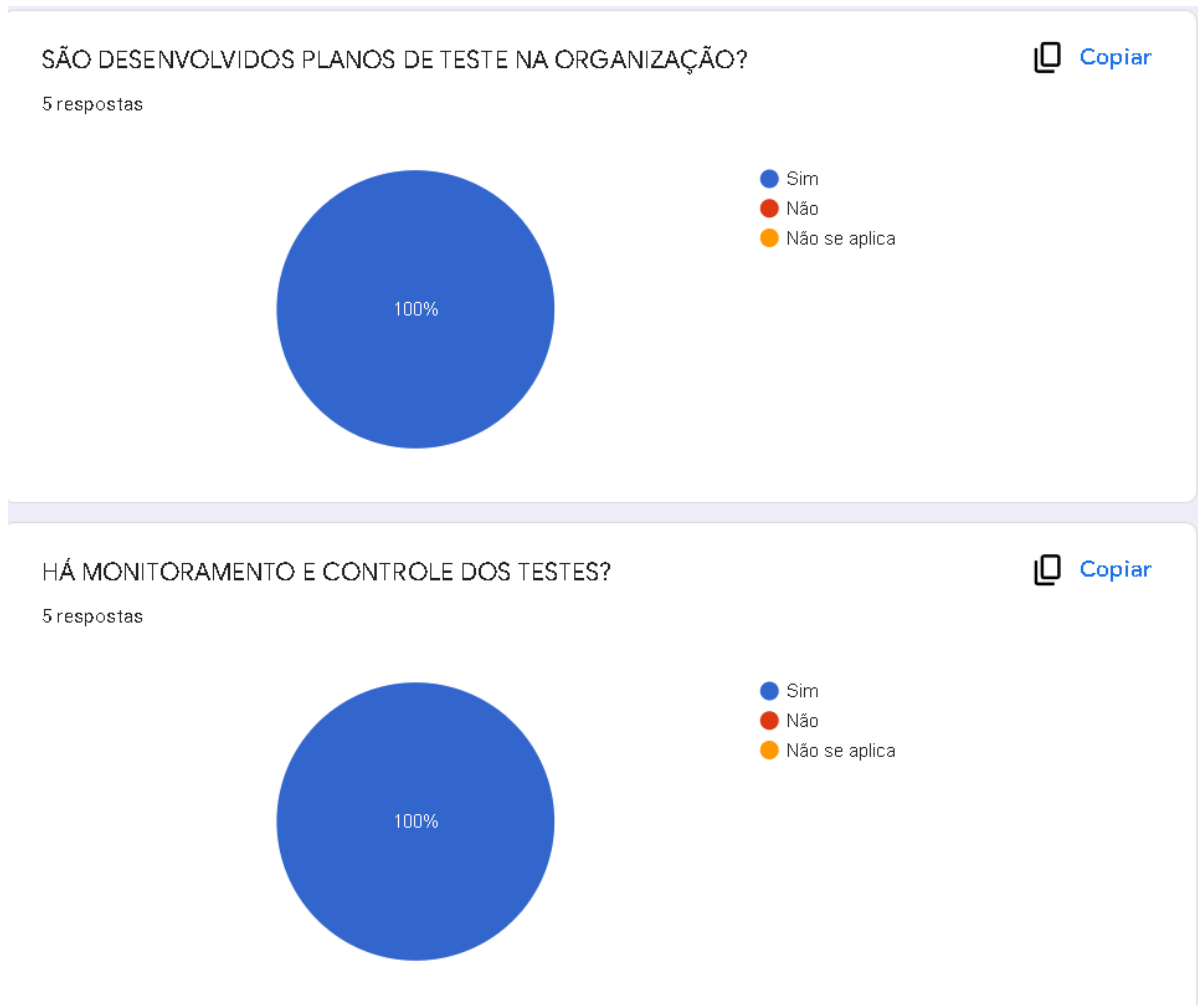


Figura 4 Planejamento e monitoramento de testes - Elaborado pelo autor

4.3. MODELAGEM E AMBIENTE DE TESTES

Na Figura 5, todas as respostas indicaram que as organizações possuem um processo para modelagem e execução dos testes. No entanto, na pergunta referente à existência de ambientes de testes gerenciáveis e repetíveis, duas das cinco pessoas indicaram não saber do que se trata essa pergunta. Apesar de duas pessoas não conhecerem a área relacionada a ambientes de testes, a diretriz continua sendo aplicável, visto que um ambiente de testes é fundamental para a execução de testes e a maioria das pessoas que responderam o formulário conheciam a área de processo. Ou seja, a diretriz referente a ambiente de testes continua sendo válida, pois os testes precisam de pelo menos um ambiente para serem executados.

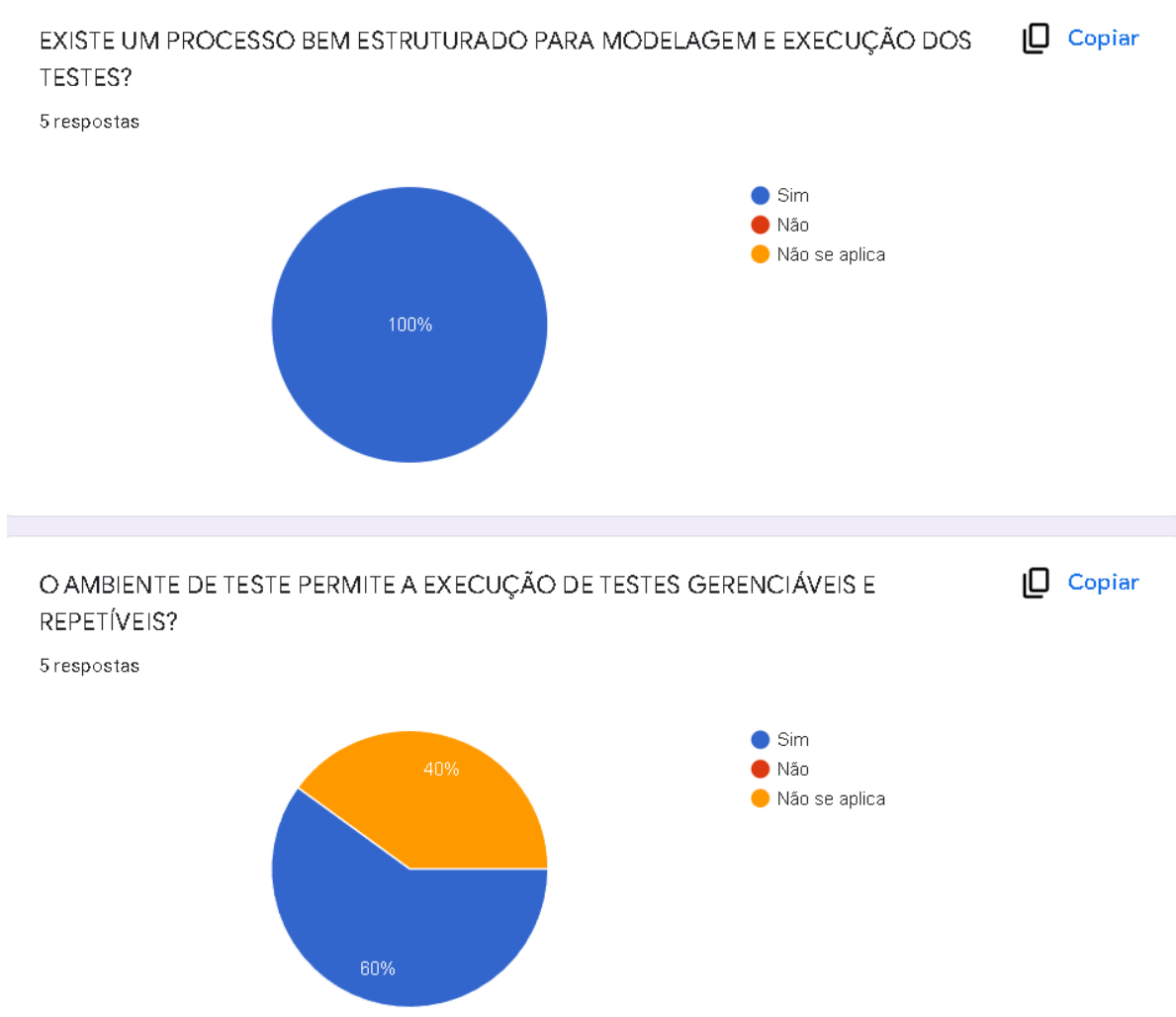


Figura 5 Modelagem e ambiente de testes - Elaborado pelo autor

4.4. CONSIDERAÇÕES FINAIS DO CAPÍTULO

Com base nas respostas referentes ao formulário, pode-se concluir que a maioria dos respondentes conhece os processos abordados pelas áreas de processo do nível 2 do TMMi, mesmo sem possuir familiaridade com esse modelo de maturidade. Sendo assim, conclui-se que a aplicação das diretrizes parece ser válida, pois elas parecem estar alinhadas com processos já existentes em organizações ágeis.

5. CONCLUSÃO

Pode-se concluir que os processos associados às áreas de processo do nível 2 do TMMi já são conhecidos por algumas organizações, mas o modelo em si com os seus objetivos e práticas não é difundido nas empresas. As diretrizes criadas, portanto, parecem estar de acordo com a vivência das organizações inseridas em um contexto ágil. Assim, conclui-se que as diretrizes são elegíveis para serem aplicadas nas organizações ágeis.

É válido pontuar, no entanto, que o trabalho em questão não avaliou se as diretrizes são válidas para corroborar com o processo de implementação do TMMi em contextos ágeis, ficando, portanto, esta avaliação para ser realizada em trabalhos futuros através da aplicação no contexto da execução de projetos.

6. REFERÊNCIAS

[1] Brhel, M., Meth, H., Maedche, A., & Werder, K. 2015, maio. Exploring principles of user-centered ágil software development: A literature review. Information and Software Technology. Elsevier BV.

[2] BSTQB - BRAZILIAN SOFTWARE TESTING QUALIFICATIONS BOARD. Certified Tester Foundation Level Syllabus: 2018br. 2019. Disponível em

[3] ISTQB - International Software Testing Qualifications Board Certified Tester Syllabus Foundation Level CTFL 3.1. Disponível em https://bstqb.org.br/b9/doc/syllabus_ctfl_3.1br.pdf . Acesso em 23, ABRIL, 2022

[4] TMMi - Test Maturity Model integration - Disponível em <https://tmmi.org/tm6/wp-content/uploads/2018/11/TMMi-Framework-R1-2.pdf>

[5] TMMi in the agile world - disponível em <https://www.tmmi.org/tm6/wp-content/uploads/2019/07/TMMi-in-the-ágil-world-V1.3.pdf> Acesso em 24, ABRIL, 2022

[6] ISTQB Glossary, 2022. Disponível em: <https://glossary.istqb.org/en/search/>. Acesso em: 23, 04, 2022.

[7] "diretriz", in Dicionário Priberam da Língua Portuguesa [em linha], 2008-2021, <https://dicionario.priberam.org/diretriz> [consultado em 28-04-2022].

[8] Riley, T. and Goucher, A., 2010. Beautiful testing. Sebastopol, Calif.: O'Reilly.

[9] Test monitoring-and-test-control disponível em:
<https://www.browserstack.com/guide/test-monitoring-and-test-control> Acesso em
03/05/2022

[10] Guia PMBOK® 6a. ed. – EUA: Project Management Institute, 2017. BORGES, Carlos; ROLLIM, Fabiano.

APÊNDICE A - POLÍTICA DE TESTES

Template de política de testes preenchida pelo autor para servir de demonstração de como montar uma política de testes em uma empresa:

Um documento de alto nível (à nível de empresa) descreve os princípios, a abordagem e os principais objetivos da organização em relação aos testes.

- Descrever o contexto da organização.
- Impactos na política
- Escrever as diretrizes em forma estruturada

Template para política de testes. Preenchimento realizado pelo autor:

1) Definição de teste

O teste na organização tem como objetivo garantir que os produtos estejam de acordo com os critérios de aceitação estabelecidos durante a Sprint e que as principais demandas de negócio estipuladas com o cliente sejam cobertas pelos testes.

2) Processo de teste

O processo de teste se inicia pelo planejamento dos mesmos, onde é estipulado o que será coberto pelos testes e o que será coberto pelos testes, assim como os riscos que as áreas não cobertas pelos testes podem acarretar. Os testes são definidos antes de começarem a ser executados, assim como a distribuição dos mesmos entre os testadores. O resultado de cada teste será registrado na ferramenta Jira com o status “aprovado”, “falhou”, “em espera” e deverá possuir os registros da execução do mesmo. O objetivo para a criação de um processo bem definido de teste com o planejamento e estratégias condizentes com a realidade de cada projeto, tem como principal objetivo que esse processo possa ser repetível em qualquer projeto da organização, mesmo que em momentos de estresse (altas demandas, time reduzido)

3) Padrões de testes

São seguidos os padrões das seguintes ISOs:

ISO/IEC/IEEE 29119-1, ISO/IEC/IEEE 29119-2

4) Medições de teste

Os resultados dos testes funcionais são divididos em:

Número total de casos de teste.

Número de casos de teste aprovados.

Número de casos de teste com falha.

Número de casos de teste bloqueados.

Número de defeitos encontrados.

Número de defeitos aceitos.

Número de defeitos rejeitados.

Número de defeitos adiados.

Cada módulo da aplicação deverá ter no mínimo 5 testes funcionais e no mínimo 60% dos testes devem ter sido aprovados com sucesso.

5) Riscos

Os riscos devem ser avaliados de acordo com os quadrantes de testes que não serão executados e no que o mau funcionamento dos mesmos podem acarretar para a organização.

7) Comentários

Os planos e estratégias de testes de cada projeto da organização devem estar alinhados com essa política de testes e seguir as diretrizes estipuladas nos artefatos que a organização oferece. Podendo haver adaptações e alterações dependendo da necessidade de cada projeto

Fonte do template:

<https://www.gcreddy.com/2022/01/test-documentation-templates.html> acesso no dia 02/05/2022

APÊNDICE B - ESTRATÉGIA DE TESTES

Templates para a estratégia de testes, disponíveis em:

<https://strongqa.com/qa-portal/testing-docs-templates/test-strategy>

Template escolhido pelo autor:

Estratégia de teste:

Histórico de alterações:

Versão	Data	Histórico de Alterações

Revisores:

Este documento deve ser revisado pelo seguinte:

Nome	Assinatura	Título / Responsabilidade	Data	Versão

Aprovações:

Este documento deve ser aprovado pelo seguinte:

Nome	Assinatura	Título / Responsabilidade	Data	Versão

Distribuição:

Status do documento:

Este é um documento controlado.

Embora este documento possa ser impresso, a versão eletrônica deve ser um documento controlado localizado em uma loja de documentos acordada. As cópias impressas deste documento não são controladas.

Documentos relacionados:

Esses documentos fornecerão informações adicionais.

Ref não	Número de referência do Doc	Título	Versão

Glossário dos Termos:

Liste os termos utilizados neste documento.

Prazo	Acrônimo	Definição

Tabela de Conteúdos

1.Introdução

1.1.Objetivos

1.2.Escopo

2.Testando

visão

geral

2.1.Teste

ciclo

de

vida

2.2.Abordagem de teste

2.3.Padrões

2.4.Estágios

de

teste

2.5.Organização

da

equipe

de

teste

2.6.Revisões

e

Inspeções

2.7.Documentação

do

teste

2.8.Planos

de

teste

2.9.Especificações

de

teste

2.10.Scripts

de

teste

2.11.Execução

de

teste

2.12.Critérios

de

suspensão

e

2.13.Critérios	de	entrada	e	saída
2.14.Resultados	do	teste		captura
2.15.Test				Harnesses
2.16.Abordagem	para	testes	de	regressão
2.17.Abordagem	para	testes	de	segurança
2.18.NHS CFH Engagement				
3.Gerenciamento				de
teste3				
3.1.Gestão		da		Qualidade
3.2.Abordagem	ao	Gerenciamento	de	Incidentes
3.3.Test				
3.4.Relatórios		de		progresso
3.5.Relatório		de		fase
3.6.Iniciativas		de		Melhoria
4.Controle				de
teste3				
5.Dados		de		teste
6.Testing				Environments3
7.Ferramentas				de
teste3				

7.1.Ferramentas de gerenciamento de teste
3

7.2.Ferramentas de automação de teste
3

1. Introdução

1.1. Objetivos

Este documento define a Estratégia de Teste do Projeto painel clínico e a abordagem das etapas de teste incrementais necessárias para garantir a aceitabilidade da solução entregue. Cobre todas as fases e lançamentos. A base dos processos de teste e aceitação será baseada na Orientação existente do Painel Clínico. Esses processos precisarão ser aprimorados para abraçar as declarações de aceitação e critérios dentro do documento.

1.2. Âmbito

Esta estratégia de teste cobrirá o seguinte:

- Identificando os sucessivos tipos de testes a serem realizados ao longo do ciclo de vida do desenvolvimento para operações ao vivo
- Detalhes dos testes em andamento de aprimoramento e mudança de serviço
- O escopo de cada tipo de teste
- Identificando como as expectativas comuns e os padrões de teste devem ser alcançados para todos os tipos de testes
- Os requisitos técnicos, de recursos e ambientais de alto nível exigidos
- Os principais procedimentos de teste e garantia de qualidade que serão necessários.

2. Visão geral de teste

2.1. Testar ciclo de vida

O ciclo de vida do teste a ser seguido é definido em detalhes dentro desta seção. O ciclo de vida é baseado no modelo V de testes, um modelo que é amplamente utilizado na indústria. A estratégia de teste identifica as linhas de base - tanto testes quanto entregas de desenvolvimento, que serão testadas em cada estágio de desenvolvimento, ou seja, testes ao longo do ciclo de vida. Ele será projetado para se adequar às etapas definidas no Processo de Garantia Comum. O Fornecedor assumirá a liderança no gerenciamento da definição e execução de todas as etapas de teste ao longo do ciclo de vida do projeto, com o suporte adequado dos recursos NHS CFH e Trust.

2.2. Abordagem de teste

O Fornecedor propõe uma abordagem padrão para testes, com base na metodologia utilizada para as implantações de Conexão para Saúde na Inglaterra. A abordagem proposta seguirá o modelo acordado, que identifica tanto as etapas de teste quanto às entregas que devem ser testadas em cada etapa do ciclo de vida do projeto.

A abordagem se enquadra em quatro estágios principais de teste, listados em ordem:

- Teste do sistema
- Teste de Integração
- Pronto para testes de operações, incluindo testes de volume e desempenho
- Teste de aceitação do usuário, incluindo qualquer teste de implantação

Há também um Estágio de Teste realizado uma vez para aspectos não funcionais:

- Teste central

Os testes de Aceitação e do Usuário podem ser realizados em paralelo (este último pode ser iniciado após o Teste de Integração), mas está previsto que eles serão realizados separadamente.

A metodologia do Ciclo de Teste de Projeto (excluindo testes centrais 'não funcionais') é descrita diagramaticamente abaixo.

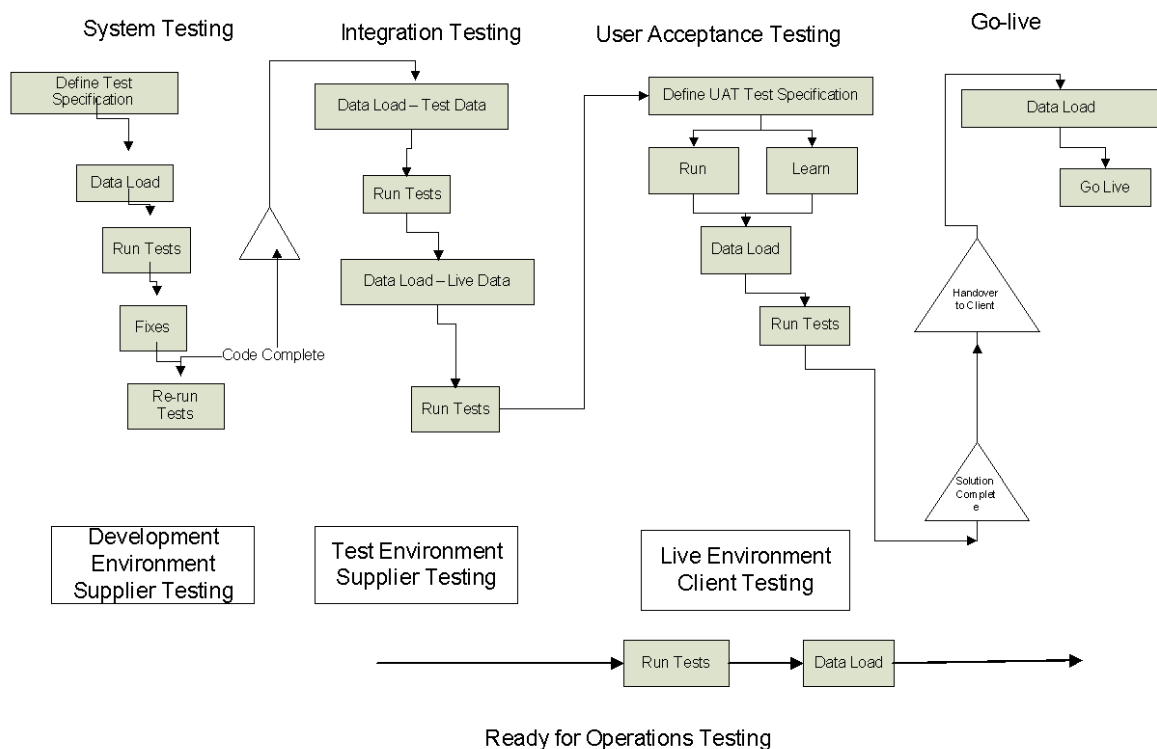


Figura - Diagrama do Ciclo de Vida de Teste

2.3. Padrões

Consulte quaisquer normas que serão cumpridas, especialmente se houver um elemento de segurança no software.

Além disso, para evitar quaisquer equívocos com o uso da terminologia, se o BS7925-1 não for aderido, por favor, defina os termos de teste utilizados e seus significados na organização fornecedor em uma tabela dentro desta seção.

Os seguintes níveis de gravidade são propostos para questões decorrentes ao longo do ciclo

de vida do teste:

- *Problema de teste Nível 1 – impede que um elemento crítico dos Serviços de Confiança funcione ou seja realizado, o que tem um impacto direto ou indireto sobre os pacientes e/ou usuários finais.*
- *Problema de teste Nível 2 – todos os elementos dos Trust Services ainda podem funcionar com uma solução alternativa, porém a funcionalidade ou o desempenho são severamente afetados;*
- *Problema de teste Nível 3 – todos os elementos dos Trust Services ainda podem funcionar com uma solução alternativa, porém a funcionalidade ou o desempenho necessários são impactados materialmente;*
- *Problema de teste Nível 4 – todos os elementos dos Trust Services ainda podem funcionar, porém há um impacto menor de funcionalidade/desempenho; e*
- *Problema de teste Nível 5 – todos os elementos dos Trust Services ainda podem funcionar, porém há pequenos defeitos cosméticos sem impacto funcional e sem impacto sobre pacientes ou serviços clínicos.*

2.4. Estágios de teste

Cada etapa de teste é uma forma discreta de teste com seus próprios objetivos, métodos e requisitos de cobertura e, portanto, um conjunto de seus próprios scripts de teste.

Uma matriz de cobertura de todas as etapas de teste / áreas de teste a serem cobertas em cada Versão de Teste é anexada abaixo. Isso está sujeito a acordo.

Mostre se os testes de módulo e sistema devem ser entregues juntos ou separadamente. Também mostre se os testes de operação e escalabilidade prontos devem ser entregues juntos ou separadamente.

Áreas de teste/ Tipo de teste	Teste central	Testes de Módulo & Sistema	Testes de Integração	Modelo de comunidade / testes de aceitação do usuário	RFO / Testes de escalabilidade	Preparação para testes ao vivo
Funcional		✓	✓	✓		
Não-Funcional	✓					
Processos de Negócios				✓		
Volume		✓	✓		✓	
Desempenho		✓	✓		✓	
Segurança (incluindo testes de penetração)			✓			
Proteção de dados		✓	✓			
Usabilidade		✓	✓	✓		
Testes de interface		✓	✓			

Áreas de teste/ Tipo de teste	Teste central	Testes de Módulo & Sistema	Testes de Integração	Modelo de comunidade / testes de aceitação do usuário	RFO / Testes de escalabilidade	Preparação para testes ao vivo
Instalação e Configuração		✓	✓			✓
Systems & Service Management & Service Level Reporting					✓	
Digno de Rede					✓	
Recuperação de desastres					✓	
Ferramentas e Processos do Helpdesk					✓	
Relatórios de informações gerenciais					✓	
Auditoria			✓			
Resiliência					✓	
Planejamento de Capacidade					✓	
Migração de dados			✓			
Processos de Treinamento, Conteúdos e Eficácia				✓	✓	
Cutover & Fallback & Go-Live Simulation					✓	
Backup, recuperação, diário					✓	
Processos de suporte de operações					✓	
Comissionamento						✓

2.4.1. Teste de módulo e sistema

Isso será conduzido pelo Fornecedor no próprio ambiente de desenvolvimento e teste do Fornecedor. Está previsto que isso será concluído antes da implantação nos sites do Trust.

2.4.2. Teste de integração

Os sistemas de TI que requerem interligação com o Painel Clínico serão definidos na

documentação de Definições de Dados.

Os testes formais de implantação necessários para serem realizados pelo Fornecedor são relativamente limitados em escopo, na forma de que o Fornecedor é obrigado a testar que o Painel Clínico é capaz de receber mensagens e dados com sucesso dos sistemas de dados Trust.

Serão utilizados scripts que refletem os fluxos de dados durante os quais os processos de carga de dados são monitorados.

2.4.3. Pronto para testes de operações

Isso inclui testes de configuração local e testes de escalabilidade/desempenho. Também haverá previsão para que uma verificação de Segurança de Saúde de TI seja realizada de acordo com o Requisito 3.13 dos Requisitos esp IG, elaborado na Nota de Orientação NPFIT-FNT-TO-IG-PRJMGT-0031 e NPFIT-FNT-TO-IG-DES-0162 IG Guidance Nota para Fornecedores ESP - Verificação de Saúde de Segurança de TI.

2.4.4. Teste de aceitação do usuário

Isso inclui testes de configuração locais e testes de usabilidade. A UAT será conduzida pouco antes do go-live e envolverá recursos de Fornecedor, CFH e Trust. A UAT será realizada em sites trust, usando o ambiente ao vivo. O Trust identifica e aloca os membros apropriados da equipe que participarão do processo global de teste. Estes incluirão, mas não necessariamente se limitam a:

- Pessoal de Mudança de Negócios
- Pessoal de treinamento
- Representantes dos usuários (como Especialistas em Assuntos)
- Pessoal de TI
- Pessoal de informação

O envolvimento dos treinadores terá as vantagens de fornecer:

- Uma oportunidade para os treinadores adquirirem conhecimentos práticos e experiência do sistema;
- Uma oportunidade para os treinadores entenderem os fluxos de trabalho;
- Um meio de auxiliar os treinadores de usuários finais no desenvolvimento subsequente de cursos de treinamento locais, no contexto de novas oportunidades de benefícios oferecidas pelo painel de instrumentos.

Nem todos os usuários serão treinados em todos os aspectos da funcionalidade, processos e procedimentos. Portanto, a seleção de usuários para atuar como testadores e 'mapeá-los' em scripts específicos, etc, precisará ser cuidadosamente considerada.

Isso não removerá a exigência de treinamento formal do usuário final. A abordagem do treinamento final do usuário será abordada no Plano de Estratégia de Treinamento e Treinamento.

2.4.5. Teste central (não funcional)

Isso irá incorporar todos os Testes Não Funcionais acordados e será realizado pelo Fornecedor. Esses testes serão realizados uma vez e as provas de testes bem-sucedidos realizadas em conjunto com um relatório de teste serão fornecidas ao CfH e aplicadas ao referido projeto de cada local para fins de aceitação.

2.5. Organização da equipe de teste

Estado que será dono e entregará todas as atividades de teste e entregas. Delineie a estrutura da Equipe de Gerenciamento de Testes e Teste mostrando responsabilidades de teste (pessoas e funções) – um gráfico de diagrama /organização / referências à documentação da Equipe de Teste são necessárias neste momento. Que nível de independência tem a Equipe de Teste? Forneça detalhes.

Todas as atividades de teste e entregas são de propriedade do Coordenador de Teste de Painel. A estrutura da Equipe de Teste ainda não foi finalizada, mas prevê-se que haverá uma equipe discreta de pessoal de Teste/QA. Eles serão empregados pelo Fornecedor, portanto não estão previstos para serem totalmente independentes.

2.5.1. Funções de execução de teste para todas as etapas

Palco Papel	Central	Sistema	Integração	Pronto para operações	Aceitação do usuário
Ambiente	Teste	Teste	Teste	Teste e para estar ao vivo	Teste e para estar ao vivo
Coordenação e Implementação	Fornecedor	Coordenador de Teste de Fornecedor + Desenvolvedores	Coordenador de Teste de Fornecedor + Desenvolvedores	Coordenação de fornecedores + Implementação do teste de confiança	Coordenação de fornecedores + Implementação do teste de confiança
Execução	Fornecedor	Especialistas em Produtos de Fornecedor	Especialistas em Produtos de Fornecedor	Trust Testers + Trust IT Staff + Trust iniciais Go Live	Teste de confiança.
Relatórios	Fornecedor	Fornecedor	Fornecedor	Fornecedor	Fornecedor
Apoio	N/A	Equipe de Projetos de Fornecedor	Equipe de Projetos de Fornecedor	Equipe de Projetos de Fornecedor	Equipe de Projetos de Fornecedor
Testemunhas		Confiar	Confiar		

2.5.2. Requisitos de suporte à equipe de teste

Identifique as equipes de suporte disponíveis para a Equipe de Teste, que inclui o fornecimento de soluções para problemas identificados, suporte do desenvolvimento no exterior, uso e envolvimento de Terceiros etc.

Fornecer detalhes da disponibilidade da equipe juntamente com os SLA's (Service Level Agreements, contratos de nível de serviço) que precisam ser cumpridos em torno de pessoal e suporte.

A equipe de teste será apoiada por uma equipe dedicada do Help Desk, equipada com a unidade de negócios do Fornecedor.

2.6. Avaliações e Inspeções

Inclua detalhes dos processos de Inspeção e passo a passo que serão empregados em todas as etapas de teste. Isso pode estar relacionado à documentação ou código recebido

pela Equipe de Teste, bem como a quaisquer entregas produzidas pela Equipe de Teste. As revisões devem ser realizadas internamente e quando apropriado externamente. As avaliações podem levar o formato de uma oficina.

2.6.1. Opiniões

Cada Etapa de Teste será executada de acordo com o Plano de Teste e Especificação de Teste aplicável a essa etapa. Cada documento será revisado internamente e submetido ao NHS CFH para revisão e aprovação.

2.6.2. Inspeções e Passo a Passo

2.7. Documentação do teste

Identifique toda a documentação do Programa de Teste (juntamente com suas referências individuais) que serão entregues durante cada uma das Fases de Teste e ciclos de teste, e ainda destaque as datas de entrega de contratos relevantes, quando aplicável.

Como exemplo, isso é definido como a tabela abaixo:

Documento	Horário
Estratégia de teste	Devido (<i>inserir não de dias</i>) dias após a assinatura do contrato
Planos de teste	Devido pelo menos (<i>inserir não de dias</i>) dias antes do início da Execução de Teste para esse tipo de teste
Especificações de teste	Devido pelo menos (<i>inserir não de dias</i>) dias antes do início da Execução de Teste para esse tipo de teste
Scripts de teste	Antes das atividades de execução para determinado tipo de teste
Relatório de teste	Rascunho devido (<i>não insira dias</i>) dias antes da conclusão desse tipo de teste. Versão de base com vencimento 1 dia após a conclusão do tipo de teste.
Aviso para iniciar os testes	
Registro de gerenciamento de problemas de teste	Relatórios disponíveis sob demanda

2.8. Planos de teste

Para cada Etapa de Teste, a cobertura específica do teste e as prioridades serão definidas dentro de um Plano de Teste que englobará o escopo detalhado e definirá os objetivos da prova para esta etapa de teste.

Um Plano de Teste será desenvolvido para cada Etapa de Teste para cada versão. Cada Plano de Teste será baseado nesta estratégia observando quaisquer diferenças e divergências necessárias devido às especificidades da fase/Liberação em teste.

O Plano de Teste deve ser revisado internamente antes de ser enviado ao NHS CFH para revisão e posterior aprovação.

2.9. Especificações de teste

As especificações de teste serão desenvolvidas para cada Estágio de Teste para cada versão, dando assim uma relação um-a-um entre o Plano de Teste e a Especificação de Teste.

Exceções a isso ocorreriam quando a complexidade de uma determinada Versão tornaria prudente dividir os Testes Não Funcionais, como Volume, Performance & Segurança dos Testes Funcionais. Qualquer divisão desse tipo seria detalhada dentro do Plano de Teste.

Descreva as técnicas de design de caso de teste a serem utilizadas. Para cada Fase de Teste, descreva a lógica para a seleção da técnica escolhida.

Todas as especificações de teste devem ser revisadas internamente antes de serem enviadas ao NHS CFH para revisão e posterior aprovação.

2.10. Scripts de teste

Os atributos de dados e as dependências associadas aos Scripts de Teste são os seguintes:

- Identificador exclusivo do script de teste
- Título
- Referência de exigência do NHS CFH a que o teste se relaciona
- Propósito do Roteiro de Teste
- Pré-requisitos
- Requisitos de dados
- Etapas de execução de teste em ordem sequencial dando resultados esperados associados
- Etapas para incluir quaisquer procedimentos de verificação necessários
- Indicador de Regressão (utilizado para indicar os testes considerados para cobrir os objetivos do teste central que formarão o pacote de regressão).

2.11. Execução de teste

Descreva como a Execução de Teste será organizada e o ambiente e o local onde isso ocorrerá. Uma atividade de execução típica será dividida nos seguintes conjuntos de execução:

- *funcionalidade principal*
- *novas funcionalidades detalhadas*
- *reteste para resolução de problemas de teste, tanto específico para correção quanto regressão*
- *Teste final de regressão*

2.11.1. Condução das Sessões de Teste

2.11.1.1. Início do dia

Cada dia em que os testes serão realizados começará com uma sessão de briefing para todos os participantes, delineando os planos para o dia, emissão de Scripts de Teste, etc.

2.11.1.2. Fim do dia

Haverá uma sessão de des briefing ("wash-up") no final de cada dia, para rever quaisquer problemas ou problemas encontrados e rever e concordar com as prioridades de quaisquer questões levantadas.

2.11.2. Realização de Testes – Adesão aos Scripts de Teste

Todos os testes estarão sujeitos a scripts formais. Não haverá variação dos scripts durante os testes formais, exceto quando os problemas forem encontrados com os próprios scripts (por exemplo, erros de digitação; o script instrui o usuário a usar um botão incorreto; o resultado esperado é descrito incorretamente; etc.). Acima de tudo, o bom senso deve prevalecer.

2.11.3. Registrando resultados reais versus resultados esperados

Os Relatórios de Teste serão atualizados com os resultados dos testes realizados. Qualquer evidência de suporte necessária ou fornecida e uma indicação sobre se a etapa de teste é

considerada como "Aprovada" ou "Falha". Isso será coletado em uma planilha do Excel.

2.11.4. Problemas de gravação

Os testes serão registrados por meio de um Registro de Teste, conforme descrito no Plano de Teste Detalhado. Este registro vai ser assinado pelo testador para certificar que o teste foi feito como roteirizado e quaisquer incidentes foram registrados. A Testemunha de Teste (se presente) assinará para confirmar que o testador executou o teste como roteirizado e registrou todos os incidentes de teste.

Um resumo de todos os Problemas, Defeitos, Observações ou Consultas decorrentes durante o Teste também será registrado no Registro de Teste, que terá um ID de referência de problemas ao Registro de Problemas Detalhados do Sistema Cs.

As Questões levantadas serão registradas no banco de dados da Pista de Testes do Fornecedor para investigação e resolução.

2.11.5. Escalonamento de Questões para resolução

Quaisquer problemas, defeitos, observações ou consultas decorrentes do teste, que não podem ser resolvidos localmente, precisarão ser formalmente escalados para o Fornecedor para resolução.

Sempre que possível, evidências adicionais (por exemplo, capturas de tela) serão fornecidas para auxiliar na resolução de problemas.

2.11.6. Priorizando questões

- Inicialmente, os Problemas de Teste (TIs) receberão uma classificação conforme especificado na Seção 2.3 acima.
- Na reunião diária de lavagem, essas classificações serão discutidas e alteradas quando necessário, com o nível adequado.
- Se um problema de teste crítico for identificado, todas as partes relevantes serão informadas.

2.11.7. Retestando

As liberações de software para corrigir defeitos descobertos durante o teste de implantação serão classificadas como de emergência ou rotina.

As liberações de software de emergência corrigem defeitos "críticos" que impedem os testes de continuar. Essas liberações serão recebidas e instaladas de acordo com o processo contratual e serão testadas imediatamente para que o restante dos testes possa continuar.

As liberações de software de rotina corrigem defeitos que não são defeitos "críticos". Estes serão tomados conforme acordado apropriado e retestado para completar os testes.

2.11.8. Funções de execução de teste

Testes centrais (não funcionais) são executados pelo Fornecedor

Palco	Sistema	Integração	Aceitação do	Pronto para
--------------	----------------	-------------------	---------------------	--------------------

Papel			usuário	operações
Ambiente	Teste	Teste	Teste e para estar ao vivo	Teste e para estar ao vivo
Coordenação & Gestão	Coordenador de Teste de Fornecedores + Desenvolvedores	Coordenador de Teste de Fornecedores + Desenvolvedores	Coordenador de Teste de Fornecedores + Coordenador de Teste de Confiança	Coordenador de Teste de Fornecedores + Coordenador de Teste de Confiança
Execução	Especialistas em Produtos de Fornecedores	Especialistas em Produtos de Fornecedores	Teste de confiança.	Trust Testers + Trust IT Staff + Trust iniciais Go Live
Relatórios	Fornecedor	Fornecedor	Fornecedor	Fornecedor
Apoio	Equipe de Projetos de Fornecedores	Equipe de Projetos de Fornecedores	Equipe de Projetos de Fornecedores	Equipe de Projetos de Fornecedores
Testemunhas	Confiar	Confiar		

2.11.9. Testar responsabilidades por atividade

Atividade	Responsabilidade / "Proprietário"	Dependência / Comentários
Desenvolvimento de especificações detalhadas do Plano de Teste e de Teste	Fornecedor	O Fornecedor cria os scripts para testes de sistema e integração. Esses cronogramas de teste fornecem scripts de teste genéricos, bem como um plano de teste de modelo, que pode ser dado ao Trust como ponto de partida para scripts de teste locais.
Desenvolvimento de scripts de teste de sistema	Fornecedor	Scripts de teste do sistema fornecidos pelo Fornecedor
Desenvolvimento de scripts de teste de usabilidade locais	Confiança apoiada pelo Fornecedor	Com base em scripts de teste obtidos a partir do Fornecedor e/ou novos scripts desenvolvidos do zero. Avaliação qualitativa baseada em testes e monitoramento dos resultados esperados em relação aos resultados reais.
Procedimentos locais do Help Desk	Confiança apoiada pelo Fornecedor	Os procedimentos locais do Help Desk exigem revisão e aprovação pelo Fornecedor, para garantir

		que os procedimentos locais estejam em conformidade com as exigências do fornecedor
--	--	---

Toda a execução do teste deve ser conduzida de forma transparente em um ambiente que espelha o ambiente ao vivo previsto o mais próximo possível. Os testes de sistema e módulo serão realizados no ambiente de teste do Fornecedor com Integração, Teste pronto para operações e aceitação do usuário realizado em sites trust.

2.12. Critérios de suspensão e retomada

Especifique os critérios utilizados para suspender toda ou uma parte da atividade de teste nos itens de teste associados a esta Estratégia.

Especifique as atividades de teste que devem ser repetidas, quando os testes são retomados.

A atividade de teste será suspensa se houver mais do que um certo número de questões "críticas" levantadas durante qualquer estágio de teste. A retomada estará condicionada a essas questões que foram tratadas para a satisfação dos recursos do Fornecedor, DO NHS CFH e Trust.

2.13. Critérios de entrada e saída

Dentro da Estratégia de Teste, os principais requisitos de alto nível para entrada e saída da fase de teste devem ser listados com referências apropriadas aos Planos de Teste de fase individual, onde os critérios de entrada e saída são descritos em termos mais específicos. Os critérios de saída são especialmente importantes para qualquer sistema crítico de segurança.

2.13.1. Tabela de Critérios de Entrada e Saída

2.14. Captura de resultados de teste

Descreva como os resultados serão capturados durante cada fase de teste.

Por exemplo, "Durante a execução do teste, o Diretor de Teste da Ferramenta de Gerenciamento de Teste será usado para capturar todos os resultados do teste".

Declarar o processo de qualquer divergência entre os resultados esperados e reais – estes devem ser investigados e uma questão de teste associada levantada.

Todos os resultados dos testes serão capturados pelos testadores em planilhas do Excel. Qualquer divergência entre os resultados esperados e reais será considerada como um problema. Os problemas serão registrados no Registro Diário de Problemas de Teste e todos os resultados serão tabulados para reportar ao NHS CFH no Relatório de Teste.

2.15. Arreios de teste

Os arreios de teste não devem ser necessários, pois todos os testes de módulo e sistema estão planejados para serem realizados em um ambiente de teste integrado usando dados de teste derivados dos feeds que serão usados no ambiente ao vivo. Quando os módulos e sistemas forem implantados em sites de Confiança, eles serão testados em um sub-local do ambiente ao vivo – devido à natureza do ambiente, áreas de teste e sites do Dashboard podem ser configurados sem comprometer o ambiente ao vivo.

2.16. Abordagem para testes de regressão

O teste de regressão é realizado quando uma parte componente do produto global é modificada e garante que nenhuma nova falha tenha sido introduzida como resultado desta modificação.

Com o uso de scripts de teste bem definidos e um formulário padrão para captura de dados, prevê-se que todos os testes serão repetitivos, o que tornará os testes de regressão muito mais simples.

2.17. Abordagem para testes de segurança

Os testes de segurança envolvem o teste dos níveis de acesso especificados na fase de projeto. As permissões do Active Directory especificadas serão testadas nas etapas de Aceitação do Sistema e do Usuário em termos de uso de contas de usuário de teste com as permissões apropriadas concedidas. Na fase Ready for Operations, o Trust estará testando que as permissões reais do Usuário concedidas dentro do Active Directory estão corretas para essas Funções do Usuário.

O Trust é responsável por decidir a extensão de qualquer teste de penetração que precise ser realizado e pela realização desses testes.

2.18. Engajamento cfh do NHS

Todos os documentos de teste serão submetidos ao NHS CFH para revisão e aprovação antes do início da fase necessária de testes. Os recursos cfh terão acesso a testes de testemunhas, conforme necessário.

O NHS CfH fornecerá todos os documentos NPFIT mencionados no documento "Common Assurance – Test Strategy" para permitir que a Equipe de Teste prepare os documentos de teste para um padrão e formato acordados.

3. Gerenciamento de testes

3.1. Gestão da Qualidade

O Coordenador de Testes é responsável por todos os procedimentos, normas e processos a serem adotados para todas as atividades de teste. É responsabilidade do Coordenador de Testes direcionar consultas de qualidade à pessoa ou equipe apropriadas para garantir que quaisquer problemas sejam resolvidos em tempo hábil.

3.2. Abordagem para gerenciamento de incidentes

3.2.1. Captura de problemas de teste

Os problemas de teste serão capturados em planilhas do Excel. Serão utilizados os seguintes atributos:

- Identificador único
- Nome do criador
- A questão do teste de data e hora foi levantada
- Fase de teste
- Número de compilação/liberação da linha de base
- Título e Descrição da Questão do Teste
- Nível de gravidade do problema de teste
- Contra o qual o Script de Teste o defeito foi encontrado
- Informações de suporte, por exemplo, capturas de tela

Todas as questões serão inscritas no Registro Diário de Questões de Teste, alocadas prioritárias e reportadas ao Coordenador de Testes para inclusão no Relatório de Teste.

3.2.2. Tratamento de Incidentes

Incidentes serão relatados diariamente. Todos os incidentes serão registrados no Registro de Problemas diários de teste e reportados ao Coordenador de Teste, responsável por garantir que os recursos do NHS CFH e trust sejam informados conforme necessário e para relatar quaisquer incidentes no Relatório de Teste.

3.3. Métricas de teste

Todos os testes vão ser realizados sob a supervisão direta do Coordenador de Testes. Se o Coordenador de Teste não puder viajar para um determinado local (por exemplo, se os testes forem realizados em mais de um local ao mesmo tempo), um subordinado deve ser delegado como Coordenador de Teste de Local.

3.4. Relatórios de progresso

3.4.1. Relatório de teste

Isso deve ser colhido e submetido ao final dos testes. O Coordenador de Testes ou seu adjunto nomeado é responsável por sua produção.

Os seguintes detalhes dos títulos necessários para o Relatório de Teste:

- Resumo da Gestão
- Resultados de execução de teste contra casos de teste
- Instruções sobre configuração do sistema e ambiente de teste.
- Testes que passaram e o requisito associado do NHS CFH
- Testes não realizados ou falhou, contra scripts e exigência do NHS CFH e razões pelas quais
- Questões de teste pendentes levantadas contra os requisitos do NHS CFH por prioridade/gravidade com planos de resolução conhecidos e avaliação de impacto funcional/técnico
- Questões de teste pendentes que não podem estar diretamente ligadas a um requisito específico e avaliação de impacto funcional/técnico, juntamente com um plano de resolução
- Medidas contra critérios de saída definidas no Plano de Teste

3.5. Iniciativas de Melhoria

A Equipe de Teste realizará uma "lavagem" na conclusão de cada etapa do Ciclo de Vida de Teste e será solicitada a enviar material para um Relatório de Lições Aprendidas ao Coordenador de Testes para inclusão no Relatório Final. Analisando as possíveis lições aprendidas ao final de cada etapa, elas serão então aplicadas na fase de sucesso.

3.5.1. Controle de testes

Esta estratégia de teste depende da Estratégia de Liberação do Fornecedor e da Estratégia de Gerenciamento de Configuração. A referência a esses documentos será incluída nesta seção. Esses documentos formarão a política em todas as versões e implantações em qualquer ambiente de teste para reconstruções de linha de base, patches de software e gerenciamento de alterações de configuração de todas as interdependências.

Dentro desta seção destaque como a Equipe de Teste confirmará o conteúdo de uma versão e a implantação do código em um ambiente de teste adequado.

Descreva como todo o material de teste será entregue e controlado na linha de base através do processo CM e do conjunto de ferramentas de suporte.

Todo o material de teste será proveniente do ambiente de desenvolvimento do Fornecedor. Este material só será liberado em sites trust para testes de integração quando tiver passado na fase de testes do Módulo/Sistema para a satisfação do Coordenador de Teste e quando a autoridade para liberar foi obtida do Gerente de Garantia da Qualidade e Gerente de Desenvolvimento de Produto responsável pelo Painel Clínico. O material de teste será entregue eletronicamente ou em CD-Rom, dependendo da quantidade de dados envolvidos.

4. Dados de teste

Descreva como e onde os dados de teste serão obtidos para cada fase de teste, ou seja, o teste do módulo receberá o conjunto de dados para um determinado módulo da equipe de desenvolvimento/ Fornecedor como parte da entrega de software com base.

Esclareça os requisitos para diferentes pacotes de dados de teste, ou seja, o pacote de dados volumosos será separado do conjunto de dados funcionais, pois os requisitos e a cobertura diferem e depende tanto da funcionalidade a ser exercida quanto do volume de dados necessários para suportar os testes.

Uma tabela (exemplo abaixo) resumindo a fonte dos dados do teste deve ser incluída:

Tipo de teste	Fonte de Dados de Teste
Teste de módulo e sistema	Dados manuais entregues a partir do fornecedor de software (teste de módulo) ou com base em dados de teste de módulo para cada aplicativo (teste do sistema). Os dados de teste do sistema devem garantir que as dependências de dados

	de aplicativos cruzados tenham sido consideradas e resolvidas.
Teste de Integração	NHS CFH entrega de dados 'anonimizados' com base em uma solicitação da Equipe de Teste. Para testes de V&P, NHS CFH entrega de dados 'anonimizados' em volume.
Pronto para operações / teste de escalabilidade	Cópia dos dados do teste de integração em volume apropriado
Comunidade modelo / UAT	NHS CFH entrega de 'anonimizado' conforme exigido pelo NHS CFH ou grupo de usuários apropriado
Preparação para o Teste Ao Vivo	NHS CFH entrega de dados 'anonimizados' com base em uma solicitação da Equipe de Teste. Além disso, dados do NHS central e dados locais do Trust NHS

5. Ambientes de teste

5.1. Especificação

5.1.1. Identificação dos componentes físicos, das comunicações, do sistema e do middleware necessários

O ambiente utilizado no local será composto por um servidor de aplicativos e um servidor de banco de dados. Os servidores de aplicativos e banco de dados serão instalados de acordo com as Notas de Instalação do Painel, que fazem parte do Kit de Ferramentas de Implementação do Projeto.

5.1.2. O modo ou estado de uso

Os servidores serão dedicados ao produto Clinical Dashboard.

5.1.3. Outros softwares ou suprimentos necessários para suportar testes

Para ser acordado

5.1.4. Requisitos de segurança e acesso à área de teste e equipamentos

Será necessário espaço no escritório com acesso exclusivo para a Equipe de Teste e recursos de Confiança/CfH.

5.1.5. Ferramentas de teste e utilitários necessários

Fornecedor usará Test Track para levantar, registrar e rastrear problemas desde a identificação até o fechamento

5.1.6. Qualquer outra necessidade de teste

Será necessário um PC cliente por testador, com o software apropriado instalado. Todos os documentos NHS CfH referenciados pelos recursos do NHS CfH devem ser disponibilizados

aos testadores.

5.1.7. Licenças adicionais de software

É responsabilidade do Trust obter quaisquer licenças adicionais de software que possam ser necessárias durante a fase de teste.

5.1.8. Suporte recursos e habilidades necessárias para manter os ambientes de teste

O suporte no local será fornecido pelo Trust com o apoio dos recursos do Fornecedor, se necessário

5.1.9. Identifique uma fonte para qualquer uma das necessidades que não estão disponíveis no momento para o grupo de teste

Os Gerentes de Projetos de Fornecedores e Trust de cada site se relacionarão de perto uns com os outros e garantirão que as necessidades sejam atendidas em tempo hábil, a fim de agilizar os testes.

5.1.10. Se algum outro projeto for fazer uso paralelo de qualquer um dos acima, consulte-o.

Não está previsto para ser aplicável

5.1.11. Informações sobre o uso de quaisquer stubs para imitar o processo de ponta a ponta

Não se prevê ser necessário, pois o sistema funciona de forma "de ponta a ponta".

5.1.12. Um modelo para definir a relação entre teste e ambiente ao vivo

O Ambiente de Teste será montado no escritório do Fornecedor em primeira instância e os testes do sistema serão realizados. Quando o ambiente é configurado no local e os testes do sistema são concluídos, o processo continuará com testes de integração. Uma vez que isso esteja concluído, será realizado um teste de aceitação do usuário e de operações, em paralelo ou em sucessão. Os testes serão realizados em uma área de teste dos servidores que eventualmente sediarão a versão ao vivo. Não está previsto que um banco de dados de teste separado seja necessário.

6. Ferramentas de teste

6.1. Ferramentas de gerenciamento de testes

O Test Track será usado para rastrear problemas, correções e resolução. Não está previsto que quaisquer outras ferramentas de teste sejam.

6.1.1. Problemas de teste

Os problemas de teste serão considerados como tendo surgido quando os produtos sob teste apresentarem comportamento contrário ou fora do requisito acordado. Qualquer desvio será registrado como um problema de teste na faixa de teste. A gestão dessas Questões de Teste será de propriedade da Equipe de Teste para garantir resolução, reteste e eventual encerramento.

6.2. Ferramentas de automação de testes

Para automação de teste de serviço está planejada a utilização do Postman. Para os testes de componentes e de ponta a ponta será utilizado o Cypress. Os testes serão integrados a um Pipeline para fornecer feedbacks rápidos aos desenvolvedores que poderão ter feedbacks rápidos das builds geradas durante o desenvolvimento.

APÊNDICE C - PLANO DE TESTES

Templates para criação de plano de testes, disponíveis em:

<https://www.mammoth-ai.com/how-to-write-a-test-plan-with-the-ieee-829-standard/>

Template escolhido pelo autor:

A seguir estão as seções do documento do plano de teste de acordo com as normas IEEE 829.

1. Identificador do plano de teste
2. Referências
3. Introdução
4. Itens de teste
5. Recursos a serem testados
6. Recursos não testados
7. Abordagem
8. Critérios de aprovação/falha
9. Critérios de suspensão
10. Resultados de teste
11. Tarefas de teste
12. Necessidades Ambientais
13. Responsabilidades
14. Necessidades de pessoal e treinamento
15. Agendamento
16. Riscos e Contingências
17. Aprovações

Vejam cada componente do Documento do Plano de Teste. Vamos apresentar o documento do plano de teste de acordo com as Normas IEEE 829.

Identificador do plano de teste:

O identificador do plano de teste é um número único para identificar o plano de teste.

Exemplo: ProjectName_0001

Referências:

Esta seção é para especificar toda a lista de documentos que suportam o plano de teste que você está criando atualmente.

Exemplo: SRS (Especificação de Requisitos do Sistema), Documentos de Caso de Uso, Estratégia de Teste, Plano de Projeto, Diretrizes de Projeto etc.,

Introdução:

Introdução ou resumo inclui o propósito e o escopo do projeto

Exemplo: O objetivo deste documento é testar a funcionalidade do 'ProjectName'

Itens de teste:

Uma lista de itens de teste que serão testados

Exemplo: Os testes devem ser feitos tanto no front-end, quanto no back-end do aplicativo nos ambientes Windows/Linux.

Recursos a serem testados:

Nesta seção, listamos todos os recursos que serão testados dentro do projeto.

Exemplo: Os recursos a serem testados são Página de Login, Dashboard, Relatórios.

Recursos para não serem testados:

Nesta seção, listamos os recursos que não estão incluídos no projeto.

Exemplo: O pagamento usando PayPal recursos está prestes a ser removido do aplicativo. Não há necessidade de testar esse recurso.

Abordagem:

A estratégia overall de como os testes serão realizados. Contém detalhes como Metodologia, Tipos de Teste, Técnicas de Teste. É explicado como o teste será realizado. Inclui detalhes sobre as fontes de dados de teste, entradas e saídas, procedimentos de teste e prioridades

Exemplo: Seguimos metodologia ágil neste projeto, utilizando BDD com cenários descritos em linguagem gherkin.

Critérios de aprovação/falha:

Nesta seção, especificamos os critérios que serão usados para determine pass ou fail percentagem de itens de teste.

Exemplo: Toda a funcionalidade principal do aplicativo deve funcionar como pretendido e a porcentagem de aprovação de casos de teste deve ser superior a 95% e não deve haver nenhum bug crítico.

Critérios de suspensão:

Nesta seção, especificamos quando parar os testes.

Exemplo: Se alguma das principais funcionalidades não estiverem funcionais ou o sistema tiver problemas de login, o teste deve ser suspenso.

Resultados de teste:

A lista de documentos precisa ser entregue em cada fase do ciclo de vida do teste. A lista de todos os artefatos de teste.

Exemplos: Casos de teste, relatório de bugs

Tarefas de teste:

Nesta seção, especificamos a lista de tarefas que precisamos concluir no projeto atual.

Exemplo: O ambiente de teste deve estar pronto antes da fase de execução do teste. O relatório do resumo do teste precisa ser preparado.

Necessidades ambientais:

Lista de hardware, software e quaisquer outras ferramentas necessárias para um ambiente de teste.

Responsabilidades:

Especificamos a lista de funções e responsabilidades de cada tarefa de teste.

Exemplo: O plano de teste deve ser preparado pelo Test Lead. A preparação e execução dos testes deve ser realizada pelos testadores.

Necessidades de pessoal e treinamento:

Planeje o curso de capacitação para aprimorar as habilidades de recursos no projeto para alcançar as metas desejadas.

Horário:

Detalhes completos sobre quando começar, terminar e quanto tempo cada tarefa deve ocorrer.

Exemplo: Realizar execução de teste – 120 horas-homem, Relatório de Teste – 30 horas-homem

Riscos e Contingências:

Nesta seção, especificamos a probabilidade de riscos e contingências para superar esses riscos.

Exemplo: *Risco* – Em caso de estimativa orçamentária errada, o custo pode ser excessivo. *Plano de Contingência* – Estabeleça o escopo antes de iniciar as tarefas de teste e preste atenção no planejamento do projeto e também acompanhe as estimativas orçamentárias constantemente.

Aprovações:

Quem deve assinar e aprovar o teste project

Exemplo: O gerente de projeto deve concordar com a conclusão do projeto e determinar as etapas a seguir em frente.

APÊNDICE D - MONITORAMENTO E CONTROLE DOS TESTES

Template com um exemplo de monitoramento de um conjunto de testes, disponível em: <https://www.guru99.com/how-you-can-achieve-project-goals-through-test-monitoring-control.html>

	A	B	C	D	E	F
1						
2	Plano de monitoramento					
3					Versão: 1.0	
4						
5						
6						
7	Histórico de revisão					
8		Data	Versão	Descrição	Autor	
9		25-jan.-18	1.0	Rascunho inicial	Marcelo	
10						
11						
12						

+ ≡ Histórico de revisão ▾ Informações do projeto ▾ Conteúdo ▾

Plano de monitoramento			
			Version: 1.0
Informações do Projeto			
Nome do projeto	Projeto Honbra		
Descrição	Projeto para logística de atendimento ao cliente e coleta de amostras de sangue		
Data:	9-mai.-22		
Dono(a) do produto:	Maria Silva		
Autor(a):	Marcelo Engenheiro de Testes		
Local:	Localização		
Informações do revisor(a)	Nome	Departamento	Posição
	Marcelo	Departamento de testes I	Engenheiro júnior
Informações do aprovador(a)	João	Departamento de testes I	Engenheiro sênior

Plano de monitoramento							
INFORMAÇÕES DE MÉTRICAS							
Número	Métricas	Tipo de métrica	Finalidade Coletada	Período para coleta	Avaliar Método	Critérios	Conclusão
1	Definir quais métricas precisam ser coletadas	Tipo de métrica	Propósito da coleção das métricas em questão	Frequência em que esta métrica precisa ser coletadas	Método para avaliação da métrica	Critério para aprovação da métrica	Conclusão a partir da métrica
2	Resultados dos testes no serviço (API) de recebimento de amostra.	Qualidade de software	Checar se o recebimento de amostras já pode ser executado em ambiente de produção.	Semanal	Checar se algum dos testes de API falhou.	100%	Se a quantidade de testes aprovados for menor que 100%, então, falta serem corrigidos os serviços para recebimento de amostra ser executado com sucesso.

APÊNDICE E - MODELAGEM E EXECUÇÃO DE TESTES

Exemplo de modelagem e execução de testes elaborado pelo autor:

Modelagem de Teste - Funcionalidade de recebimento de amostras.

Segue exemplo de modelagem de um teste de software caixa-preta utilizando a técnica de múltiplas partições. O contexto desse teste é de uma clínica que faz coleta e recebimento de amostras, uma das funcionalidades do sistema é a de receber amostras em lotes, onde várias amostras são selecionadas e o usuário, do time de atendimento, pode recebê-las em apenas um clique.

História de usuário:

[Como usuário] do time de atendimento [Eu quero] Poder realizar o recebimento de amostras coletadas do paciente em lotes [Para então] Eu não precisar fazer o recebimento de amostra de uma por uma, facilitando, portanto, meu trabalho.

[CT01]

Pré requisito - Estar logado em uma conta com perfil de atendimento

Nota: Executar testes com valores de X para 1, 10, 100, 1000

Dado que o usuário está logado na página de recebimento de amostras

Quando o usuário seleciona X amostras

E aperta no botão “receber amostras”

Então o status das amostras devem ser alteradas para “Recebida” e status 200 deve ser retornado pelo endpoint de “/receberAmostra”

Resultado do teste:

para os valores 1, 10 e 100 os resultados obtidos foram os esperados.

No entanto, para X com valor igual a 1000 as amostras não foram alteradas para “Recebida” e o endpoint de “/receberAmostra” retornou status 408

APÊNDICE F - QUESTIONÁRIO ACERCA DAS ÁREAS DE PROCESSO DO TMMi

- A SUA EMPRESA POSSUI UMA POLÍTICA DE TESTES?
- O TIME DE QUALIDADE DE SOFTWARE DESENVOLVE ESTRATÉGIAS DE TESTE?
- SÃO DESENVOLVIDOS PLANOS DE TESTE NA ORGANIZAÇÃO?
- HÁ MONITORAMENTO E CONTROLE DOS TESTES?
- EXISTE UM PROCESSO BEM ESTRUTURADO PARA MODELAGEM E EXECUÇÃO DOS TESTES?
- O AMBIENTE DE TESTE PERMITE A EXECUÇÃO DE TESTES GERENCIÁVEIS E REPETÍVEIS?