



Iniciativa da CNI - Confederação
Nacional da Indústria

SÉRIE TECNOLOGIA DA INFORMAÇÃO - SOFTWARE

TESTE DE SISTEMAS





Iniciativa da CNI - Confederação
Nacional da Indústria

SÉRIE TECNOLOGIA DA INFORMAÇÃO - SOFTWARE

TESTE DE SISTEMAS



CONFEDERAÇÃO NACIONAL DA INDÚSTRIA – CNI

Robson Braga de Andrade
Presidente

DIRETORIA DE EDUCAÇÃO E TECNOLOGIA – DIRET

Rafael Esmeraldo Lucchesi Ramacciotti
Diretor de Educação e Tecnologia

SERVIÇO NACIONAL DE APRENDIZAGEM INDUSTRIAL – SENAI

Conselho Nacional

Robson Braga de Andrade
Presidente do Conselho Nacional

SENAI – Departamento Nacional

Rafael Esmeraldo Lucchesi Ramacciotti
Diretor-Geral

Gustavo Leal Sales Filho
Diretor de Operações



*Iniciativa da CNI - Confederação
Nacional da Indústria*

TECNOLOGIA DA INFORMAÇÃO - SOFTWARE

TESTE DE SISTEMAS



© 2019. SENAI – Departamento Nacional

© 2019. SENAI – Departamento Regional de Santa Catarina

A reprodução total ou parcial desta publicação por quaisquer meios, seja eletrônico, mecânico, fotocópia, de gravação ou outros, somente será permitida com prévia autorização, por escrito, do SENAI.

Esta publicação foi elaborada pela equipe de Educação a Distância do SENAI de Santa Catarina, com a coordenação do SENAI Departamento Nacional, para ser utilizada por todos os Departamentos Regionais do SENAI nos cursos presenciais e a distância.

SENAI Departamento Nacional

Unidade de Educação Profissional e Tecnológica - UNIEP

SENAI Departamento Regional de Santa Catarina

Gerência de Educação

FICHA CATALOGRÁFICA

S491t

Serviço Nacional de Aprendizagem Industrial. Departamento Nacional.
Teste de sistemas / Serviço Nacional de Aprendizagem Industrial.
Departamento Nacional, Serviço Nacional de Aprendizagem Industrial.
Departamento Regional de Santa Catarina. Brasília : SENAI/DN, 2019.
143 p. : il. (Série Tecnologia da informação - Software).

ISBN 978 - 85 - 505 - 2423 - 8

1. Software – Controle de qualidade. 2. Engenharia de software. 3. Software - testes. I. Serviço Nacional de Aprendizagem Industrial. Departamento Regional de Santa Catarina. II. Título. III. Série.

CDU: 004.415.53

SENAI

Serviço Nacional de
Aprendizagem Industrial
Departamento Nacional

Sede

Setor Bancário Norte • Quadra 1 • Bloco C • Edifício Roberto
Simonsen • 70040-903 • Brasília – DF • Tel.: (0xx61) 3317-
9001 Fax: (0xx61) 3317-9190 • <http://www.senai.br>

Lista de Ilustrações

Figura 1 - Definições sobre a qualidade esperada de um software	16
Figura 2 - Processo de teste	17
Figura 3 - Relação entre o processo de teste e desenvolvimento	19
Figura 4 - Subetapas da preparação do ambiente de testes	24
Figura 5 - Diferença entre os testes de caixa branca e caixa preta	34
Figura 6 - Modelo V que apresenta testes de caixa branca preta	34
Figura 7 - Fluxograma da dinâmica RCA	42
Figura 8 - Hierarquia de profissionais de teste	44
Figura 9 - Ambientes de teste	46
Figura 10 - Principais pilares do plano de teste.....	52
Figura 11 - Principais características de qualidade de software.....	63
Figura 12 - Dimensões do teste.....	65
Figura 13 - Modelo V de desenvolvimento x teste	66
Figura 14 - Características do casos de teste.....	69
Figura 15 - Fluxo básico e alternativos de um caso de uso	69
Figura 16 - Conceitos gerais sobre ferramentas de automação de teste	77
Figura 17 - Teste de performance.....	78
Figura 18 - Teste funcional	78
Figura 19 - Escolha de testes para automação	81
Figura 20 - Seleção do Workspace do Eclipse	83
Figura 21 - Menu para criar novo item no Eclipse	83
Figura 22 - Menu para criar novo projeto Java	84
Figura 23 - Tela de cadastro do projeto Java	84
Figura 24 - Menu para criar novo item no Eclipse	85
Figura 25 - Menu para criar nova classe Java	85
Figura 26 - Tela de cadastro da classe Java.....	86
Figura 27 - Área de edição da classe Java.....	86
Figura 28 - Menu para criar novo item no Eclipse	87
Figura 29 - Menu para criar novo item não especificado no menu	87
Figura 30 - Tela de busca do novo item a ser criado no Eclipse.....	88
Figura 31 - Tela de cadastro do caso de teste JUnit	88
Figura 32 - Área de edição da classe de teste Java.....	89
Figura 33 - Menu para executar um item no Eclipse	89
Figura 34 - Seleção para executar o teste pelo JUnit.....	89
Figura 35 - Tela de resultado do teste pelo JUnit.....	90
Figura 36 - Área de edição da classe de teste Java.....	90
Figura 37 - Tela de resultado do teste pelo JUnit.....	91
Figura 38 - Tela de busca do novo item a ser criado no Eclipse.....	91
Figura 39 - Tela de cadastro da suíte de teste JUnit.....	92
Figura 40 - Tela de resultado do Teste pelo JUnit.....	93

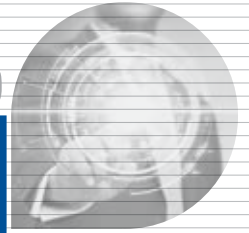
Figura 41 - Painel do XAMPP com os módulos Apache e MySQL em execução.....	94
Figura 42 - Seleção da linguagem para instalação.....	95
Figura 43 - Tela inicial do procedimento de instalação	95
Figura 44 - Tela para selecionar diretório de instalação do XAMPP	96
Figura 45 - Tela para definir usuário e senha da ferramenta e banco de dados	96
Figura 46 - Tela com opção de instalação da ferramenta em nuvem	97
Figura 47 - Tela para iniciar a execução da instalação	97
Figura 48 - Tela para mostrar o progresso da instalação	98
Figura 49 - Tela inicial da ferramenta a partir de acesso local	98
Figura 50 - Tela da ferramenta para cadastro de projeto de teste	99
Figura 51 - Tela da ferramenta com exemplo de projeto de teste.....	99
Figura 52 - Tela da ferramenta para cadastro de requisito de teste	99
Figura 53 - Tela da ferramenta para cadastro de caso de teste.....	100
Figura 54 - Tela de instalação do Selenium IDE	101
Figura 55 - Tela de administração de extensões do Firefox.....	101
Figura 56 - Tela inicial do Selenium IDE.....	102
Figura 57 - Tela inicial do Selenium com a definição do projeto de teste.....	103
Figura 58 - Tela do Selenium para cadastro de novo caso de teste.....	103
Figura 59 - Tela inicial do Selenium com a definição do caso de teste	104
Figura 60 - Definição do endereço do sistema que será testado	104
Figura 61 - Script gerado após gravação do teste com as ações do testador	105
Figura 62 - Resultado da execução do Script de teste	105
Figura 63 - Teste de performance em granularidade menor	106
Figura 64 - Estrutura de arquivo do JMeter.....	108
Figura 65 - Tela com interface CLI do JMeter.....	108
Figura 66 - Tela com interface gráfica do JMeter	109
Figura 67 - Definição do plano de teste	109
Figura 68 - Cadastro de grupo de usuários	110
Figura 69 - Configuração do grupo de usuários.....	110
Figura 70 - Cadastro de requisições HTTP	111
Figura 71 - Configuração da requisição HTTP	111
Figura 72 - Cadastro de asserção de resposta.....	112
Figura 73 - Configuração da asserção de resposta.....	112
Figura 74 - Criação do relatório de sumário.....	113
Figura 75 - Configuração do relatório de sumário	113
Figura 76 - Criação do gráfico de resultados	113
Figura 77 - Gráfico de resultados	114
Figura 78 - Criação da árvore de resultados	114
Figura 79 - Árvore de resultados.....	114
Figura 80 - Execução do teste no JMeter	115
Figura 81 - Resultado do teste no relatório de sumário	115
Figura 82 - Resultado do teste no gráfico de resultados.....	115

Figura 83 - Resultado do teste na árvore de resultados	116
Figura 84 - Estrutura do relatório de log de teste	122
Figura 85 - Estrutura do relatório de incidente de teste.....	123
Quadro 1 - Relação dos processos de teste e desenvolvimento	18
Quadro 2 - Relação de Fatores de qualidade de software	40
Quadro 3 - Níveis de severidade para a análise de causa raiz	42
Quadro 4 - Relação de perfis de profissionais de teste.....	44
Quadro 5 - Relação de atividades por profissional de teste	45
Quadro 6 - Relação de tipos de testes por equipes responsáveis	45
Quadro 7 - Fluxo do gerenciamento de risco.....	48
Quadro 8 - Relação de itens dos elementos dos modelos de plano de teste	55
Quadro 9 - Vantagens e desvantagens da equipe de desenvolvimento na função de testadores	57
Quadro 10 - Vantagens e desvantagens da equipe independente de testes	57
Quadro 11 - Relação de tipos de testes x responsáveis.....	70
Tabela 1 - Exemplo da relação de fatores de qualidade e riscos	64

Sumário

1 Introdução	13
2 Visão Geral do Teste de Software	15
2.1 Processo de Testes.....	16
2.1.1 Etapas do Processo de Teste	19
2.2 Tipos de Testes.....	33
2.2.1 Tipos de Testes Específicos	35
2.3 Gestão do Processo de Testes	39
2.4 Melhorias no Processo de Testes	41
2.5 Infraestrutura de Teste.....	43
2.5.1 Papéis e Responsabilidades de Teste.....	44
2.5.2 Ambiente de Teste	46
2.6 Análise de Risco	47
3 Planejamento e Execução dos Testes	51
3.1 Desenvolvimento do Plano de Testes	52
3.1.1 Tarefas para desenvolver o planejamento	56
3.1.2 Estratégia de Teste	62
3.2 Elaboração do Teste.....	67
3.2.1 Documento de Teste	67
3.2.2 Plano de Caso de Teste	68
3.3 Execução do Teste	70
3.3.1 Execução por Tipo de Teste	71
4 Automação do Processo de Teste	75
4.1 Conceitos de Automação de Testes.....	76
4.1.1 Tipos de Testes Automatizados.....	79
4.1.2 Testes Progressivos e Regressivos	80
4.1.3 Drivers x Stubs	80
4.1.4 Testes Candidatos à Automação.....	80
4.2 Automação de Testes Unitários.....	81
4.3 Automação de Gerência e Planejamento de Testes	93
4.4 Automação de Testes Funcionais	100
4.5 Automação de Testes de Performance	106
5 Relatórios de Teste.....	119
5.1 Tipos de Relatórios de Testes.....	120

6 Virtudes Profissionais, Organização e Qualidade	127
6.1 Virtudes profissionais.....	128
6.1.1 Organização	128
6.1.2 Planejamento e administração do tempo	128
6.1.3 Iniciativa e autonomia.....	131
6.2 Organização do trabalho.....	132
6.3 Qualidade.....	134
6.3.1 Qualidade total	135
Referências.....	139
Minicurriculo do Autor	141
Índice	143



Olá, seja bem-vindo à unidade curricular Teste de Sistemas.

Nesta unidade curricular, você irá obter uma visão geral sobre este tema e verá os principais conceitos de teste de sistemas, envolvendo as diversas etapas das atividades relacionadas aos profissionais de teste de software. Mas como essas etapas são constituídas? São constituídas pelo processo de teste, que irá nortear todas as atividades de teste, assim como irá acompanhar a gestão desse processo e possibilidades de melhorias. Ainda nas etapas mencionadas, você conhecerá os diferentes tipos de testes, ambientes de testes, entre outras técnicas.

O planejamento de testes também compõe as etapas do teste, e você acompanhará a forma de desenvolvimento de documentos de teste, como o plano de teste, o documento de teste e demais documentos relacionados.

Saiba que a cobertura de teste é algo de grande importância nos testes de sistemas, mas que é limitada por questões como cronograma e custos do projeto. Dessa forma, uma maneira de aumentar a cobertura, sem comprometer o projeto, é automatizar os testes, e, aqui, você terá a oportunidade de estudar conceitos relacionados à automação de teste. Por fim, ao concluir as etapas do teste, os resultados dos testes de sistemas precisam ser apresentados. Para essa apresentação, também existe um procedimento, cujas formalizações e desenvolvimentos de documentos de conclusão você vai ver durante o seu estudo.

Agora que você já tem uma ideia do conteúdo que será estudado, siga em frente e descubra o quão interessante é esse universo de teste de software e observe sua grande importância no desenvolvimento de sistemas de qualidade.

Bons estudos!

Visão Geral do Teste de Software

2



Neste capítulo você identificará os passos do processo de teste, tendo uma visão mais clara do motivo pelo qual os testes são executados e de que forma podem ser realizados, seguindo um processo eficiente.

Antes de abordarmos diretamente o processo de teste, é interessante que você conheça a sua evolução, que também está relacionada com o desenvolvimento de softwares. Dessa forma, como a arquitetura dos sistemas geralmente propõe a composição de diferentes partes ou componentes, que interagem entre si, para que o objetivo do software seja alcançado, ou seja, para que o sistema realize o seu propósito, é importante buscar uma garantia de seu funcionamento.

Com base nessa afirmação, pode-se prever que para a correta execução do software, além dos testes unitários, realizados no processo de desenvolvimento do sistema, também é necessário testar o software sob outras perspectivas. Assim, é possível garantir que as integrações entre os diferentes componentes do sistema estejam funcionando de forma correta, bem como analisar os requisitos funcionais e não funcionais, entre outras técnicas e procedimentos de testes.

E você sabia que essa preocupação sobre a importância do teste de forma mais abrangente é relativamente recente? É isso mesmo! Há algumas décadas, o teste de software era considerado um “mal necessário”, sendo realizado de forma superficial, apenas para demonstrar o funcionamento aos seus clientes.

Nos tempos atuais, com o aumento exponencial da complexidade e novas funcionalidades inerentes às novas necessidades dos softwares, aliado à competitividade do mercado, os testes de software passaram a desempenhar um papel fundamental no processo de desenvolvimento, ao ponto de constituir um novo processo, o processo de teste.

Mas lembre-se que nesse processo o principal objetivo é a validação de forma mais aprofundada do correto funcionamento do sistema, com base na especificação dos seus requisitos funcionais e não funcionais.

No entanto, é importante ressaltar que apesar da abordagem mais aprofundada dos testes, não é viável testar todo o sistema ou garantir que este não terá defeitos, tendo em vista o tempo que seria necessário para realizar tal tarefa e os custos associados ao projeto de desenvolvimento. Ao longo do seu estudo, serão apresentados argumentos mais detalhados que justificam o fato do teste não cobrir todo o sistema, assim como métodos para garantir o máximo de cobertura possível.

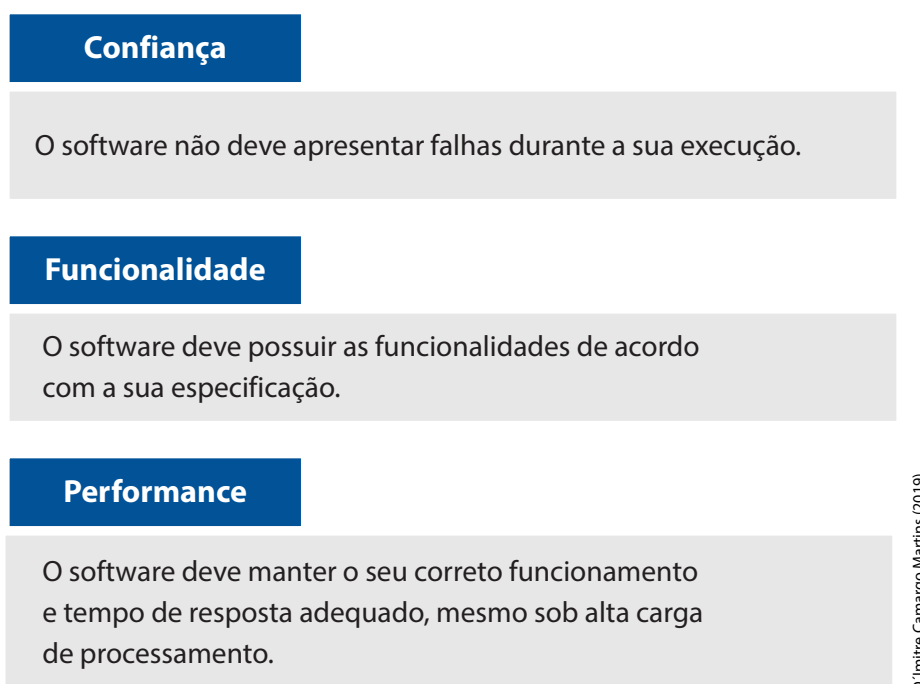
Ao final deste capítulo, você terá subsídios para:

- a) conhecer as etapas do processo de teste;
- b) conceituar os diferentes tipos de teste;
- c) entender a gestão do processo de teste;
- d) propor melhorias no processo de teste;
- e) reconhecer a infraestrutura de testes;
- f) identificar os papéis e responsabilidades da realização de teste.

Veja, a seguir, as etapas do processo de teste e entenda como é percorrido o fluxo desse processo, desde os procedimentos iniciais até a entrega final dos resultados do teste.

2.1 PROCESSO DE TESTES

A realização de testes de software tem como objetivo conferir qualidade aos sistemas, por isso é importante pontuar as três definições sobre a qualidade esperada de um software submetido a testes. Observe!



D'Imitre Camargo Martins (2019)

Figura 1 - Definições sobre a qualidade esperada de um software
Fonte: Do autor (2019)

Os testes são realizados de forma estruturada, por meio do processo de teste, que pode apresentar diferenças entre as organizações, para que reflitam suas reais necessidades. Porém, de forma geral, tendem a seguir passos e padrões, identificados como:

**FIQUE ALERTA**

É importante destacar que esta etapa ocorre paralela às etapas de Especificação, Execução e Entrega, tendo em vista as atividades de revisão que são necessárias nas etapas mencionadas.

Na Preparação, ocorre a disponibilização das condições de trabalho, ou seja, o ambiente de teste é disponibilizado, considerando os equipamentos necessários, profissionais envolvidos e ferramentas de forma geral.

Na etapa de Especificação são construídos os Casos de Testes, são escritos os códigos para testes automatizados, é desenvolvido o Roteiro de Testes e aplicado os testes estáticos, testes estes que revisam a documentação de especificação do software.

Já na Execução, os testes dinâmicos são aplicados de acordo com o planejamento prévio, seguindo os passos do Roteiro de Testes e registrando os resultados obtidos.

Chegar na etapa final, a Entrega, significa que todos os testes foram concluídos e que o sistema estará apto, de acordo com o objetivo geral do teste, para ser disponibilizado no ambiente de produção.

E veja bem! Tendo em vista os altos custos de manutenção de software, associados aos produtos em suas fases finais de desenvolvimento, a realização de testes já é iniciada em conjunto com o processo de desenvolvimento. Assim, as correções de defeitos podem ser realizadas com a máxima antecedência.

Nesse cenário, no quadro seguinte você vai ver uma relação entre as etapas do processo de teste e de desenvolvimento e as atividades realizadas em paralelo nos dois processos. Acompanhe!

PROCESSO DE TESTE	PROCESSO DE DESENVOLVIMENTO	AÇÕES
Planejamento	Planejamento	Integração dos planos Preparação da estratégia e plano de testes
Especificação	Projeto lógico e físico	Revisão dos planos de testes Desenvolvimento e revisão dos Casos de testes e Roteiro de testes Atualização do projeto de desenvolvimento
Execução	Implementação	Busca de defeitos e correções
Execução	Implantação	Busca de defeitos e correções

Quadro 1 - Relação dos processos de teste e desenvolvimento
Fonte: Molinari (2008)

**CURIOSIDADES**

Você sabe qual a diferença entre **Verificação** e **Validação** no processo de teste? A **Verificação** tem como propósito inspecionar ou revisar os artefatos gerados pelas etapas do processo de teste, enquanto a **Validação** irá atestar se o software atende às especificações de requisitos, validando o seu correto funcionamento.

Ainda no cenário no qual os processos de teste e desenvolvimento ocorrem em paralelo, o ponto de partida do ciclo de testes é a gestão de requisitos de um projeto de software, por meio do planejamento dos testes, baseado no documento de requisitos do futuro sistema.

Esse ponto de partida reforça a dinâmica do processo de teste, que deve ocorrer paralelo ao desenvolvimento, pareando as suas etapas com as diversas etapas da construção do software.

Na próxima figura é possível observar a relação entre os diferentes processos, com as atividades de teste executadas entre as etapas que se relacionam no teste e desenvolvimento, a partir da definição da gerência de requisitos.

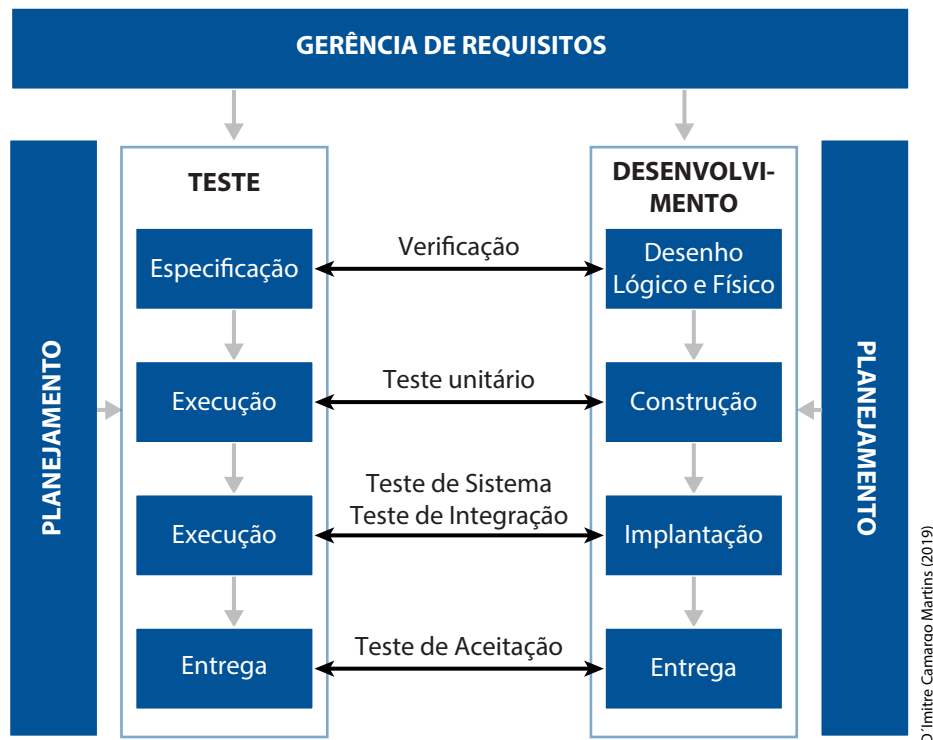


Figura 3 - Relação entre o processo de teste e desenvolvimento
Fonte: Do autor (2019)

Agora que você viu os conceitos gerais sobre o processo de teste, chegou a hora de aprofundar o conhecimento sobre esse interessante assunto. Para isso, acompanhe os próximos tópicos para conhecer mais sobre as etapas e subetapas que compõe o processo de teste.

2.1.1 ETAPAS DO PROCESSO DE TESTE

Na dinâmica do processo de teste de software, organizada nas etapas de Procedimentos Iniciais, Planejamento, Preparação, Especificação, Execução e Entrega, possui nesta organização atividades e atribuições bem definidas. Portanto, é oportuno que sejam apresentadas e detalhadas individualmente.

Para essa abordagem tornar-se mais clara e objetiva, cada etapa e subetapa do ciclo de testes será apresentada por itens que irão defini-las e caracterizá-las, sendo estes:

Objetivos: Descrição do que será executado ou criado na etapa.

Artefatos de entrada: Descrição de artefatos que permitirão o início da etapa ou subetapa.

Artefatos produzidos: Descrição dos artefatos produzidos após a realização das atividades da etapa ou subetapa.

Atividades: Lista das principais atividades, para alcançar os objetivos e produzir os artefatos da etapa ou subetapa.

Acompanhe nos próximos tópicos as etapas do processo de teste, de acordo com a abordagem apresentada.

PROCEDIMENTOS INICIAIS

Nesta etapa inicial é necessário avaliar os documentos de requisitos do software a ser desenvolvido, pois irá determinar os testes necessários para o projeto.

Após essa análise, deverá ser criado o Guia Operacional de Teste (GOT), que será um documento com as principais atividades a serem realizadas no projeto de teste, incluindo os recursos necessários, como profissionais e ambientes de teste.



phongphan5922 (2017)

Saiba que esse documento, o GOT, também representará um acordo de nível de serviço entre as equipes de testes, desenvolvimento e usuários. Sendo assim, a aprovação do GOT irá definir o início das atividades de testes previstas em seu projeto.

A partir de agora, você é convidado a conhecer os itens que caracterizarão a criação do GOT.

Guia Operacional de Testes

Objetivos:

- a) Definir profissionais responsáveis pelo teste
- b) Definir os ambientes de teste
- c) Definir as responsabilidades das equipes de execução de teste
- d) Criar um plano de trabalho
- e) Realizar avaliação inicial de riscos
- f) Definir os níveis de serviços acordados entre as equipes

Artefatos de entrada:

- a) Requisitos de negócio
- b) Modelos preliminares de dados
- c) Diagramas de fluxo de dados
- d) Demais documentos de desenvolvimento que impactarão na execução dos testes

Artefatos produzidos:

- a) Guia Operacional de Teste (GOT)

Atividades:

- a) Planejar entrevistas sobre requisitos de negócio, com os usuários do novo sistema
- b) Identificar e analisar os requisitos de negócio
- c) Planejar entrevistas com a equipe de desenvolvimento
- d) Definir estimativas iniciais do projeto de teste
- e) Criar o documento GOT
- f) Definir o grau de criticidade do projeto de teste
- g) Definir os parâmetros de qualidade do software e desempenho do processo
- h) Relacionar os principais riscos do negócio
- i) Aprovar o GOT com a equipe de desenvolvimento e usuários

Depois de ver o guia operacional de testes, vamos ao planejamento.

PLANEJAMENTO

Nesta etapa, deverão ser planejadas a abordagem e a maneira de realizar os testes previstos no projeto de teste. Para isso, deverão ser desenvolvidos dois documentos principais: a Estratégia de Teste e o Plano de Teste.



FIQUE ALERTA

É importante destacar as pré-condições que viabilizarão o desenvolvimento dos documentos mencionados nesta etapa. Essas pré-condições são: ocorrer em conjunto com o levantamento de requisitos e planejamento do projeto de desenvolvimento, verificar os principais requisitos do novo sistema e a conclusão da análise de risco do projeto de teste.



Pinkypills (2018)

A etapa de planejamento possui duas subetapas, que são: criar Estratégia de Testes e criar Plano de Testes. A seguir, serão listados os itens que irão caracterizá-las, na abordagem já mencionada.

Estratégia de Testes

Objetivos:

- a) Elaborar a Estratégia de Testes.
- b) Definir prioridades dos testes de acordo com a liberação e riscos.

Artefatos de entrada:

- a) Requisitos de negócios do sistema.
- b) Documento de planejamento do sistema.
- c) GOT.
- d) Modelos de dados e demais modelos de diagramas do projeto de desenvolvimento.

Artefatos produzidos:

- a) Estratégia de Testes do Projeto.
- b) Análise de riscos do projeto de testes.

Atividades:

- a) Elaborar a análise de riscos do projeto de testes.
- b) Buscar a redução dos riscos do negócio, por meio dos testes.
- c) Avaliar requisitos e planejamento do sistema.
- d) Avaliar os modelos de dados e diagramas do projeto de desenvolvimento.
- e) Identificar indicadores de qualidade e desempenho.
- f) Definir o que será verificado no sistema.
- g) Criar o documento Estratégia de Testes do Projeto.
- h) Definir os tipos de testes que serão realizados no desenvolvimento e implantação do sistema.
- i) Definir as prioridades de testes de partes do sistema.
- j) Definir a infraestrutura a ser utilizada no projeto de teste.
- k) Definir os ambientes de testes.
- l) Verificar a necessidade de testes estáticos.
- m) Validar a estratégia de testes com as equipes envolvidas no desenvolvimento do sistema.
- n) Divulgar a estratégia de testes para os profissionais de teste envolvidos no projeto de teste.
- o) Selecionar as técnicas de testes que serão usadas no projeto de teste.

Plano de Testes

Objetivo:

- a) Criar o documento de Plano de Testes, de acordo com o projeto de testes.

Artefatos de entrada:

- a) Estratégia de testes.
- b) Análise de risco do projeto de testes.
- c) Necessidades de dados de testes.
- d) Planejamento de desenvolvimento do sistema.

Artefatos produzidos:

- a) Plano de Testes.
- b) Análise de riscos do projeto de testes.

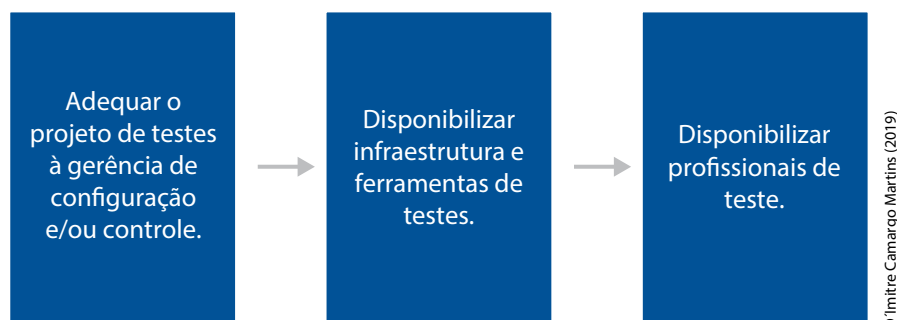
Atividades:

- a) Criar o Plano de Teste.
- b) Revisar a análise de riscos do projeto de testes.
- c) Alinhar o Plano de Teste ao Planejamento do Sistema.
- d) Aprovar o Plano de Teste.
- e) Definir as datas de execução dos testes unitários, de integração, de sistema e aceitação.
- f) Definir os indicadores de qualidade e desempenho.
- g) Medir o tamanho do sistema a ser testado em Pontos por Função.
- h) Estimar o esforço dos testes em Pontos de Teste.
- i) Preparar o cronograma para revisões definidas na Estratégia de Teste.
- j) Prever os recursos necessários para execução dos testes.
- k) Definir as responsabilidades para realizar atividades e testes específicos no projeto de testes.
- l) Identificar as necessidades de dados para os testes.
- m) Definir relatórios de acompanhamento do processo de teste e de controle de correções realizadas.
- n) Divulgar o Plano de Testes para a equipe de testes.
- o) Definir as regras para classificação dos defeitos e de sua severidade.

PREPARAÇÃO

Na etapa de Preparação ocorrerá a disponibilização do ambiente de testes, incluindo os equipamentos, profissionais de teste, ferramentas de automação, hardware e software específicos de testes.

A preparação do ambiente de testes ocorrerá em subetapas, as quais poderão ser executadas em paralelo às etapas iniciais do projeto de teste, como mostra a figura seguinte.



D'Imitre Camargo Martins (2019)

Figura 4 - Subetapas da preparação do ambiente de testes
Fonte: Do autor (2019)

Adequar o projeto de testes à Gerência de Configuração e/ou Controle

Objetivos:

- a) Adequar o projeto de teste à Gerência de Configuração e/ou Controle.
- b) Facilitar o compartilhamento de recursos com a equipe de desenvolvimento, por meio da Gerência de Configuração.

Artefatos de entrada:

- a) Arquitetura básica do ambiente de desenvolvimento do sistema.
- b) Arquitetura básica do ambiente de produção do sistema.
- c) Procedimentos e ferramentas para a Gerência de Configuração e/ou Controle.

Artefatos produzidos:

- a) Registro e controle das versões dos produtos: funcional, desenvolvimento, produto e operacional.

Atividades:

- a) Definir o nível de controle de versão e controle de mudanças necessárias para o projeto.
- b) Definir os perfis de acesso.
- c) Definir os processos de sincronização entre as equipes de teste e desenvolvimento.

Disponibilizar Infraestrutura

Objetivo:

- a) Disponibilizar a infraestrutura e ferramentas necessárias para a execução do projeto de teste.

Artefatos de entrada:

- a) Estratégia de Testes.
- b) Planos de Teste.
- c) Arquitetura mínima do ambiente de desenvolvimento do sistema.
- d) Arquitetura mínima do ambiente de execução do sistema.
- e) Ferramentas de teste necessárias para a execução dos testes planejados.

Artefatos produzidos:

- a) Infraestrutura e ferramentas disponíveis para o projeto de teste.

Atividades:

- a) Instalar hardware e software dos ambientes.
- b) Avaliar a estratégia do Plano de Teste.
- c) Configurar ambiente de acordo com as especificações do projeto.
- d) Definir perfis de uso do ambiente.
- e) Resolver dificuldades de acesso a informações e processos.
- f) Definir a origem da massa de testes e como serão disponibilizados.
- g) Preparar ambientes de teste, de acordo com o ambiente necessário para a execução do sistema em ambiente de produção.

Disponibilizar Profissionais de Teste

Objetivo:

- a) Disponibilizar equipe técnica competente para a execução do projeto de testes.

Artefatos de entrada:

- b) Estratégia de Testes.
- c) Plano de Testes.
- d) Ferramentas de testes usadas na execução dos testes.
- e) Arquitetura do ambiente de teste.

Artefatos produzidos:

- a) Equipe de teste competente e capacitada para a execução do projeto de teste.

Atividades:

- a) Avaliar estratégia e plano de testes. Identificar as características do ambiente de teste.
- b) Identificar as características das ferramentas de testes usadas.
- c) Avaliar a disponibilidade e experiência da equipe de testes.
- d) Identificar possíveis necessidades de treinamento para a equipe de testes.
- e) Fornecer treinamento para as equipes de testes, caso seja necessário.
- f) Medir o tamanho do sistema em pontos de teste ou outra métrica mais adequada ao projeto de testes.



Vasyi Dolmatov (2017)

ESPECIFICAÇÃO

Nesta etapa do processo, essencialmente, serão definidos e revisados os casos de testes e roteiros de testes, sendo que essas ações serão realizadas durante o desenvolvimento do sistema, de acordo com as entregas realizadas pela equipe de desenvolvimento. Veja como criar casos de testes.

Criar Casos de Testes

Objetivo:

- a) Criar e revisar os Casos de Testes do Projeto de Teste.

Artefatos de entrada:

- a) Estratégia de Teste.
- b) Plano de Teste.
- c) Necessidades de dados para testes.
- d) Documentação técnica do sistema.
- e) Checkpoint da realização de testes.

Artefatos produzidos:

- a) Casos de Testes.
- b) Especificação de dados para testes.
- c) Roteiros de testes automatizados (scripts).

Atividades:

- a) Criar os Casos de Testes.
- b) Revisar e alterar Casos de Testes.
- c) Criar roteiros de testes automatizados, se for necessário.
- d) Especificar as necessidades de dados de testes.

Criar Roteiros de Testes

Objetivo:

- a) Criar e revisar os Roteiros de Testes.

Artefatos de entrada:

- a) Casos de Testes.
- b) Plano de Testes.
- c) Fluxo de execução do sistema e/ou Casos de Uso da especificação do sistema.

Artefatos produzidos:

- a) Roteiro de Testes.

Atividades:

- a) Criar os Roteiros de Testes.
- b) Analisar o fluxo de execução do sistema e/ou Casos de Uso da especificação do sistema.
- c) Revisar os Roteiros de Testes.

EXECUÇÃO

Nesta etapa do processo os testes serão efetivamente realizados, de acordo com os planejamentos feitos. Mas lembre-se que os resultados desses testes devem ser devidamente registrados e, se surgir uma nova versão do sistema, por modificação do escopo do desenvolvimento ou por correção de defeitos, os testes devem ser realizados novamente.



**FIQUE
ALERTA**

É importante ressaltar que a execução dos testes ocorrerá em todos os ambientes disponíveis, ou seja, o ambiente de desenvolvimento, de teste e de produção, dependendo do tipo e propósito do teste.

Preparar dados de teste

Objetivo:

- a) Preparar dados que serão usados nos Casos de Testes e roteiros de testes automatizados (scripts).

Artefatos de entrada:

- a) Casos de Testes e roteiros de testes automatizados.
- b) Roteiros de Teste do Projeto de Teste.
- c) Documentação técnica do sistema.
- d) Criar bases de testes.
- e) Especificação dos dados de testes.

Artefatos produzidos:

- a) Base de dados de testes disponíveis.

Atividades:

- a) Identificar a disponibilidade da base de dados de testes.
- b) Estruturar o processo de captura de dados.
- c) Definir com os usuários os dados que serão usados em testes.
- d) Definir a execução dos testes de acordo com os dados utilizados.
- e) Testar as bases de dados de teste.

Executar testes

Objetivos:

- a) Executar os Casos de Testes e roteiros de testes automatizados, caso esteja previsto no Projeto de Teste.

Artefatos de entrada:

- a) Roteiros de Testes.
- b) Casos de Testes.
- c) Roteiros de testes automatizados (scripts).
- d) Resultados esperados dos testes.

Artefatos produzidos:

- a) Resultados de Testes.
- b) Relatórios de Defeitos identificados e status de resolução.
- c) Casos de Testes e roteiros de testes automatizados ajustados.

Atividades:

- a) Executar os Roteiros de Testes e Casos de Testes.
- b) Analisar resultados obtidos e resultados esperados.
- c) Gerar relatórios de defeitos encontrados.
- d) Registrar os defeitos encontrados nos testes.
- e) Definir com os usuários a execução dos testes.
- f) Informar aos usuários os resultados dos testes.
- g) Realizar as correções de defeitos encontrados nos testes.



anyaberhut (2017)

Solucionar ocorrências de Testes

Objetivo:

- a) Identificar a causa raiz dos defeitos encontrados.

Artefatos de entrada:

- a) Relatório de Defeitos com pendências a resolver.
- b) Resultados de Testes.

Artefatos produzidos:

- a) Relatório de Defeitos sem pendências.
- b) Relatório.

Atividades:

- a) Comparar resultados do Relatório de Defeitos com os resultados esperados dos testes.
- b) Identificar partes do sistema que originaram o defeito.
- c) Corrigir o defeito encontrado.
- d) Disponibilizar nova versão de partes do sistema que apresentaram defeito.
- e) Atualizar o Relatório de Defeitos, de acordo com a análise e correção destes defeitos.
- f) Corrigir Casos de Testes ou roteiros de testes automatizados, caso seja necessário.

Acompanhar execução dos Casos de Testes

Objetivo:

- a) Acompanhamento da execução dos Casos de Testes, para verificar o andamento dos testes com o seu respectivo planejamento.

Artefatos de entrada:

- a) Resultados dos Testes.
- b) Relatórios de Defeitos.
- c) Estratégia de Testes.
- d) Plano de Testes.
- e) Casos de Testes.
- f) Roteiros de Testes.

Artefatos produzidos:

- a) Análise da execução dos Casos de Testes.

Atividades:

- a) Obter os Relatórios de Defeitos.
- b) Elaborar relatórios sobre a execução dos Casos de Testes.
- c) Elaborar relatório de Análise de Resultados de Testes.
- d) Informar aos usuários e desenvolvedores os resultados dos testes.
- e) Divulgar os relatórios produzidos.

Elaborar Relatório Final de Teste

Objetivo:

- a) Reportar todas as ocorrências de testes para os gerentes do Projeto de Testes.

Artefatos de entrada:

- a) Análise dos Resultados de Testes.
- b) Estratégia de Testes.
- c) Resultados dos Testes.
- d) Relatório de Defeitos.
- e) Plano de Testes.

Artefatos produzidos:

- a) Relatório final de testes.

Atividades:

- a) Medir o alcance dos objetivos da Estratégia e Planos de Testes.
- b) Elaborar o Relatório Final.
- c) Estimar o tamanho final do Projeto de Testes.
- d) Estimar o tamanho final do sistema testado.
- e) Realizar reunião de encerramento do Projeto de Testes.
- f) Medir o impacto dos defeitos encontrados nos testes e propor melhorias de qualidade.

ENTREGA

Nesta etapa final do processo, ocorrerá a conclusão do Projeto de Testes, com o arquivamento dos documentos gerados e evidências das ocorrências de testes mais relevantes para melhorias do processo.

Avaliação e Arquivamento da Documentação

Objetivo:

- a) Avaliar a documentação de teste e realizar seu arquivamento.

Artefatos de entrada:

- a) Documentação de teste.

Artefatos produzidos:

- a) Relatório de não conformidades.
- b) Relatório final do Projeto de Testes.
- c) Documentos de testes arquivados.

Atividades:

- a) Avaliar os documentos de testes.
- b) Arquivar os documentos de testes.
- c) Criar relatório de não conformidades.
- d) Criar relatório final de testes.
- e) Avaliar os indicadores de testes.

Depois do acompanhamento dos testes, elaboração do relatório final e a entrega, vamos aos tipos de testes.

2.2 TIPOS DE TESTES

Para permitir a cobertura de testes em diferentes momentos do processo, tornam-se necessárias diferentes abordagens, ou seja, diferentes técnicas e tipos de testes.

A partir desses diferentes cenários de testes, foram criadas classificações e tipos específicos de testes, que serão apresentadas neste tópico. Acompanhe!

Há duas classificações gerais sobre os tipos de testes, sendo identificados como teste de “Caixa Branca” e “Caixa Preta”, fazendo alusão ao que está sendo observado no interior ou estrutura do software. Ou seja, a caixa branca permite a visualização nesta estrutura interna, enquanto a caixa preta permite apenas a visualização externa.

A partir dessa ideia de caixas e suas visões, o teste de caixa branca busca validar o código do software, a sua lógica de funcionamento e sua arquitetura e demais elementos técnicos.

O teste de caixa preta tem como objetivo validar a correta execução das funcionalidades do software, assim como o atendimento aos requisitos, sob o ponto de vista do usuário do sistema.

Na figura seguinte são ilustradas as principais diferenças entre os testes de caixa branca e caixa preta, de acordo com as definições expostas acima.

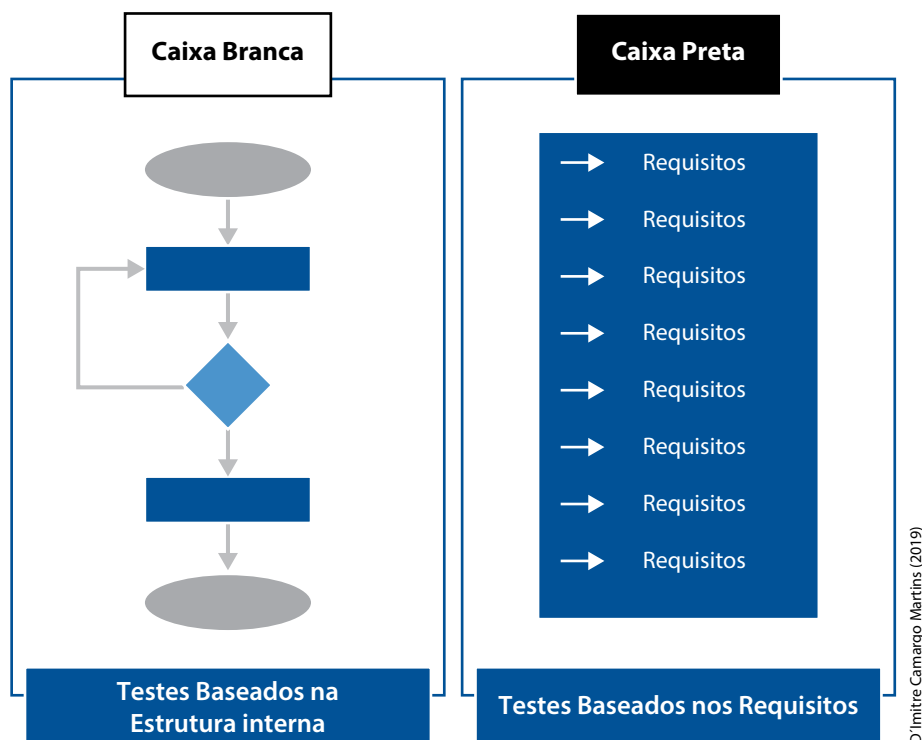


Figura 5 - Diferença entre os testes de caixa branca e caixa preta
Fonte: Do autor (2019)

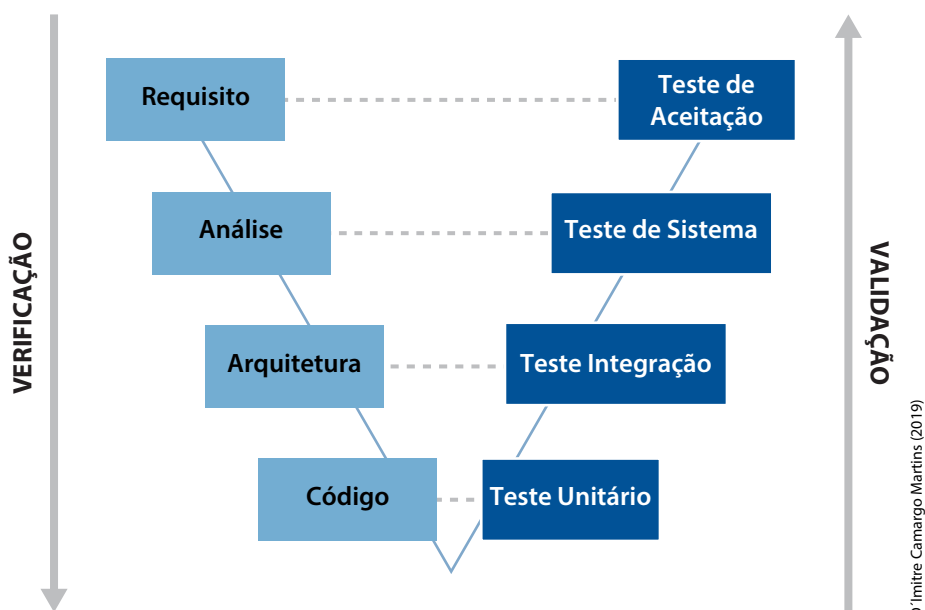


Figura 6 - Modelo V que apresenta testes de caixa branca preta
Fonte: Do autor (2019)

Na próxima figura você pode ver o modelo V de testes, que faz uma correlação entre as atividades de desenvolvimento do sistema e seus respectivos testes de validação, onde os testes unitário e de integração podem ser vistos claramente como testes de caixa branca, pois validam a estrutura do sistema. Já os testes de sistema e de aceitação são testes de caixa preta, pois buscam a validação dos requisitos ou funcionamento do sistema sem preocupar-se com os mecanismos internos deste.

Acompanhe, a seguir, a descrição desses tipos específicos de testes e de outros que fazem parte do processo de testes de sistema.

2.2.1 TIPOS DE TESTES ESPECÍFICOS

TESTE UNITÁRIO

Este teste é essencialmente criado e aplicado por desenvolvedores, onde a cada parte funcional do código pode ser criado um teste unitário para validar o resultado da execução deste trecho funcional do código.

Dessa forma, a garantia da correta execução do código é expressivamente melhorada, reduzindo a necessidade de revisão de código em estágios mais avançados do desenvolvimento e do processo de teste.

TESTE DE INTEGRAÇÃO

O teste de integração ocorre após a realização dos testes unitários, onde cada parte do sistema é testada de forma isolada. Dessa forma, o teste de integração verifica o funcionamento das partes do sistema em conjunto, ou seja, na interação e dependência entre os módulos do sistema e suas interfaces.

Mas há diferentes estratégias para abordar o teste de integração, que variam de acordo com o nível de desenvolvimento dos módulos da integração. Assim, para testes que iniciam no desenvolvimento de componentes de mais baixo nível, a abordagem é denominada Botton-up, enquanto o contrário é identificado como Top-down.

TESTE DE SISTEMA

O teste de sistema compreende diferentes testes que tem como propósito validar o funcionamento do sistema completo, com todos os seus módulos e componentes. Tem-se uma validação dos requisitos funcionais e não funcionais do sistema, identificando se sua execução está de acordo com suas especificações.

Os testes que o compreendem são o teste de carga, performance, usabilidade, segurança e recuperação.

TESTE DE ACEITAÇÃO

Já o teste de aceitação é realizado no ambiente do cliente e com a sua participação direta. Sua finalidade é validar o correto funcionamento do sistema, simulando ações reais dos usuários finais e verificando se sua execução condiz com as suas especificações, se atende às necessidades do negócio e se a usabilidade está apropriada para os futuros usuários do sistema.

TESTE DE REGRESSÃO

Os testes de regressão são necessários para validação do correto funcionamento do sistema após sofrer mudanças ou atualizações, sendo partes do sistema que já foram testados, serão testados novamente, desde que estas partes do sistema possam ter algum impacto em alterações realizadas.

TESTE DE CARGA

O teste de carga também pode ser identificado como teste de estresse. Seu objetivo é validar o limite do funcionamento adequado do sistema, com a aplicação de alto fluxo de dados, grande repetição das funcionalidades do sistema e com a possibilidade de simulação de muitos usuários conectados simultaneamente.

É possível avaliar a estabilidade do sistema, identificar módulos que podem apresentar erros em condições extremas, observar o comportamento do sistema em momentos críticos e coletar dados que possam contribuir para a otimização desse sistema.

TESTE DE CONFIGURAÇÃO

O teste de configuração vai validar o funcionamento do sistema em diferentes ambientes, e pode variar de acordo com a tecnologia usada pelo sistema. Ou seja, o teste pode validar o funcionamento em diferentes sistemas operacionais para sistemas desktops, assim como pode validar o funcionamento em diferentes browsers para sistemas web.

TESTE DE USABILIDADE

Este é um tipo de teste voltado para a equipe de designers do sistema, que por meio de diferentes técnicas procuram identificar, na simulação ou no próprio uso do sistema por seus usuários, uma boa experiência de uso do que pode ser mais agradável para esse usuário.

TESTE DE INSTALAÇÃO

Este teste irá validar os procedimentos de instalação ou de atualização documentados no sistema. Seu objetivo é garantir que tais procedimentos possam ser replicados em diferentes ambientes que irão hospedar o sistema.



TESTE DE SEGURANÇA

O teste de segurança vai validar a segurança de acesso, assim como a segurança da informação, visando garantir a segurança necessária para os usuários e suas informações armazenadas pelo sistema.

TESTE DE RECUPERAÇÃO

A intenção deste teste é validar a resiliência do sistema. Ou seja, como ele pode recuperar-se após uma situação de adversidade, como um bug, uma falha geral ou problemas relacionados à infraestrutura.

TESTE DE DESEMPENHO

Este teste também é identificado como teste de performance e valida os requisitos não funcionais relacionados ao desempenho especificado para o sistema.

TESTE DE QUALIDADE DE CÓDIGO

É um teste de baixo nível que irá analisar o código do sistema e verificar se foram aplicadas boas práticas de desenvolvimento, de acordo com o esperado para o projeto.

TESTE DE INTEROPERABILIDADE

Tem o objetivo de verificar a integração entre outros sistemas internos ou externos, para avaliar a comunicação ou troca de dados entre esses sistemas.

TESTE DE ESTÁTICOS

Neste tipo de teste não há a execução do sistema. Sendo assim, a validação ocorrerá sobre a documentação do sistema, avaliando a estrutura e qualidade dos requisitos documentados e demais registros formais, de acordo com o método de desenvolvimento adotado.

TESTE ALFA

Estes testes acontecem próximos à conclusão do desenvolvimento do sistema e são realizados por usuários que não fazem parte da equipe de desenvolvimento e testes, onde também não há um roteiro de testes definido. Tem como objetivo simular o uso real do sistema e encontrar o máximo de erros possíveis que não foram previstos nos testes documentados.

TESTE BETA

Estes testes são muito parecidos com o teste Alfa, porém ocorrem em uma fase mais próxima da conclusão do desenvolvimento, sendo uma das últimas oportunidades de encontrar defeito antes da entrega do sistema para o cliente.

Agora que você conheceu conceitos de diferentes tipos de testes, use seus conhecimentos para avançar para os próximos tópicos deste capítulo.



**FIQUE
ALERTA**

Apesar da diversidade quanto aos tipos de testes, não significa que um projeto deve compreender todos estes tipos, sendo necessário avaliar a necessidade de realização de determinados testes de acordo com as características do projeto.

2.3 GESTÃO DO PROCESSO DE TESTES

Você sabia que o processo de teste bem definido e estruturado é crucial para que seja possível alcançar a qualidade esperada no desenvolvimento de software? É isso mesmo! A gestão desse processo tem como principal objetivo monitorá-lo e garantir que os resultados produzidos sejam mais assertivos, consumindo menos recursos alocados e em menor tempo.



Para possibilitar o acompanhamento do desempenho do processo, é necessário possuir meios de medir seus resultados, constituindo indicadores que irão revelar, sobre diferentes perspectivas, o quanto o processo é eficaz.

Veja, a seguir, a definição de alguns indicadores.

Indicador de Demanda: relacionado ao número de sistemas testados por determinado tempo.

Indicador de Qualidade: vinculado aos fatores de qualidades de software, que serão apresentados mais adiante.

Indicador de Tempos: mede o tempo médio para a execução dos diferentes tipos de testes realizados.

Indicador de Entrega: mostra a quantidade de componentes entregues no sistema testado.

Indicador de Produtividade: está relacionado a quantidade de teste realizado por testador.

Indicador de Custo: retorna o custo financeiro associado às diferentes etapas do teste.


**SAIBA
MAIS**

Outros indicadores podem ser usados, de acordo com a necessidade de medição do projeto no qual os testes estão sendo realizados e o processo executado.

Lembre-se que ao tratar do indicador de qualidade é importante tornar o conceito de qualidade de software mais concreto. Para isso, foram criados subgrupos de indicadores a partir dessa definição, identificados como fatores de qualidade de software, que buscam refletir características ou atributos de software que atendam os usuários desse software.

No quadro que você verá, relaciona-se os principais fatores de qualidade de software, com seus respectivos conceitos e sugestões de cálculos para formar seus indicadores. Observe!

FATOR DE QUALIDADE DE SOFTWARE	CONCEITO	FÓRMULA DE CÁLCULO DO INDICADOR
Correção	Atender as especificações e objetos dos usuários	Erros / linha de código
Eficiência	Quantidade de recursos computacionais e linhas de código necessários para desenvolver uma função do sistema	Atual utilização / utilização alocada
Flexibilidade	Esforço necessário para alterar um software em operação	Média de dias de trabalho para mudar
Integridade	Controle de acesso aos dados por usuário não autorizado	Falhas – relativas à segurança
Interoperabilidade	Esforço necessário para integração com outros sistemas	Esforço para integrar / esforço para desenvolver
Manutenibilidade	Esforço necessário para localizar e corrigir um defeito, em sistemas em operação	Média de dia de trabalho para localizar e corrigir
Portabilidade	Esforço necessário para transferir um software em operação para outro ambiente computacional (hardware)	Esforço para portar / esforço para desenvolver
Confiabilidade	Execução das funções do software com precisão	Defeitos / linha de código
Reuso	Capacidade de reutilizar o software ou parte dele em outra aplicação	Esforço para converter / esforço para desenvolver
Testabilidade	Esforço necessário para testar um software, com o objetivo que execute as funções de acordo com o esperado	Esforço para testar / esforço para desenvolver

Quadro 2 - Relação de Fatores de qualidade de software
Fonte: Do autor (2019)

Na gestão do processo de teste é natural que cada organização selecione os indicadores mais apropriados para o seu contexto e projeto no qual o processo esteja sendo executado. Após a escolha desses indicadores, mais adequados à gestão do processo, é necessário seguir etapas determinadas, que você verá a seguir.

Caracterização dos indicadores

Nesta etapa, são selecionados os indicadores relevantes para a gestão do processo, aplicado ao projeto específico. Também são definidos os cálculos desses indicadores, métodos de aplicação dos cálculos, valores de referência dos indicadores e a maneira como estes resultados devem ser interpretados.

Caracterização das medições

As medições têm definido os mecanismos de capturar seus resultados, os momentos que essas capturas devem acontecer e os valores de resultados válidos.

Operacionalização

Nesta fase deverá ocorrer a implantação dos procedimentos previamente definidos na caracterização dos indicadores.

Monitoramento e análise

Após a implantação dos indicadores e coleta dos seus respectivos dados, deverá ocorrer a etapa de monitoramento dos resultados e análise da execução do processo de teste.

Melhorias

Esta é a última fase da gestão do processo, que consiste em intervenção no processo, caso os resultados dos indicadores não sejam satisfatórios. É nessa fase que são propostas mudanças para melhorar o processo. Mais quais seriam essas melhorias no processo de testes? Veja no próximo tópico.

2.4 MELHORIAS NO PROCESSO DE TESTES

A ação de melhoria do processo de testes é algo que tende a acontecer de forma contínua, tendo em vista a complexidade inerente ao processo. Pesquisas mostram que os custos associados à manutenção de software tendem a ser maiores com o passar do tempo. Assim, quanto mais tarde os defeitos são encontrados no sistema, mais caro será a manutenção, comprometendo, também, a qualidade do software.

Existem diversos fatores que podem influenciar a qualidade do processo e, consequentemente, nos resultados produzidos por ele. Assim, é importante identificar os principais fatores para prover melhorias no processo.

A análise da causa raiz (RCA) é uma técnica usada para identificar e corrigir falhas em processo, desenvolvida inicialmente pela indústria de energia nuclear e, posteriormente, aplicada à engenharia de software. Tem como fundamentos quatro etapas para encontrar falhas e rastreá-las até sua causa raiz, por meio de quatro perguntas:

- a) O que são falhas?
- b) Quando as falhas acontecem?
- c) Por que as falhas acontecem?
- d) Como as falhas podem ser evitadas?

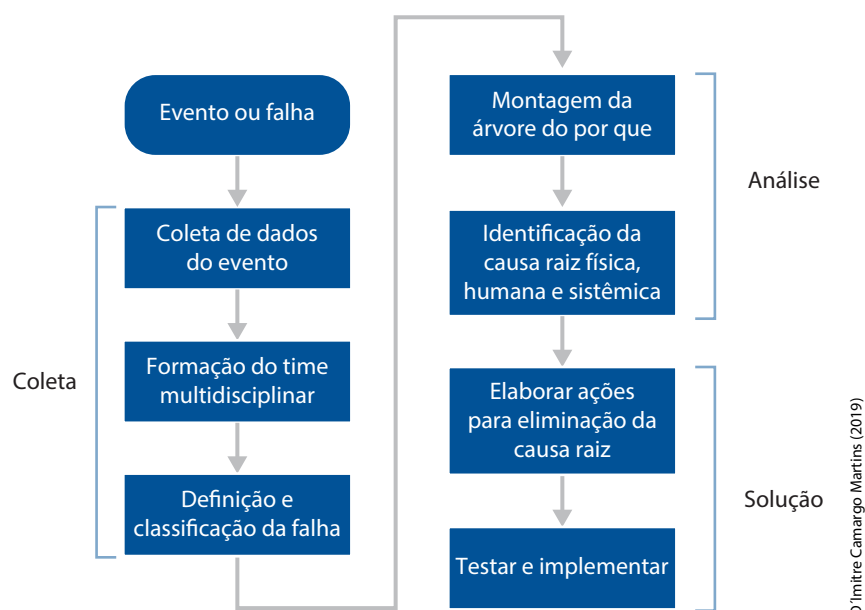


Figura 7 - Fluxograma da dinâmica RCA
Fonte: Do autor (2019)

D'Imitre Camargo Martins (2019)

Na figura anterior, você pôde ver o fluxo de ações aplicadas à técnica RCA. Para tornar mais claro a aplicação dessa técnica, na sequência serão detalhadas as abordagens para cada uma das questões, que buscam categorizar as falhas existentes no processo, entendê-las e criar mecanismos para reduzir ou eliminá-las.

Para iniciar, é importante que você saiba identificar as falhas.

O que são falhas?

No momento inicial de aplicação desta técnica, a intenção é identificar as possíveis falhas existentes no processo, criando categorias para eventuais falhas e classificando por severidade e tipo. Para tornar claro o conceito de severidade, acompanhe a caracterização deste nível de severidade de acordo com o impacto causado no produto.

No quadro a seguir há a apresentação dos níveis de severidade e suas respectivas descrições, que irão auxiliar na categorização das possíveis falhas encontradas no processo.

NÍVEL	DESCRIÇÃO
Crítico	Impede o uso do software
Severo	Alguns recursos do software não podem ser usados e não há solução de contorno
Moderado	Alguns recursos possuem limitação de uso, prejudicando a eficiência, confiabilidade e usabilidade do software
Cosmético	Problema de baixo impacto, mas que causa desconforto para o usuário do software

Quadro 3 - Níveis de severidade para a análise de causa raiz
Fonte: Do autor (2019)

E veja bem! O objetivo dessa técnica não é identificar todas as categorias de falhas existentes no processo, mas focar nas principais, com o intuito de reduzir ou remover as falhas do processo.

Como o objetivo não é categorizar todas as falhas e sim as mais relevantes para o projeto, são criadas essas categorias de falhas com os critérios de severidade, frequência de ocorrência e custo do reparo.

Dessa forma, falhas que possuem maior severidade, que são apresentadas com maior frequência e com alto custo de resolução, ganham o foco da análise, buscando-se a identificação da causa raiz com mais precisão.

Quando as falhas acontecem?

A intenção neste passo é mapear o momento em que as falhas ocorrem ou buscar rastrear quando a falha foi introduzida, a partir da sua manifestação. Portanto, os registros dos testes são de grande valia, pois é possível manter o histórico da ocorrência das falhas e, dependendo do momento desta ocorrência, interpretar quando a falha foi introduzida no software.

Por que as falhas acontecem?

Nesta etapa o objetivo é de fato encontrar o “porquê” ou a causa raiz da falha, ou seja, o real motivo pelo qual tais falhas ocorreram. Para isso, é necessário buscar de forma incessante a causa da falha. Essa etapa também é uma investigação entre falhas que podem ser desencadeadas por outras falhas.

Como as falhas podem ser evitadas?

Na etapa final da técnica RCA é onde realmente ocorre a manutenção do processo e, consequentemente, a sua melhoria, para que as falhas identificadas e categorizadas possam ser evitadas ou previamente detectadas.

De acordo com a regra de Pareto, que prevê a distribuição de até 80% das falhas em apenas 20% do código do software, as falhas tendem a concentrar-se em módulos específicos do sistema. Assim, é possível identificá-las e categorizá-las, evitando a sua incidência, uma vez que se conhece a sua causa raiz.



**SAIBA
MAIS**

Existem alguns métodos para realizar a análise de problema, para buscar a sua causa raiz. Visite o site <<https://www.lean.org.br/conceitos/20/cinco-porques---metodo-de-analise-de-problema-para-buscar-sua-causa-raiz.aspx>> e confira mais detalhes a respeito!

2.5 INFRAESTRUTURA DE TESTE

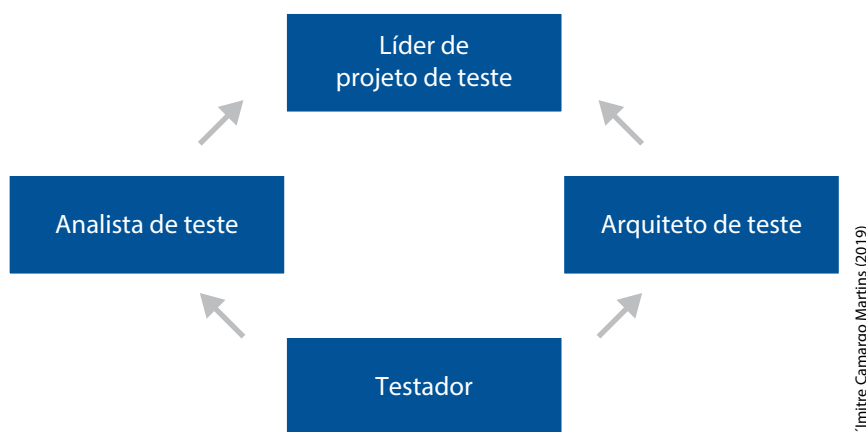
A infraestrutura do teste compreende todos os recursos necessários para que os testes possam ser realizados, englobando as pessoas com suas respectivas capacidades técnicas e seus papéis dentro do processo de teste, além do ambiente de teste, que deverá prover os recursos tecnológicos para apoiar as atividades das pessoas inseridas no contexto.

E você sabe quais são as responsabilidades da equipe de testes? Veja na sequência.

2.5.1 PAPÉIS E RESPONSABILIDADES DE TESTE

A equipe de testes deverá possuir diferentes papéis, com diferentes funções e responsabilidades, tendo como funções básicas:

- a) Líder de Projeto de Teste (LT)
- b) Arquiteto de Teste (AT)
- c) Analista de Teste (AN)
- d) Testador (TE)



D'Imite Camargo Martins (2019)

Figura 8 - Hierarquia de profissionais de teste
Fonte: Do autor (2019)

Essas funções podem ser exercidas por diferentes profissionais, ou ainda, quando a equipe é pequena e multidisciplinar, o mesmo profissional pode acumular mais de uma função ou papel no processo de teste.

Na sequência serão apresentados quadros que associam os papéis a seus respectivos perfis, suas atividades, dentro de uma matriz de responsabilidade. Também é mostrada a distribuição de responsabilidades por tipos de testes. Confira!

Líder de Projeto de Teste (LT)	Profissional responsável pela liderança de um projeto de teste específico e normalmente também está relacionado ao processo de desenvolvimento.
Arquiteto de Teste (AT)	Profissional responsável por estruturar o ambiente de teste, selecionar as ferramentas de testes utilizadas e preparar a equipe de testes para a execução do planejamento.
Analista de Teste (AN)	Profissional responsável por modelar os casos de testes e criar os scripts de testes.
Testador (TE)	Profissional responsável por executar os casos e scripts de testes.

Quadro 4 - Relação de perfis de profissionais de teste
Fonte: Do autor (2019)

ETAPAS	ATIVIDADES	LT	AT	AN	TE
Procedimentos Iniciais	Elaborar guia operacional de testes	x			
Planejamento	Estabelecer estratégia de testes	x	x		
	Estabelecer plano de teste	x	x		
	Revisar estratégia de teste	x	x		
Preparação	Revisar plano de teste	x	x		
	Adequação do projeto de teste para gerência de configuração e controle de mudança	x	x	x	
	Disponibilizar infraestrutura e ferramentas de teste	x	x	x	
Especificação	Disponibilizar profissionais de teste	x	x		
	Elaborar roteiro de teste	x	x	x	
	Elaborar casos de teste	x	x	x	
	Preparar dados de teste				
	Executar teste				x
	Solucionar ocorrências de teste				x
	Acompanhamento da execução dos testes	x		x	x
Entrega	Elaborar relatório de testes	x	x	x	
	Avaliar e arquivar documentos produzidos pelos testes	x	x	x	

Quadro 5 - Relação de atividades por profissional de teste
Fonte: Do autor (2019)

TIPOS DE TESTE	EQUIPES RESPONSÁVEIS PELOS TESTES			
	TESTE	DESENVOLVIMENTO	PRODUÇÃO	USUÁRIOS
Unitário		x		
Integração	x	x		
Sistema	x	x		x
Aceitação	x	x		x
Estresse	x	x	x	
Performance	x	x	x	x
Regressão	x	x		
Funcional	x	x		x

Quadro 6 - Relação de tipos de testes por equipes responsáveis
Fonte: Do autor (2019)

**FIQUE ALERTA**

Como pôde ser visto, dada a importância dos testes e sua abrangência no desenvolvimento de software, todos os atores deste contexto participam dos testes e das validações do sistema.

Ainda assim há uma proporção da quantidade de profissionais de testes para desenvolvedores, que pode variar de 1 para 1 até de 1 para 5, dependendo do tamanho do projeto, velocidade de desenvolvimento e urgência de entrega do software.

2.5.2 AMBIENTE DE TESTE

O ambiente de teste possui grande importância no contexto de validação de software, pois torna possível a correta aplicação do processo de teste, por meio de equipamentos, redes de dados e softwares de testes, definidos no planejamento dos testes.

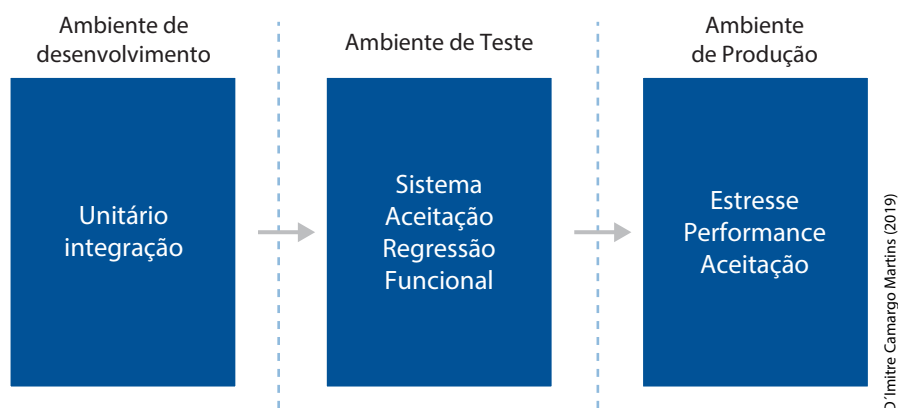


Figura 9 - Ambientes de teste
Fonte: Do autor (2019)

Dimitre Camargo Martins (2019)

A figura anterior apresentou um esquema de promoção de ambientes e os principais tipos de testes que são executados em cada ambiente de teste. A responsabilidade pela preparação desses ambientes é do arquiteto de testes, assim como os softwares de testes que devem estar disponíveis em cada ambiente.

A importância desses ambientes consiste em garantias de qualidade ao sistema, pois a execução em adequado ambiente de testes permite encontrar defeitos que seriam encontrados em ambiente de produção e não detectados em desenvolvimento.

Outra vantagem do ambiente de teste, independentemente do desenvolvimento, é a possibilidade de continuar a construção do sistema sem interferir nos testes ou de haver o risco de os testes impactarem no desenvolvimento do sistema.

Mas para que o ambiente de teste alcance seus objetivos, auxiliando a equipe de testes e garantindo qualidade ao sistema, é importante considerar algumas características. Veja quais são!

- a) Sistema operacional.
- b) Arquitetura do sistema em desenvolvimento.
- c) Identificação de componentes.
- d) Meio de acesso ao sistema.
- e) Linguagem de programação usada para o desenvolvimento do sistema.
- f) Conectividade entre os ambientes.

Sobre o ambiente de teste isolado ou independente dos demais ambientes, é importante observar outras características, relacionadas a seguir:

- a) Ambiente isolado com processamento independente.
- b) Especificação do ambiente similar ao ambiente de produção.
- c) Acesso ao ambiente restrito à equipe de testes.
- d) Uso de massa de dados com representatividade quantitativa e qualitativa.

Diante de ambientes de testes confiáveis, a execução dos testes torna-se mais eficiente e sua efetividade em encontrar possíveis defeitos aumenta consideravelmente. Sendo assim, esta estrutura do ambiente de trabalho possui grande peso na realização das atividades de testes.

Nesse processo, é fundamental estar atento à análise de risco. Vamos em frente para saber mais!

2.6 ANÁLISE DE RISCO

O risco é a possibilidade de algo acontecer, seja algo indesejado ou algo que possa representar uma oportunidade. E para a possibilidade destes eventos, no contexto do processo de teste, há uma gestão específica de riscos, para garantir não venham a se concretizar e transformar em perdas para a organização.

E veja bem! Normalmente, o risco é algo indesejado, mas a sua presença não pode ser negada ou negligenciada. Na verdade, ela deve ser analisada para que possa ser identificada a probabilidade de tal risco tornar-se real, pois quando existe uma chance elevada de ocorrência concreta de um risco, este torna-se uma ameaça que deve ser tratada.

Mas como tratar a ameaça? Para isso, deve ser criado um mecanismo que impeça uma ameaça de tornar-se um evento. No contexto de análise de riscos, estes mecanismos são identificados como mecanismos de controle, ou somente como controle.

O controle deve eliminar as ameaças que poderiam tornar um risco em ocorrência. Porém, quando o controle não atinge seus objetivos, são identificadas as vulnerabilidades, que poderão aumentar a probabilidade de um risco tornar-se real.

Após esses conceitos iniciais sobre risco, é possível afirmar que a análise de riscos consiste em avaliar os riscos, as ameaças associadas, os controles para mitigar estas ameaças e as vulnerabilidades causadas pelas falhas de controle.

Nesse momento, é natural questionar a aplicação da análise de riscos no processo de testes de software e, a partir desse questionamento, será relacionada a análise aos testes.

Mas saiba que a atividade de teste de software é algo oneroso para as organizações, e a cobertura total de testes sobre um sistema costuma ser inviável em termos de orçamento e prazo do projeto. Desta forma, a análise de risco irá auxiliar a identificar as funcionalidades do sistema que representam maior risco ou perda para o negócio, para que os testes possam ser mais concentrados e exaustivos.

Portanto, ao fazer a análise de riscos, é importante considerar dois critérios:

- a) Probabilidade de ocorrência do risco.
- b) Impacto e perda ao negócio, causada pelo risco.

Os riscos podem assumir diferentes faces e podem estar presentes em diferentes momentos do projeto de testes. Por exemplo, pode ser considerado um risco a adoção de uma nova tecnologia pela equipe de desenvolvimento ou ainda uma importante funcionalidade do sistema que cause impacto ao negócio, caso um defeito seja encontrado.

A quantidade de testes a serem realizados em um sistema pode ser medida pela quantidade e relevância de riscos identificados no projeto. Por isso, é muito importante que a análise de riscos seja feita de forma correta, para evitar que os testes sejam insuficientes ou muito extensos.

No quadro seguinte, você verá uma classificação dos riscos, com a comparação do impacto ao negócio e a probabilidade de ocorrência, que tornará a decisão sobre o que testar mais objetiva. Como destacado no quadro a seguir, esta classificação é identificada como:

- a) **AA** = Alto impacto e Alta probabilidade de ocorrência.
- b) **AM** = Alto impacto e Média probabilidade de ocorrência.
- c) **MA** = Médio impacto e Alta probabilidade de ocorrência.
- d) **MM** = Médio impacto e Média probabilidade de ocorrência.

Deste modo é mais fácil identificar as prioridades de realização de testes, permitindo melhorar o dimensionamento da sua cobertura e a alocação adequada da equipe de teste.

IMPACTO QUE O RISCO CAUSA AO NEGÓCIO	PROBABILIDADE DE OCORRÊNCIA DO RISCO		
	ALTA	MÉDIA	BAIXA
Alto	AA	AM	AB
Médio	MA	MM	MB
Baixo	BA	BM	BB

Quadro 7 - Fluxo do gerenciamento de risco
Fonte: Do autor (2019)

Como destacado no quadro, os riscos classificados como AA, AM, MA e MM devem ter prioridade sobre a execução de testes, permitindo melhorar o dimensionamento da cobertura de testes e a alocação adequada da equipe de teste.

Sobre a análise de riscos, é importante ressaltar que essa atividade deve ser revisada durante todo o projeto de teste, pois as probabilidades de ocorrência de risco e o impacto ao negócio podem mudar no decorrer do projeto.

Mas não para por aqui. Ainda tem muita informação importante sobre o teste de sistemas. Por isso, continue seus estudos, testando sua curiosidade por novos horizontes do saber.



CASOS E RELATOS

Testando o processo

Em uma tradicional empresa de desenvolvimento de sistemas, a equipe de testes era muito dedicada às suas atividades, contudo falhas no sistema eram frequentemente identificadas pelos usuários finais, quando o software já estava em produção. Para buscar a solução deste problema, a empresa contratou Carlos, para atuar como coordenador desta equipe de testes, com o objetivo de elevar a qualidade de seus sistemas. Assim que Carlos iniciou seu trabalho ele percebeu que os testes eram realizados sem procedimentos estruturados, com a aplicação de poucas técnicas de testes e sem a correta avaliação de risco. Diante deste cenário o novo coordenador implantou um processo de teste que visava resolver todos os problemas de gestão e operação, com isso as novas atualizações dos sistemas desta empresa ganharam qualidade e confiabilidade dos seus usuários.



RECAPITULANDO

Neste primeiro capítulo, você teve uma visão geral do teste de software, identificando os passos de todo o processo e verificando a importância da eficiência na realização dos testes. Você viu os tipos e o ambiente de testes e também analisou os riscos, cuja gestão é fundamental para que não ocorram perdas para a organização. Nesse sentido, você pôde ver que a presença de riscos não pode ser negligenciada, mas sim deve ser levada para análise e solução.

Planejamento e Execução dos Testes

3



Neste capítulo serão abordados os conceitos e métodos para realizar o planejamento de teste, por meio do desenvolvimento do Plano de Testes, as tarefas envolvidas para criar tal planejamento, assim como serão definidos os procedimentos para identificar a Estratégia de Teste. Com base neste planejamento, você vai ver a forma de produzir os testes, seus documentos e o plano de caso de testes.

A partir de seus estudos neste capítulo você terá subsídios para:

- a) definir um plano de testes;
- b) empregar documentos de testes e planos de caso de teste;
- c) identificar problemas de sistemas por meio de aplicação de teste;
- d) reconhecer a execução dos diferentes tipos de teste.

Inicie seus estudos neste capítulo conhecendo a importância do planejamento antes da aplicação de testes.

3.1 DESENVOLVIMENTO DO PLANO DE TESTES

Para começar, é importante tornar clara a necessidade e o motivo pelo qual os testes devem ser planejados, e a partir deste planejamento defini-los e posteriormente executá-los. Como mencionado neste material didático, os custos de testes aumentam muito em decorrência da evolução do desenvolvimento do sistema. Dessa forma, é mais barato e interessante para as organizações que os defeitos dos sistemas sejam encontrados o mais cedo possível.

Diante disso, o planejamento dos testes proporciona maiores garantias, para que o teste seja estruturado, organizado e que siga metodologias eficientes.

Para obter os resultados desejados na execução dos testes, é criado o documento de planejamento, identificado como Plano de Testes, que é a maneira de documentar grande parte do Projeto de Teste e permitir que os testes possam ser controlados e repetidos em testes de regressão para futuras manutenções ou de funcionalidades críticas, além de limitar a cobertura de testes sobre o sistema.

O Plano de Teste possui três pilares ou propósitos principais, que devem ser evidenciados, sendo eles:

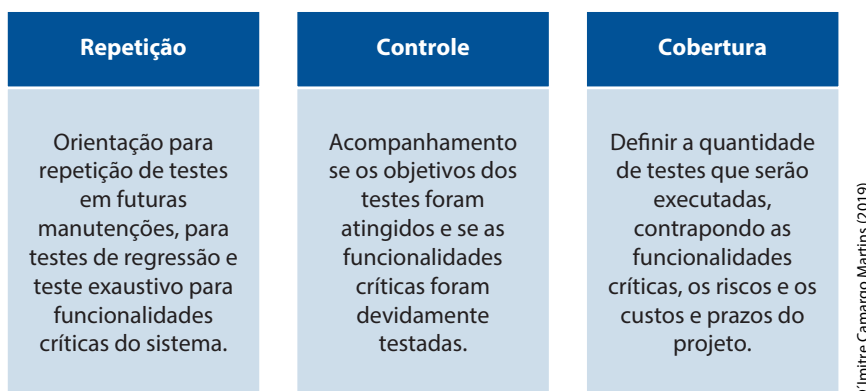


Figura 10 - Principais pilares do plano de teste
Fonte: Do autor (2019)

O Plano de Teste é um dos principais documentos de teste e o instrumento necessário para alcançar os requisitos de teste. Este plano é criado a partir dos requisitos do sistema, do negócio e de teste. Diante da sua relevância, no próximo tópico você vai ver a maneira de desenvolver um Plano de Testes. Vamos lá!



FIQUE ALERTA

O Plano de Teste irá descrever como os testes serão conduzidos, traçando o caminho pelo qual a execução dos testes deve percorrer. Dependendo da abordagem no desenvolvimento desse plano, pode até ser considerado como o principal plano do projeto de teste.

É fundamental saber que, devido à importância do Plano de Teste, é normal ser dedicado boa parte do tempo do projeto de teste na elaboração deste documento. Assim, quando esta etapa é realizada com eficiência, a tendência é a diminuição de problemas ou eventos não previstos na execução do projeto.

VANTAGENS	DESVANTAGENS
Baixo custo inicial	Reduzir a qualidade final do sistema
Rodízio de desenvolvedores	Maior custo de futura manutenção
	Tendência à informalidade de testes
	Risco de não realizar todos os testes
	Falta de conhecimento do negócio

Após a conclusão do Plano de Teste, é crucial a sua atualização, de acordo com possíveis alterações do escopo do projeto de teste, com a atualização da análise de riscos e com o esforço consumido pelo projeto.

Para desenvolver o Plano de Testes, pode-se usar como referência diferentes normas e boas práticas, que ditam como este planejamento pode ser estruturado, de acordo com a perspectiva de métodos de trabalho de cada norma ou boas práticas.

Aqui serão consideradas três diferentes abordagens, a partir da norma IEEE 829:2008 do instituto IEEE (Institute of Electrical and Electronic Engineers), do manual CSTE (Certified Software Testes Exam) do instituto QAI (Quality Assurance Institute), e do guia PMBOK (Project Management Body of Knowledge), do instituto PMI (Project Management Institute). Veja como são listados os principais elementos do Plano de Teste para cada modelo proposto por esses institutos.

IEEE 829:2008

Introdução:

- a) Identificador do Plano de Teste
- b) Escopo
- c) Referências
- d) Nível de sequência de teste
- e) Classe de teste e visão das condições de teste

Detalhes para este nível do Plano de Teste:

- a) Itens de teste
- b) Matriz de rastreabilidade de testes
- c) Funcionalidades que serão testadas
- d) Funcionalidades que não deverão ser testadas
- e) Abordagem do teste
- f) Critérios de liberação e falhas dos testes
- g) Requisitos de suspensão e retomadas de testes
- h) Entregas do teste

Gerência de teste:

- a) Tarefas do teste
- b) Ambientes de teste
- c) Responsabilidades
- d) Integração entre módulos do sistema
- e) Recursos e alocação
- f) Treinamentos
- g) Cronograma, custos e estimativas
- h) Riscos e planos de resposta

Geral:

- a) Procedimentos de garantia de qualidade
- b) Métricas
- c) Cobertura dos testes
- d) Glossário do documento
- e) Procedimento de versionamento do documento

CSTE

- a) Escopo de Teste
- b) Objetivos de Teste
- c) Premissas
- d) Análise de risco
- e) Estrutura de teste
- f) Funções e Responsabilidade
- g) Recursos e cronograma de teste
- h) Gerenciamento de dados de teste
- i) Ambiente de teste
- j) Ferramentas de teste

PMBOK

- a) Escopo
- b) Custo
- c) Tempo

- d) Qualidade
- e) Integração
- f) Recursos Humanos
- g) Comunicação
- h) Riscos
- i) Suprimentos

Nos modelos apresentados, note que apesar de possuírem diferenças visíveis em seus elementos e a abordagem de forma geral, há uma relação entre eles, a partir dos itens de cada elemento destes modelos. Veja mais no quadro seguinte.

PLANO DE TESTE PMBOK	PLANO DE TESTE IEEE 829:2008	PLANO DE TESTE CSTE
Escopo	Identificação do Plano de Teste Referências Introdução Funcionalidades testadas Funções testadas pelo usuário Funções não testadas pelo usuário	Escopo de teste Objetivos do teste Premissas Análise de risco Estrutura do teste Ferramentas de teste
Custo	Métricas	Métricas
Tempo	Cronograma	Estrutura do teste Recursos e cronograma de testes
Qualidade	Estratégia dos testes Critérios de conclusão Critérios de interrupção Critérios de retomada	Gerenciamento de dados de teste
Integração	Ambiente de teste	Ambiente de teste
Recursos humanos	Equipe Treinamentos Alocação Responsabilidades	Funções e responsabilidades
Comunicação	Entrega do Plano de teste Entrega da Estratégia de teste Entrega dos Casos de Teste	Comunicação
Riscos	Riscos do processo de teste Plano de riscos e respostas	Análise de riscos
Suprimentos	Não há relação	Ferramentas de teste

Quadro 8 - Relação de itens dos elementos dos modelos de plano de teste
Fonte: Do autor (2019)

Com base nas informações sobre a importância e possíveis modelos de referência do Plano de Teste, no próximo tópico deste capítulo serão apresentadas as tarefas e procedimentos para desenvolver um Plano de Teste. Acompanhe atento!

3.1.1 TAREFAS PARA DESENVOLVER O PLANEJAMENTO

Ao iniciar o desenvolvimento do Plano de Teste é necessário possuir determinados documentos, que serão usados como insumos para estruturar e definir o conteúdo desse planejamento.

E você sabe qual é o principal documento? É o Plano de Desenvolvimento do projeto de desenvolvimento do software a ser testado. Nesse plano estão definidas as atividades de teste, permitindo que os planos de desenvolvimento de teste estejam bem alinhados.

Outros documentos também são necessários, como os requisitos de negócio, requisitos do sistema, que darão origem aos requisitos de teste, assim como demais documentos do projeto de desenvolvimento, tendo em vista que o teste, apesar de ser independente do desenvolvimento, possui seu projeto diretamente atrelado ao projeto de desenvolvimento.

Para criar o planejamento dos testes é possível seguir um roteiro com tarefas específicas, listadas a seguir:

- a) Estruturar a equipe de testes.
- b) Identificar os riscos do projeto.
- c) Criar o documento de Plano de Teste.

Cada tarefa mencionada possui subtarefas e atividades associadas. A partir da compreensão de cada uma delas, será possível identificar o trabalho necessário para criar o planejamento e seu principal documento, o Plano de Teste. Vamos iniciar pela estruturação da equipe de testes.

Estruturar a equipe de testes

Esta é a primeira tarefa, pois definirá como os testes serão conduzidos e, conseqüentemente, planejados, tendo em vista que nesse momento também será eleito o líder do projeto de teste, o qual terá como uma de suas responsabilidades criar o planejamento dos testes.

As equipes de testes podem ser estruturadas de diferentes formas, cada uma destas formas possui vantagens e desvantagens.

A primeira maneira de estruturar a equipe de teste é atribuir as atividades de testes para a equipe de desenvolvimento, ou seja, a equipe que já está desenvolvendo o sistema também terá atribuições relacionadas aos testes deste software.

Como principal vantagem dessa estrutura está o custo imediato associado ao projeto, pois não será necessário solicitar novos recursos que consumirão o orçamento deste projeto.

Nesses casos, normalmente é observado que as atividades relacionadas ao desenvolvimento possuem maior prioridade e, para que o prazo do projeto possa ser cumprido, as atividades de testes são negligenciadas e substituídas por atividades de desenvolvimento que possam concluir a entrega do sistema.

Ainda nessa abordagem, há problemas quanto à falta de conhecimento das regras de negócio do sistema, uma vez que os desenvolvedores devem testar módulos do sistema, desenvolvidos por outros membros da equipe, para evitar a validação do próprio trabalho.

A seguir, veja um resumo das vantagens e desvantagens desta estrutura de equipe.

VANTAGENS	DESVANTAGENS
Baixo custo inicial	Reduzir a qualidade final do sistema
Rodízio de desenvolvedores	Maior custo de futura manutenção
	Tendência à informalidade de testes
	Risco de não realizar todos os testes
	Falta de conhecimento do negócio

Quadro 9 - Vantagens e desvantagens da equipe de desenvolvimento na função de testadores
Fonte: Do autor (2019)

Outra possibilidade de estrutura de equipe de teste é formar uma equipe independente e focada somente nas atividades de testes. Saiba que essa é uma tendência entre as organizações, em função das diversas vantagens observadas a médio e longo prazo.

Ao abordar a questão de custos do projeto, esta estrutura de equipe certamente apresentará impactos no orçamento do projeto, pois despenderá diversos recursos.

Contudo, os resultados obtidos por equipes com essa estrutura justificam o investimento, pois os custos de manutenção de um software são muito maiores em fases avançadas do projeto. Assim, como esse tipo de equipe é capaz de identificar defeitos no sistema com maior eficiência, irá reduzir despesas futuras para a organização.

Para este perfil de equipe também há vantagens e desvantagens, as quais serão relacionadas a seguir.

VANTAGENS	DESVANTAGENS
Defeitos identificados mais cedo	Alto custo inicial
Redução de futuras manutenções	Maior tempo de liberação
Menor custo de manutenção a médio e longo prazo	Tendência da equipe de DEV não realizar testes unitários
Profissionais especializados	Problemas de relacionamento entre a equipe de teste e desenvolvimento

Quadro 10 - Vantagens e desvantagens da equipe independente de testes
Fonte: Do autor (2019)

A maneira como as equipes serão estruturadas, buscando maximizar as vantagens e eliminar ou reduzir as desvantagens, vai depender diretamente da organização e do projeto de desenvolvimento. Contudo, há riscos que, quando concretizados, são grandes responsáveis pelo fracasso do projeto de teste e, independentemente da estrutura de equipe a ser adotada, estes riscos devem ser tratados com alta prioridade.

Os riscos que podem ser a causa do fracasso do projeto de teste são listados na sequência:

- a) Parte dos requisitos não são testados.
- b) Informalidade na execução e registro dos testes.
- c) Falta de apoio gerencial ao processo de teste.
- d) Defeitos encontrados tardiamente, com o sistema já em produção.
- e) Ausência de automação de testes.
- f) Ausência de metodologia para executar os testes.

Conclui-se que a estrutura da equipe terá que suprir as necessidades da organização, do projeto de testes e ser capaz de mitigar os riscos mencionados, para que o sistema possa ser entregue com a qualidade esperada.

Identificar os riscos do projeto

Nesta tarefa serão avaliados os riscos do projeto de desenvolvimento, que serão inerentes ao projeto de teste, assim como também deverão ser analisados os riscos associados diretamente ao projeto de teste.

Mas como iniciar essa tarefa? A tarefa deve ser iniciada com o estudo da análise de risco e plano de contingência do projeto de desenvolvimento, para que estes riscos já identificados e seus respectivos planos de respostas possam estar presentes no projeto de teste.

Após a identificação dos riscos do projeto do sistema, será necessário levantar os riscos específicos dos testes e, para guiar essa ação, é possível seguir subtarefas que irão compor os riscos dos testes.

A seguir serão relacionadas estas subtarefas e os seus respectivos conceitos, que possibilitarão o entendimento dos riscos do projeto de teste.

Entender os principais processos do negócio

Possibilitará entender o impacto de defeitos no sistema

Identificar a severidade de possíveis falhas

Define a importância e necessidade do teste nos principais processos do negócio que o sistema atua

Identificar os componentes do sistema

Analisar a arquitetura do sistema, identificar os componentes ou interfaces que estarão interagindo com o sistema e ambientes de software e hardware que suportarão o sistema

Identificar os recursos necessários para o projeto de teste

Estimar e garantir os recursos necessários para a completa execução dos testes, previstos no projeto de teste

Fases de teste

Definir os testes por fase, ou seja, quais testes devem ser executados em determinadas fases em que o teste será aplicado

Definir requisitos do ambiente de teste

Garantir que os requisitos do ambiente de teste sejam similares aos ambientes de desenvolvimento e de produção

Identificar ferramentas de testes para a sua execução

Escolher e providenciar a aquisição de ferramentas usadas pela equipe de teste no projeto

Identificar possíveis vulnerabilidades na organização

Verificar possíveis limitações da organização ou de setores externos à TI, que impactarão no fornecimento de recursos ou na atuação da equipe de teste

D'Imitre Camargo Martins (2019)

Guiado pelas subtarefas de identificação de riscos, é possível entender o que poderá impactar e comprometer o projeto de teste, e com estas informações o Plano de Teste pode ser criado com maior robustez e alcançar o sucesso no planejamento. Vamos ver como criar o documento de plano de teste? prossiga para saber mais.

Criar o documento de Plano de Teste

Após as tarefas anteriores, que irão embasar o documento de Plano de Teste, será necessário desenvolver este documento, para tornar claro para todos os envolvidos no projeto o planejamento a ser seguido na realização dos testes, assim como servir de literatura de referência para o ciclo de vida do sistema.

Para desenvolver esta tarefa, que possui certa complexidade, é interessante dividi-la em partes menores. Para isso, o Plano de Teste poderá ser desenvolvido a partir das subtarefas listadas a seguir:

- a) Estabelecer os objetivos do teste.
- b) Desenvolver os roteiros de teste.
- c) Definir a administração do teste.
- d) Escrever o Plano de Teste.

Cada uma dessas subtarefas será descrita com os seus respectivos objetivos e a forma como devem ser realizadas, para compor a tarefa principal de produzir o Plano de Teste.

Estabelecer os objetivos do teste

Nesse momento deverão ser declarados os objetivos do projeto de testes, sendo que estes devem derivar dos objetivos do projeto de desenvolvimento ou, caso o projeto de desenvolvimento não possua tais objetivos definidos, os objetivos de teste devem ser validados com a gestão do projeto de desenvolvimento.

**FIQUE ALERTA**

Lembre-se que o recomendado é possuir poucos objetivos, para que a condução do projeto de teste, guiada por estes objetivos, seja clara e bem definida.

Após a definição dos objetivos, é necessário classificá-los por prioridade, buscando-se ter o mesmo número de objetivos em cada nível de prioridade. Outra ação a ser tomada, é criar os critérios para determinar se os objetivos foram alcançados.



anyaberkut(2018)

Ao longo da realização dos testes, esses objetivos serão relacionados com os testes executados que, ao serem concluídos, irão determinar se os objetivos foram cumpridos.

Desenvolver os roteiros de teste

O roteiro de teste é um conjunto de casos de testes com determinado relacionamento entre eles, ou seja, é um roteiro que irá determinar como estes casos de testes deverão ser executados e o impacto ou relevância da relação destes testes.

Este roteiro é definido a partir dos casos de uso, do levantamento de requisitos do sistema, ou até mesmo por um conjunto de funcionalidades que devem ser testadas.

Chegou a hora de definir a administração do teste. Confira!

Definir a administração do teste

A administração contida no Plano de Teste irá abordar o cronograma, os pontos de controle do projeto de teste e os recursos necessários para a execução dos testes.

Neste elemento do Plano de Teste é evidenciado que este documento terá a função de um contrato, onde as suas cláusulas devem ser respeitadas por todos os envolvidos no projeto de teste.

Escrever o Plano de Teste

Para escrever o Plano de Teste, podem ser usados diferentes modelos de referência, como já citado neste capítulo, onde cada equipe e organização irá selecionar o modelo que melhor se adequar às suas necessidades.

Nesta abordagem, será usado como referência o modelo provido pelo instituto QAI (Quality Assurance International), composto por quatro partes, listadas a seguir:

- a) Informações Gerais.
- b) Plano.
- c) Especificação e avaliação.
- d) Descrição dos testes.

Em cada uma das partes há um conjunto de informações, que serão relacionadas em seus respectivos agrupamentos.

Informações Gerais

- a) Índice das funcionalidades do sistema e os testes que serão realizados.
- b) Informações sobre o ambiente de teste e os critérios de aceite.
- c) Taxa de defeitos esperados para o sistema.
- d) Objetivos do teste.
- e) Referências para os documentos do projeto.

Plano

- a) Descrição do sistema que será testado.
- b) Membros da equipe de teste.
- c) Controles do projeto de teste.
- d) Orçamento do projeto de teste.
- e) Etapas do teste.
- f) Cronograma e orçamento.
- g) Equipamentos usados nos testes.
- h) Softwares usados nos testes.
- i) Profissionais alocados para os testes.
- j) Treinamento.
- k) Lista de testes que devem ser executados em cada etapa.

Especificação e avaliação

Especificações:

- a) Funcionalidades de negócio (a partir de requisitos funcionais).
- b) Funcionalidades estruturais (a partir de requisitos não funcionais).
- c) Relacionamento entre testes e funcionalidades.
- d) Progressão dos testes.

Métodos e restrições:

- a) Metodologia do teste usada.
- b) Ferramentas de teste usadas.
- c) Método de registros dos testes.

Avaliação:

- a) Critérios de avaliação dos resultados dos testes.
- b) Descrição de técnicas para criar dados de teste.

Descrição dos testes

Identificação dos testes que serão executados:

- a) Identificar o método de controle dos testes.
- b) Descrever os dados de entrada para os testes.
- c) Descrever os resultados esperados dos testes.
- d) Descrever os procedimentos de execução dos testes.

Ao término desta subtarefa, o desenvolvimento do planejamento poderá ser concluído, possibilitando definir a estratégia de teste e a elaboração efetiva dos testes que serão executados em um projeto de testes.

Acompanhe as definições, conceitos e estruturas destes artefatos de testes nos próximos tópicos!

3.1.2 ESTRATÉGIA DE TESTE

A estratégia de teste determina o caminho que os testes seguirão, de que forma serão aplicados e em que momento serão realizados. Essas decisões são baseadas fundamentalmente nos riscos do projeto de teste que precisam ser tratados.

Há dois elementos principais na estratégia de teste, identificados por Fatores de teste, que são o tratamento dos riscos de testes identificados, e as Fases de teste, que são as fases do processo de desenvolvimento em que os testes serão realizados.

A elaboração da estratégia de teste pode ser baseada em diferentes aspectos, estando relacionada diretamente aos riscos identificados na análise de riscos do projeto de testes ou em fatores de qualidade do software, que podem ser impactados pelos riscos identificados, ou ainda nos tipos, técnicas e fases do teste. Diante dessas possibilidades, para desenvolver a estratégia de testes, elas serão abordadas separadamente, apontando seus principais conceitos.

**CURIOSIDADES**

Você sabia que o modelo V da engenharia de software irá prever as fases de testes de acordo com a fase de desenvolvimento do sistema?

Estratégia elaborada por fatores de qualidade

Neste método, para definir a estratégia, os riscos identificados serão associados a determinados fatores de qualidade de software, previamente estabelecidos no mercado ou que sejam relevantes para a organização que terá seu negócio como contexto do software testado.

As principais características de qualidade de software são listadas a seguir:

Conectividade Integração entre os componentes do sistema
Continuidade Execução do software de forma ininterrupta
Segurança Dados só podem ser alterados por usuários autorizados
Eficiência Desempenho do sistema nos recursos disponíveis
Funcionalidade Funcionalidades de acordo com a especificação de requisitos
Usabilidade Facilidade de uso do sistema
Performance Processamento da informação de acordo com a especificação não funcional

D'Imitre Camargo Martins (2019)

Figura 11 - Principais características de qualidade de software
Fonte: Do autor (2019)

Após identificar as características que irão impactar no negócio, ou usar as principais características mencionadas anteriormente, será preciso relacioná-las com os riscos identificados e, para cada relação destas informações cruzadas, aplicar uma classificação de impacto do risco sobre o fator de qualidade, em uma escala de 0 a 3, para que desta forma possa ser identificado os riscos ou fatores de qualidade com maior relevância que, por consequência, deverão receber maior atenção dos testes na estratégia traçada.

Acompanhe a comparação das informações de fatores de qualidade e riscos, para compreender melhor essa forma de elaborar a estratégia de teste.

FATORES DE QUALIDADE	RISCO 1	RISCO 2...N	RELEVÂNCIA
Conectividade	2	3	5
Continuidade		2	2
Segurança	1		1
Eficiência	2	1	3
Funcionalidade		2	2
Usabilidade	2		2
Performance		3	3

Tabela 1 - Exemplo da relação de fatores de qualidade e riscos
Fonte: Do autor (2019)

Note que na tabela foram apresentadas apenas duas colunas para os riscos. Porém, na aplicação real desse método, cada risco associado aos fatores de qualidade seria representado por uma nova coluna na tabela, e como demonstrado, a soma dos valores da classificação irá demonstrar os fatores de qualidade que possuem maior relevância na realização dos testes.

Estratégia elaborada por tipos, técnicas e fases do teste

Nesta abordagem, para definir a estratégia de teste, além dos riscos identificados, serão considerados outros aspectos, como os tipos de testes que deverão ser executados, as técnicas de teste, úteis para avaliar a estrutura ou funcionalidades do software, e os estágios ou fases do teste, de acordo com as fases de desenvolvimento do sistema.

Na figura que você vai ver a seguir, são apresentadas as dimensões do teste, que ilustram a abordagem aqui tratada para criar a estratégia de teste, traçando o direcionamento do teste em de que forma testar, como testar e quando testar.

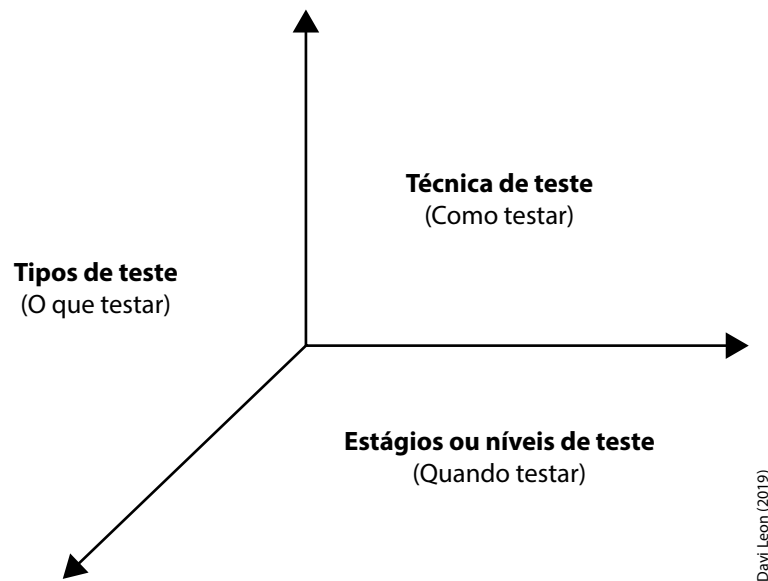


Figura 12 - Dimensões do teste
Fonte: Do autor (2019)

A seguir, veja uma descrição dessas dimensões do teste e de que forma atuam no contexto do desenvolvimento da estratégia de teste.

Técnicas de teste

As técnicas, no contexto de teste de software, resumem-se em dois tipos, estrutural e funcional. Você sabe quais são as diferenças entre elas? Fundamentalmente cada técnica tem propósitos de diferentes validações, ou seja, enquanto a técnica estrutural busca a análise da estrutura ou codificação do sistema, assim como a integração entre os diferentes módulos do sistema, a técnica funcional irá analisar o funcionamento do sistema, isso quer dizer, irá avaliar se as funcionalidades do sistema atendem os seus requisitos, assim como se o resultado do processamento dos dados, pelo sistema, irá apresentar o comportamento esperado.

Para cada tipo de técnica há um conjunto de testes específicos, e para a técnica de teste estrutural, a relação de tipos de testes é apresentada a seguir.

- a) Teste de estresse.
- b) Teste de execução.
- c) Teste de contingência ou recuperação.
- d) Teste de operação.
- e) Teste de conformidade.
- f) Teste de segurança.

A técnica de teste funcional também possui uma gama de tipos de testes, com o propósito de validar essencialmente se os requisitos foram implementados da forma correta. Veja quais são os seus testes.

- a) Teste de requisitos.
- b) Teste de regressão.
- c) Teste de usabilidade.
- d) Teste de integração.
- e) Teste de controle.
- f) Teste paralelo.

Após o entendimento da diferença entre as técnicas de teste e seus respectivos tipos, junto com as particularidades do projeto, a equipe de teste deve ser capaz de selecionar os testes que deverão compor a estratégia de teste, sobre a ótica destas técnicas.

Fases de teste

Esta dimensão é usada como referência na estratégia de teste para identificar o momento e os tipos que serão realizados, e esses momentos são baseados nas fases de desenvolvimento.

Conforme já abordado no capítulo 2, no qual você estudou os tipos de testes de caixa branca e caixa preta, o modelo V representa muito bem esta relação entre os momentos e tipos de testes com as suas respectivas fases de desenvolvimento. Acompanhe a próxima figura.

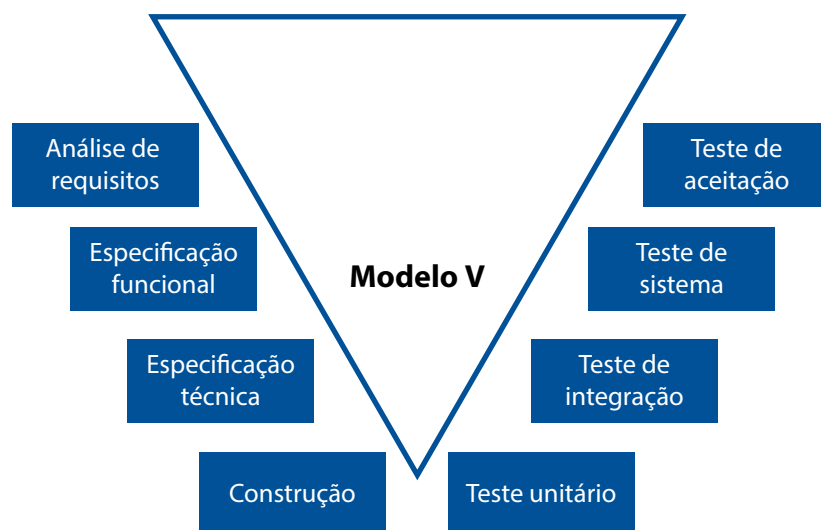


Figura 13 - Modelo V de desenvolvimento x teste
Fonte: Do autor (2019)

Davi Leon (2019)

Qual o primeiro tipo de teste a ser executado? É o teste unitário, que irá validar a codificação das funções implementadas. Logo após o desenvolvimento de um conjunto de funções, será testada a integração dessas funções, de acordo com a especificação técnica ou estrutural do sistema. Assim que o desenvolvimento

do sistema for concluído, serão realizados testes das suas funcionalidades, a fim de validar a especificação funcional. Por último, será realizado o teste de aceitação, normalmente executado pelo cliente, para validar se o comportamento do sistema está de acordo com os seus requisitos, e permitir a sua implantação em ambiente de produção.

Essas fases do teste também são úteis para alimentar a estratégia destes, em conjunto com as suas respectivas técnicas e os tipos de testes pertinentes, para estas técnicas e fases de execução dos testes.

Quer conhecer as ações relacionadas à elaboração dos testes, ou mais especificamente, os casos de testes? Para isso, vamos continuar os estudos!

3.2 ELABORAÇÃO DO TESTE

A elaboração do teste representa o momento em que os cenários de testes serão criados para a sua seguinte documentação e posterior execução. Nesse processo de elaboração, é muito usual lançar mão do recurso de cenários e casos de testes, a fim de simular ações que serão executadas pelos usuários no uso do sistema, e validar o comportamento do sistema em meio a estes cenários de uso ou de teste.

Para cada cenário de teste, que é mais amplo e abstrato, tem-se normalmente um conjunto de casos de testes, com ações de teste mais específicas, como pequenos passos sequenciais de validação, dentro do cenário a ser testado.

Nas próximas etapas do estudo, você verá quais são os documentos necessários para a elaboração dos testes e o procedimento para definir um plano de casos de teste a definição destes casos.



FIQUE ALERTA

Fique alerta, na engenharia de software é comum o termo “Caso de Uso”, porém o “Caso de Teste” possui significado diferente, sendo que este pode ser originado de um caso de uso.

3.2.1 DOCUMENTO DE TESTE

A documentação do teste é muito importante para esta atividade profissional, pois todas as ações e o resultado destas precisam estar documentadas. Dessa forma, existem documentos específicos que você saberá a seguir.

Especificação do projeto de teste

Neste documento são apresentadas as funcionalidades e características do sistema que deverão ser testadas no projeto de testes. Ainda, são reunidas as informações que referenciam os casos de testes, seus respectivos procedimentos e os resultados esperados, para as funcionalidades e características do sistema, cobertas pelo projeto.

Plano do caso de teste

Os casos de teste são definidos neste documento, no qual cada caso de teste deverá possuir a descrição dos dados de entrada, os resultados de saída esperados, as ações e pré-condições para a execução do teste.

Especificação do procedimento de teste

A identificação dos passos de uso do sistema para contemplar cada caso de é descrita neste documento, ou seja, cada ação executada no sistema que estiver especificada nele irá guiar o analista de teste na execução completa do caso de teste.

Na sequência, veja em maiores detalhes o documento de Plano de Caso de Teste, com suas especificidades e exemplos de aplicação destes casos. Em frente!

3.2.2 PLANO DE CASO DE TESTE

No planejamento dos casos de teste serão descritos os diversos cenários de teste, que compreenderão os requisitos do sistema, além de um conjunto de casos de testes, que fará parte de cada cenário de teste, em que cada caso irá reunir informações usadas no cenário pertencente, os resultados esperados e a massa de dados necessária para as validações almejadas.

O caso de teste é a especificação mais detalhada do teste a ser executado, e deve conter estas especificações, dispostas na estrutura apresentada a seguir.

- a) Identificação das condições de teste: Pré-condições. Pós-condições. Critérios de aceitação.
- b) Identificação do próprio caso de teste ou do que será testado.
- c) Detalhes dos dados de entrada e saída.
- d) Critérios especiais, se for usada massa de dados para o teste.
- e) Especificação das configurações do ambiente de teste.
- f) Tipo de execução do teste (manual ou automático).
- g) Fase do teste em que o caso será executado.
- h) Identificar possíveis dependências entre outros casos de teste.

Para que a validação do teste seja efetiva e eficiente, além da estrutura de informações necessários para um caso de teste bem descrito, é importante avaliar se ele possui as características que seguem.

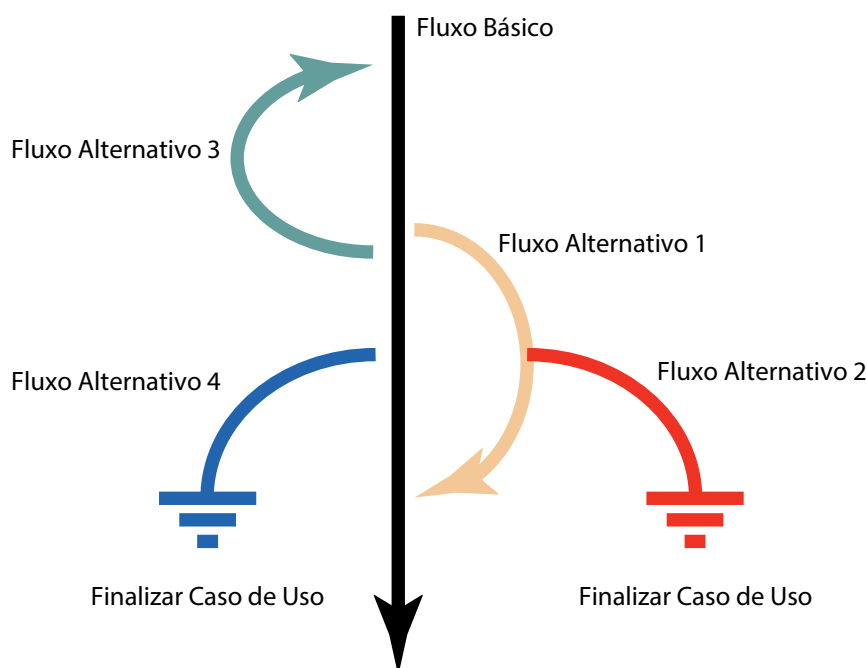
Características do Caso de Teste

Efetivo Testar exclusivamente o que se pretende testar
Econômico Descrição objetiva, sem procedimentos desnecessários
Reutilizável Pode ser usado em outras oportunidades
Rastreável Pode ser relacionado ao requisito validado
Autoexplicativo Pode ser testado por qualquer profissional

D'Imitre Camargo Martins (2019)

Figura 14 - Características do casos de teste
Fonte: Do autor (2019)

Os casos de teste são baseados em especificações funcionais do sistema, descritas em casos de uso. Como esses casos de uso podem possuir diferentes fluxos de uso de determinada funcionalidade, os casos de teste também devem possuir diferentes fluxos de testes, para espelhar os seus respectivos casos de uso.



D'Imitre Camargo Martins (2019)

Figura 15 - Fluxo básico e alternativos de um caso de uso
Fonte: Do autor (2019)

Como você viu, a figura representa o comportamento do fluxo principal ou básico de um caso de uso genérico e seus fluxos alternativos que, conforme mencionado, os casos de teste também precisam possuir nestes cenários.

**FIQUE ALERTA**

Para criar os casos de teste não funcional, como validações acerca das especificações de desempenho, segurança, usabilidade, entre outros, o analista de teste deve ter como fonte de referência os demais documentos do projeto de desenvolvimento que detalham estes tipos de especificação do sistema.

No próximo tópico serão apresentados os devidos procedimentos para executar o planejamento feito até aqui. Por isso, conclua esse ciclo de aprendizado sobre o processo de teste, acompanhando informações sobre a execução de testes.

3.3 EXECUÇÃO DO TESTE

A execução dos testes pode acontecer sobre testes estáticos, que compreendem a análise do documento de requisitos do sistema, e testes dinâmicos, que abordam a implementação, em diferentes níveis, dos requisitos do sistema, com foco nas funcionalidades e características não funcionais destas implementações. Neste tópico serão tratados essencialmente os testes dinâmicos, de acordo com os tipos de testes executados, assim como o registro da execução de tais testes.

Mas lembre-se de que a definição dos responsáveis pela execução dos testes é fundamental e, como já abordado no planejamento dos testes, essas responsabilidades devem estar expostas de forma clara no Plano de Teste do projeto. Veja na sequência os responsáveis por tipos de teste.

TIPOS DE TESTE	RESPONSÁVEIS
Teste unitário	Desenvolvedor
Teste de integração	Analista de sistema
Teste de sistema	Analista de teste
Teste de aceitação	Usuário e analista de teste

Quadro 11 - Relação de tipos de testes x responsáveis
Fonte: Do autor (2019)

A execução dos testes deve ser guiada pelos documentos previamente definidos, incluindo as instruções, as especificações dos ambientes e as ferramentas que devem ser usadas nesses procedimentos.

Durante a execução dos testes, os resultados obtidos devem ser analisados e registrados, para demonstrar o progresso da realização, os defeitos encontrados e os testes aprovados ou concluídos.

**SAIBA
MAIS**

O BSTQB (Brazilian Software Testing Qualification Board) é a organização brasileira que representa o ISTQB (International Software Testing Qualification Board) e provê certificações para analista de teste com reconhecimento internacional. Visite o site, a seguir, e confira mais detalhes a respeito! < <https://www.bstqb.org.br/>>.

A seguir será descrito o procedimento de teste por tipo, incluindo o teste unitário, teste de integração, teste de sistema e teste de aceitação. Siga atento!

3.3.1 EXECUÇÃO POR TIPO DE TESTE

Os tipos de teste descritos neste tópico – teste unitário, teste de integração, teste de sistema e teste de aceitação – compreendem os previstos no modelo V, e estão diretamente relacionados com as principais fases do desenvolvimento, correspondendo às fases ou estágios do teste. Dessa forma, esses tipos de teste terão este enfoque.

Teste unitário

Devem ser executados exclusivamente pelos programadores que implementam o código do sistema, por tratar-se também de implementações com o propósito de validar o resultado do processamento das funções implementadas no sistema.

A realização destes testes deve ser feita na mesma linguagem de programação em que o sistema está sendo codificado, podendo ser implementada pelo desenvolvedor que codificou a função a ser testada ou, ainda, por outro programador do time de desenvolvimento do projeto.

Este teste tem o objetivo de validar pequenas partes das funcionalidades do sistema, com o intuito de conferir maior qualidade ao código do sistema e reduzir erros de implementação que poderão impactar no correto funcionamento do sistema.

Teste de integração

Os testes de integração devem ser realizados após os testes unitários, pois estes consistem em validar a junção ou integração das diversas funções implementadas isoladamente e que, em conjunto, deverão fornecer o comportamento esperado para as funcionalidades do sistema.

Como este teste irá validar a interação entre as diferentes camadas do sistema, normalmente é realizado pelo analista de sistema, com principal atenção nas camadas relacionadas a seguir:

- a) Camada de apresentação.
- b) Camada de execução, onde as regras de negócio estão implementadas.
- c) Camada de dados, referente à estrutura da base de dados do sistema.
- d) Camada de rede, irá interligar algumas camadas do sistema.

Após a validação do funcionamento das diferentes funções de forma integrada, em todas as camadas do sistema, o processo tem andamento para o teste de sistema. Saiba mais sobre esse teste!

Teste de sistema

O teste de sistema irá validar o sistema em sua primeira versão disponível, com o propósito de verificar se o conjunto de implementação, configuração, ambiente, entre outros, atende os requisitos definidos para o sistema.

Este teste é realizado pelo analista de teste e deverá ser executado até obter as evidências de que o sistema apresenta o correto funcionamento quanto a estrutura e requisitos funcionais, seguindo as atividades macro listadas na sequência.

- a) Configurar o ambiente de teste com as características mais próximas do ambiente de produção.
- b) Criar dados de teste.
- c) Identificar os casos de testes que serão executados.
- d) Preparar scripts ou procedimentos de teste.
- e) Avaliar os resultados de teste e defeitos encontrados.
- f) Registrar os defeitos encontrados e notificar a equipe de desenvolvimento.
- g) Testar defeitos corrigidos.
- h) Finalizar defeitos corrigidos.
- i) Notificar novamente a equipe de desenvolvimento para defeitos não corrigidos em novos testes.

Após a realização e aprovação de todos os testes previstos para essa fase, o sistema encontra-se apto para ser disponibilizado ao cliente. Porém, ainda deve ser realizado o teste de aceitação, para que o sistema esteja disponível em ambiente de produção. Quer saber mais sobre o teste de aceitação? Continue com atenção!

Teste de aceitação

Esses testes representam uma homologação do sistema por parte do cliente ou sua equipe responsável pelo sistema, situação em que é comum haver apoio da equipe de testes.

Normalmente os envolvidos nesse teste não possuem conhecimento técnico sobre o sistema. Dessa forma, a validação concentra-se na camada de apresentação e nas regras de negócio implementadas no sistema.

Após a validação desses testes, o cliente aceita formalmente a entrega do sistema e esta versão deverá ser disponibilizada no ambiente de produção, concluindo o ciclo de testes do projeto.



CASOS E RELATOS

Fases de teste no desenvolvimento

Em uma tradicional empresa de desenvolvimento de sistemas, havia uma equipe altamente qualificada, com excelentes analistas de sistemas e programadores, no entanto os clientes reportavam frequentemente falhas no sistema ou funcionalidades que não estavam de acordo com as especificações realizadas. Com as reclamações recorrentes a direção desta empresa resolveu implantar um processo de teste estruturado baseado no modelo V, onde em cada fase do desenvolvimento haveria uma etapa correspondente de teste, assim reduzindo o risco de erros entregues aos clientes. Após a realização desta ação da empresa, os sistemas desenvolvidos tornaram-se mais robustos, atendendo com mais precisão as necessidades de seus usuários.

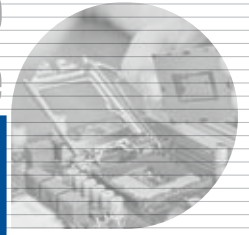


RECAPITULANDO

Nesta etapa do seu estudo, você conheceu o planejamento e a execução de testes, e aprendeu a elaborar um documento de testes que vai definir os procedimentos para identificar a estratégia de teste. Esse plano deve possuir três propósitos: repetição, controle e cobertura. Você viu também que quanto mais cedo forem identificados os defeitos dos sistemas, mais cedo poderão ser corrigidos, evitando custos desnecessários para a organização. Também aprendeu que a execução pode acontecer em testes estáticos ou dinâmicos, sendo este último utilizado para exemplificar o registro da execução dos testes neste capítulo. Por fim, você estudou a execução dos seguintes tipos de testes: unitário, integração, teste de sistema e de aceitação.

Automação do Processo de Teste

4



Neste capítulo, você irá conhecer os conceitos de automação de testes além dos tipos de testes automatizados, como testes progressivos e regressivos, drivers e stubs, testes unitários, testes funcionais e de performance e alguns testes que são candidatos a automação. Você também vai conhecer como é realizada a automação de gerência de planejamento de testes. Sendo assim, ao final dos estudos deste capítulo você terá subsídios para:

- a) compreender os conceitos de automação de testes;
- b) reconhecer os tipos de testes automatizados;
- c) identificar como é realizada a automação de gerência de planejamento de testes.

Inicie seus estudos compreendendo os conceitos de automação de testes. Bons estudos!

4.1 CONCEITOS DE AUTOMAÇÃO DE TESTES

Você sabe qual é o objetivo da automação de teste? É aumentar a eficiência na execução, para que mais testes possam ser realizados em menos tempo e com resultados mais precisos, possibilitando, assim, o aumento na cobertura de testes sobre o projeto.



Nesse contexto, é importante afirmar que o papel do analista de teste é imprescindível, pois mesmo que a carga de trabalho manual seja reduzida com a automação, os testes automatizados precisam ser desenvolvidos por profissionais de teste, nesse caso, mudando o foco de atuação desse profissional.

Automação nos testes pode parecer recente, porém é um movimento que iniciou há quase trinta anos e, desde então, vem evoluindo de acordo com as tecnologias de seu tempo, iniciando com ferramentas de captura de texto em mainframes até os frameworks comerciais dos dias atuais.

Algo importante a ser destacado são as dificuldades inerentes à adoção de testes automatizados. Essas dificuldades já foram mapeadas, tendo em vista sua recorrência, e são apresentadas a seguir.

Expectativas irreais: É preciso estar atendo aos problemas que devem e podem ser solucionados pela automação de testes, para que as expectativas não sejam frustradas por intenções irreais de uso da automação.

Suporte ao teste automatizado: Para conservar as ferramentas de automação de testes com as configurações adequadas a suas aplicações e com uso de forma objetiva, é necessário manter um suporte especializado, assim como o apoio da gerência para preservar esta atividade.

Escolha de profissionais capacitados: Os profissionais envolvidos nos testes automatizados devem possuir competências no uso das ferramentas de automação, assim como conhecer o processo de teste.

Novo projeto para testes: O teste automatizado deve ser visto como um novo projeto, derivado do projeto de teste, mas com a finalidade de manter e pesquisar por ferramentas que atendam à demanda da organização, porém é comum os gestores não fazerem esta distinção de projeto.

Nesta introdução sobre automação de teste foram destacados aspectos gerais do tema. A seguir serão apresentados os conceitos da automação de teste, os tipos de testes automatizados e definições diretamente relacionadas ao tema.

Você vai entender quando os testes devem ser manuais ou automatizados e, caso a segunda opção seja a resposta, como implementar este tipo de teste.

Para iniciar seu estudo, é importante destacar a diferença entre teste e automação de teste. Conceitualmente, o teste é uma atividade realizada por um testador para aferir a qualidade ou o correto funcionamento de um sistema ou parte dele. Já a automação de teste é a ação de um software específico que irá simular as ações de um testador.

Neste conceito também são destacados três dimensões ou visões sobre o teste automatizado.

a) Visão efetiva: Apesar de haver automação no teste, a efetividade para atingir seu objetivo deve ser a mesma de um teste manual.

b) Visão evolutiva: O teste manual vai evoluir mais rapidamente que um automatizado, devido à flexibilidade do testador e à necessidade de manutenção do software de automação.

c) Visão econômica: O teste automatizado é mais econômico de acordo com a quantidade de vezes que é reutilizado, porém se for usado somente uma vez tende a ser mais caro que um teste manual.

A seguir são apresentados conceitos gerais relacionados às características das ferramentas de automação de teste.

Recorder

É uma espécie de gravador que registra as ações do usuário e armazena a gravação na forma de um script de teste.

Script

É o resultado da gravação de uma ferramenta de automação e pode ser executado para repetir as ações gravadas anteriormente pela ferramenta.

Playback

É o recurso da ferramenta que possibilita a execução do script de teste com os dados gravados de teste.

Think time

Recurso da ferramenta de automação de teste para simular um atraso na execução do teste, com o intuito de testar um atraso natural no uso do sistema testado.

Data-driven test

É o uso de uma massa de dados em um script de teste, no qual a execução do script será dirigida pelos dados inseridos no teste.

D'Imire Camargo Martins (2019)

Figura 16 - Conceitos gerais sobre ferramentas de automação de teste
Fonte: Do autor (2019)

Características de ferramentas de automação de teste de performance:

Virtual user

É a simulação de um usuário do sistema testado, sendo possível criar muitos usuários virtuais simultâneos.

Controller

É o controle, por meio de monitoramento, da execução dos scripts de teste.

Agent

É um agente ou instância da ferramenta de automação responsável por executar o script de teste na máquina em que está instanciado.

Ponto de Sincronização

É a sincronização de usuários virtuais, criados pela ferramenta de automação de teste, necessária quando os comandos de execução do script de teste são enviados pela rede de dados.

D'Imitre Camargo Martins (2019)

Figura 17 - Teste de performance
Fonte: Do autor (2019)

Características de ferramentas de automação de teste funcional:

Property check point

É um recurso da ferramenta de automação usado para verificar as propriedades de um objeto de interface.

Image region check point

É um recurso da ferramenta de automação usado para verificar a região da interface da aplicação testada, com base em uma imagem previamente armazenada.

Text check point

É um recurso da ferramenta de automação usado para verificar se os textos da interface da aplicação testada estão corretos.

Object mapping

É um recurso da ferramenta de automação usado para verificar as propriedades de um objeto de interface e utilizá-lo ao longo da execução do script de teste.

D'Imitre Camargo Martins (2019)

Figura 18 - Teste funcional
Fonte: Do autor (2019)

**CURIOSIDADES**

Você sabia que a implantação de automação de testes pode requerer um projeto específico, com planejamento do que deve ser automatizado, custos e recursos necessários e cronograma de execução? Ao pensar em automação de testes avalie estes fatores!

Agora que você já viu as características das ferramentas de automação de teste, conheça os principais tipos de testes automatizados. Continue atento!

4.1.1 TIPOS DE TESTES AUTOMATIZADOS

Existem diferentes tipos de automatização de teste, para aplicar em diferentes fases do projeto de teste ou para diferentes propósitos de validações do sistema a ser testado.

Neste tópico serão relacionados os principais tipos de automação de teste, de acordo com as ferramentas de automação disponíveis no mercado, como segue:

a) Automação para Planejamento de Teste

Usado para planejar os testes, criar os requisitos de teste e casos de teste.

b) GUI Test Drivers por Command Script

Usado para execução dos testes em GUI (Graphical User Interface) ou interface gráfica, situação em que as ações de teste gravadas e armazenadas em scripts podem ter seus scripts customizados, com recursos de programação, para novos cenários de testes.

c) GUI Test Drivers por Visual Script

Usado para execução dos testes em GUI (Graphical User Interface) ou interface gráfica, mas, nesse caso, o acesso ao script gerado é totalmente visual, impossibilitando maiores customizações.

d) Teste de Carga e Performance

Usado para testar desempenho e performance do sistema por meio da simulação de entrada de uma grande massa de dados para processamento ou de vários usuários utilizando o sistema simultaneamente.

e) Test Execution Managers

Usado para a gestão da automação de testes.

f) Code Coverage

Usado para analisar o código do sistema em tempo de execução, ou seja, enquanto o sistema está sendo executado.

g) Defect Tracking

Usado para manter e gerenciar um banco de dados com os defeitos encontrados nos testes realizados.

h) Unit Testing

Usado para o teste unitário do código implementado, aplicado em pequenas unidades do código ou funções específicas.

4.1.2 TESTES PROGRESSIVOS E REGRESSIVOS

A aplicação da automação de testes pode atender a dois perfis diferentes de teste que, consequentemente, possuem propósitos diferentes. São os progressivos e os regressivos. Saiba mais sobre cada um deles.

Os testes regressivos buscam a validação de testes já realizados no passado, porém sobre novas versões do mesmo sistema. Isso quer dizer que quando o sistema sofre uma atualização, decorrente de manutenção ou evolução, funcionalidades que possam ser afetadas por estas mudanças no sistema devem ser testadas novamente, para garantir que seu funcionamento permaneça o mesmo, comparado à versão do sistema anterior a tais mudanças.

Já a aplicação dos testes progressivos tem como objetivo validar melhorias implementadas no sistema a fim de identificar novos limites de processamento, de carga, de desempenho, entre outros, para garantir a evolução do sistema e reduzir o risco quanto a limitações no uso do sistema.

4.1.3 DRIVERS X STUBS

O objetivo deste tópico não é mostrar o desenvolvimento de Drivers ou Stubs de teste, mas definir os conceitos destas técnicas que auxiliarão na automação de testes.

O Driver pode ser considerado uma interface simulada, usada para conectar-se ao sistema ou módulo a ser testado em casos em que o teste precisaria de um sistema externo para fornecer dados de entrada para o sistema testado.

O Stub possui o papel inverso do Driver quanto à simulação de dados do sistema, ou seja, nesse caso, o Stub irá simular uma interface para tratar a saída de dados do sistema testado.

4.1.4 TESTES CANDIDATOS À AUTOMAÇÃO

Como você viu, a automação de teste passa a ser interessante quando ele precisa ser repetido ao longo do projeto ou para futuras versões do sistema testado, entre outras características. Desse modo, apesar de todos os testes estarem passíveis de automação, é importante identificar aqueles cuja automação será mais vantajosa. A seguir, acompanhe algumas características referentes à escolha de testes para automação.

Teste pontual

Testes realizados pontualmente para atender uma demanda específica e que provavelmente não tornarão a ser executados não devem ser automatizados.

Teste reaproveitado

Testes que devem ser reutilizados no futuro, para manutenção ou evolução do sistema, como testes regressivos, são bons candidatos à automação.

Teste com massa de dados

Testes que precisam usar grande massa de dados para inserir no sistema testado, são propícios à automação.

Cronograma do teste

Testes que possuem as características para serem automatizados, mas em razão da falta de tempo hábil no cronograma de teste, não haveria tempo disponível para investir na automação, desta forma este teste pode ser sinalizado para uma futura automação.

Teste de defeito

Testes para validar a correção de defeitos são fortes candidatos a automação, pois normalmente testes como esses devem ser executados em novas versões do sistema testado.

D'Imitre Camargo Martins (2019)

Figura 19 - Escolha de testes para automação
Fonte: Do autor (2019)

**FIQUE ALERTA**

A automação de teste deve ser aplicada com base em avaliação criteriosa, para que o custo e esforços de sua implantação possam ser justificados.

4.2 AUTOMAÇÃO DE TESTES UNITÁRIOS

Como já conceituado no decorrer deste material didático, o teste unitário consiste em realizar testes sobre o código do sistema, em pequenas partes desse código, que pode ser uma classe, uma função, um método, uma rotina, entre outras classificações que agrupam trechos do código do sistema.

Metodologias de desenvolvimento ágil ou que usam o TDD, desenvolvimento dirigido por testes, fomentam a criação de testes unitários que podem ser reutilizados no futuro, evidenciando uma razão para haver automação sobre este tipo de teste.



monsiti(2018)

Além da razão mencionada para automatizar os testes unitários, existem outras diversas vantagens, relacionadas a seguir:

- a) Problemas são encontrados mais cedo, durante o desenvolvimento do sistema.
- b) Proveem garantias de funcionamento, principalmente quando houver mudanças, aplicando teste regressivo.
- c) Proporcionam a abertura para mudança do código quando necessário.
- d) Possibilitam aos desenvolvedores que não construíram o código conhecê-lo.
- e) Tendem a tornar o código mais confiável.
- f) Aumentam a cobertura de testes sobre o sistema desenvolvimento.

Após entender a importância e vantagens sobre a automação desse tipo de teste, veja como ela poderá ser implementada, por meio de conceitos de testes unitários automatizados e um exemplo prático.

Os conceitos referentes à automação de testes unitários são:

a) Unidade de teste

A parte do código que será testada

b) Caso de teste unitário

Código criado para evocar o código a ser testado

c) Suíte de teste

Conjunto de casos de teste unitário

É importante ressaltar as pré-condições necessárias para iniciar a implantação da automação de testes unitários. Ou seja, o código deve estar pronto na parte que for testado e deve haver documentação que descreva o resultado esperado da parte do código testado. Naturalmente, antes de iniciar a execução dos testes, também devem ser escritos os casos de testes que devem estar associados às suas respectivas suítes.

Com base nos conceitos e pré-condições apresentadas, é possível apresentar um exemplo prático, com a ferramenta JUnit.

Mas o que é o JUnit? É uma ferramenta específica para testes unitários, com classes específicas de testes já disponíveis, que pode ser integrada ao IDE (Integrated Development Environment) ou ambiente de desenvolvimento, no qual, nesse caso, iremos usar o Eclipse.

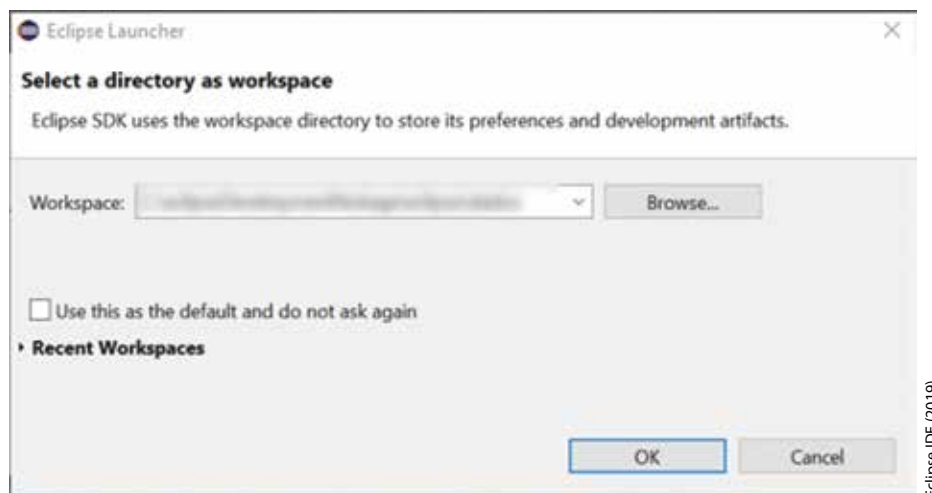
Esta ferramenta é indicada para desenvolvimentos que utilizam a linguagem de programação Java. Porém, existem outras ferramentas com a mesma estratégia de uso do JUnit, aplicadas a outras linguagens de programação orientada a objeto.

Neste exemplo, será criada uma classe Java no IDE Eclipse, que será testada por meio do JUnit, que por sinal já é pré-instalado no Eclipse.

Assim, basta realizar o download do Eclipse no link <<https://www.eclipse.org/downloads/>>.

Após acompanhe o exemplo que segue:

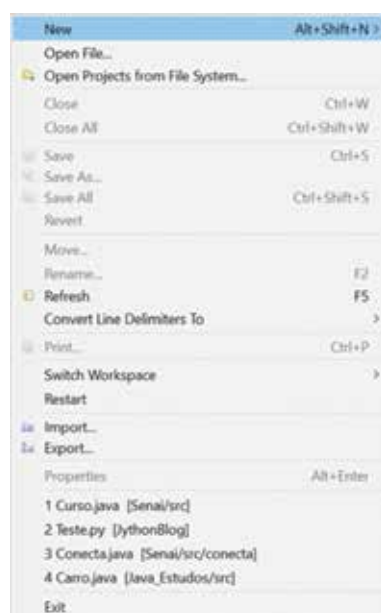
- a) Ao executar o Eclipse, defina a área de trabalho ou “Workspace”, sendo este o diretório do computador em que ficará salvo o código desenvolvido no Eclipse, como segue na próxima figura:



Eclipse IDE (2019)

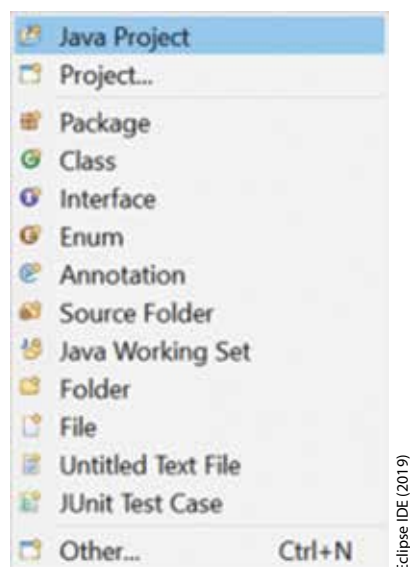
Figura 20 - Seleção do Workspace do Eclipse
Fonte: Eclipse IDE (2019)

- b) Após definir o Workspace, o Eclipse será aberto e será necessário criar um projeto Java, por meio do menu “File>New>Java Project”, como é demonstrado nas próximas figuras:



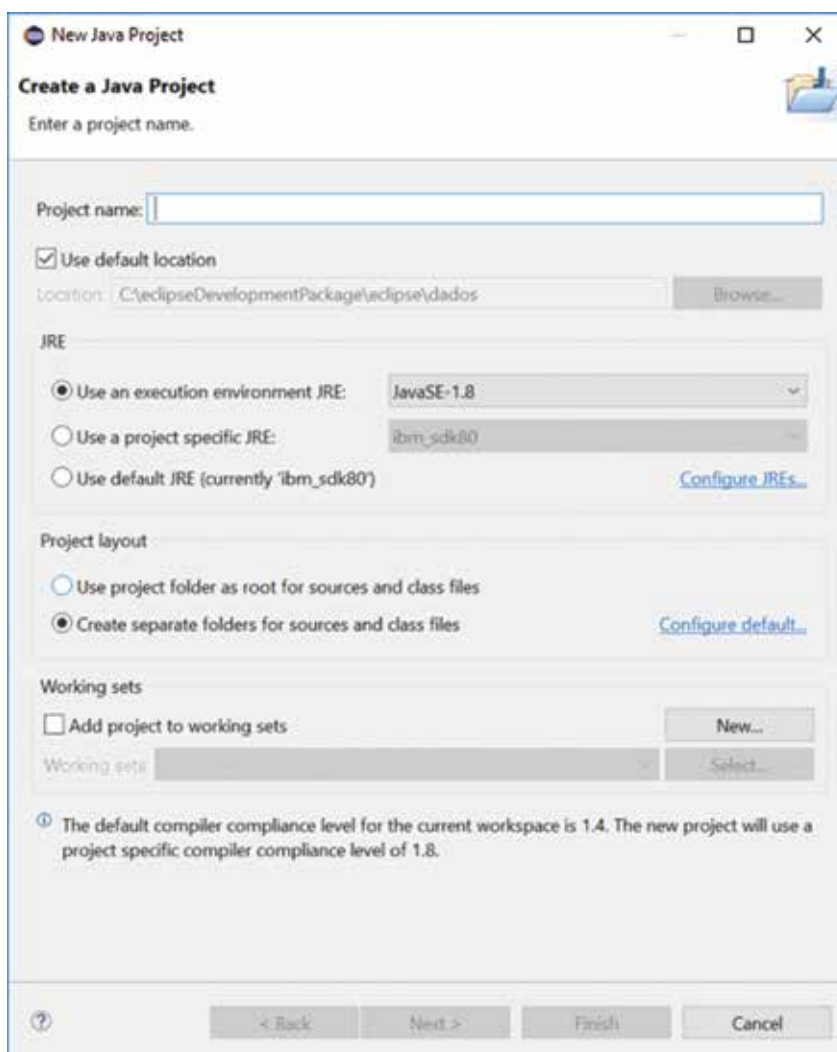
Eclipse IDE (2019)

Figura 21 - Menu para criar novo item no Eclipse
Fonte: Eclipse IDE (2019)



Eclipse IDE (2019)

Figura 22 - Menu para criar novo projeto Java
Fonte: Eclipse IDE (2019)



Eclipse IDE (2019)

Figura 23 - Tela de cadastro do projeto Java
Fonte: Eclipse IDE (2019)

c) Com o projeto criado, o próximo passo é criar uma classe Java, que será a unidade de teste do exemplo, clicando com o botão direito do mouse sobre o projeto criado no passo anterior, como segue:



Figura 24 - Menu para criar novo item no Eclipse
Fonte: Eclipse IDE (2019)

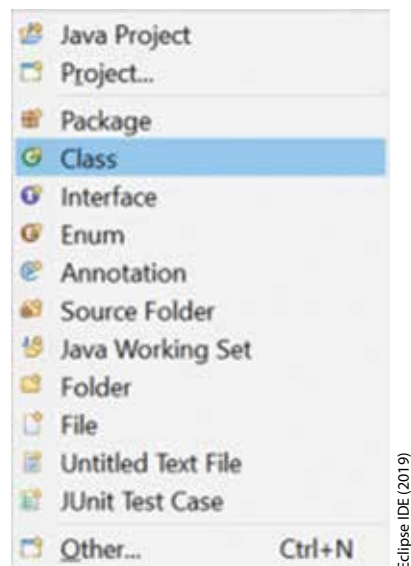


Figura 25 - Menu para criar nova classe Java
Fonte: Eclipse IDE (2019)

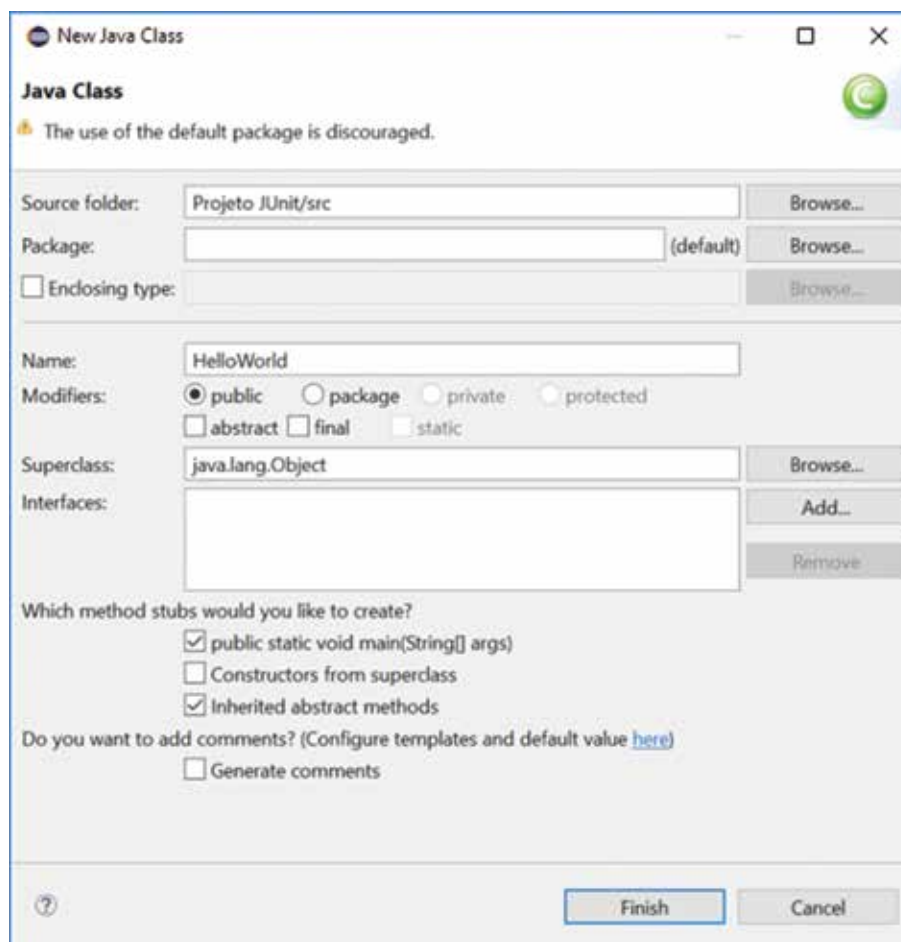


Figura 26 - Tela de cadastro da classe Java
Fonte: Eclipse IDE (2019)

d) Após criar a classe e abri-la no editor do Eclipse, deve ser inserido o código ilustrado na próxima figura, para que o teste possa ser representado com base no retorno desta simples classe.

```
1  
2 public class HelloWorld {  
3  
4     public static String say() {  
5  
6         return "Hello, World!";  
7     }  
8 }  
9  
10  
11
```

Figura 27 - Área de edição da classe Java
Fonte: Eclipse IDE (2019)

e) Logo após criada a classe Java que será testada, devemos criar o teste automatizado sobre ela, clicando com o botão direito do mouse sobre ela como pode ser visto a seguir.



Figura 28 - Menu para criar novo item no Eclipse
Fonte: Eclipse IDE (2019)

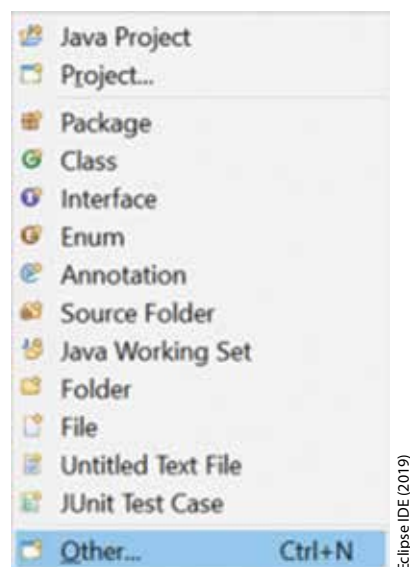
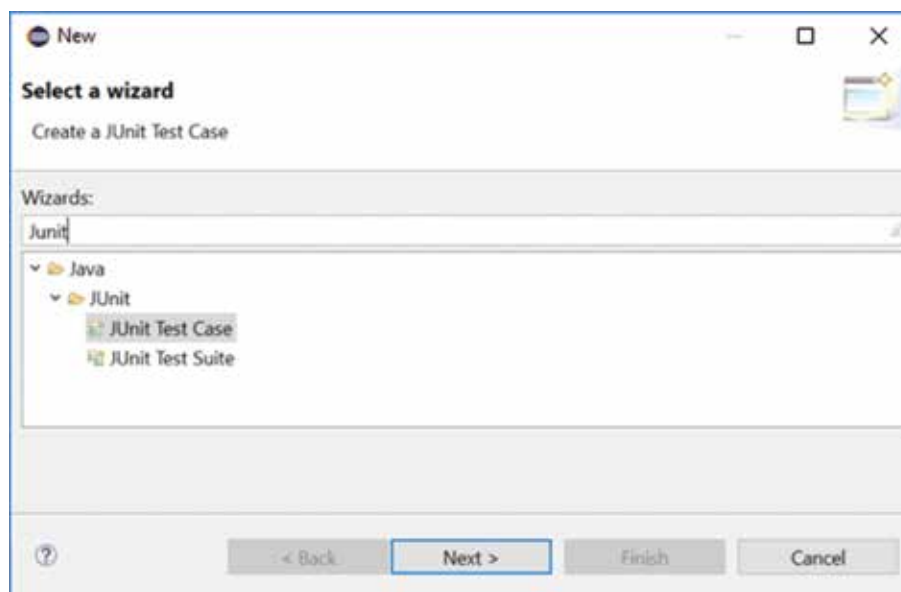
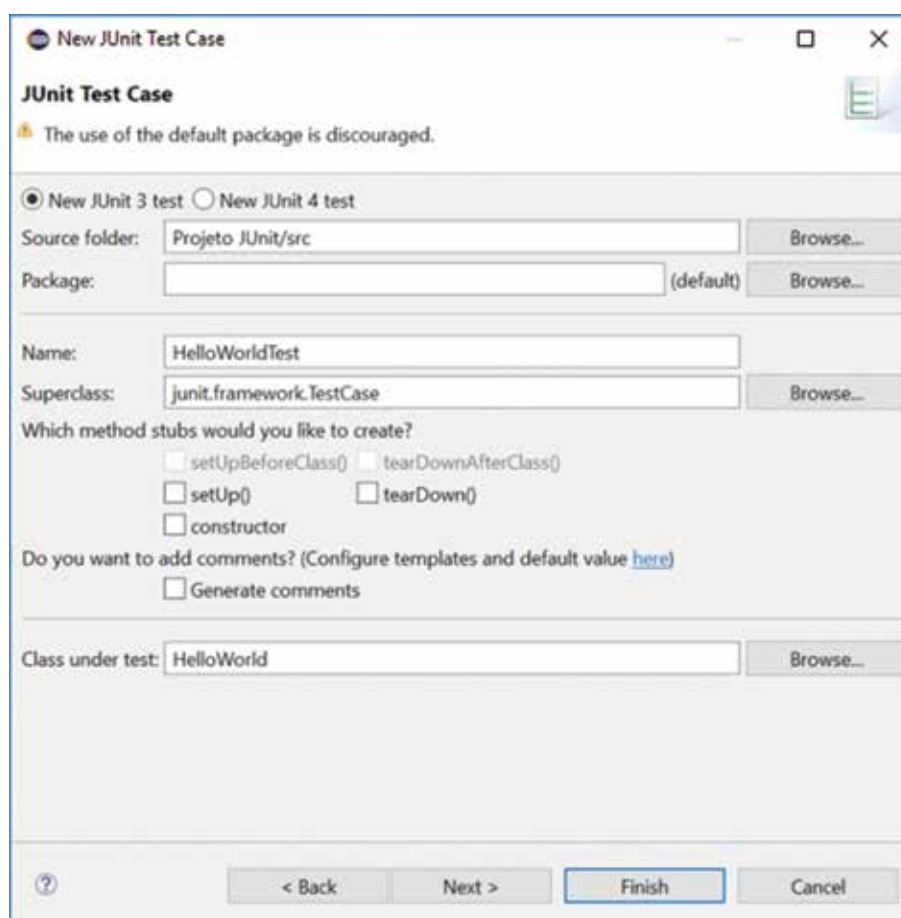


Figura 29 - Menu para criar novo item não especificado no menu
Fonte: Eclipse IDE (2019)



Eclipse IDE (2019)

Figura 30 - Tela de busca do novo item a ser criado no Eclipse
Fonte: Eclipse IDE (2019)



Eclipse IDE (2019)

Figura 31 - Tela de cadastro do caso de teste JUnit
Fonte: Eclipse IDE (2019)

f) Na sequência, será inserido o código do caso de teste, como pode ser visto na figura.



```
1 import junit.framework.TestCase;
2
3 public class HelloWorldTest extends TestCase {
4
5     public void testExecutavelHelloWorld() {
6
7         String StringEsperado = "Hello";
8         String StringRetorno = "HelloWorld say!";
9
10        assertEquals(StringEsperado, StringRetorno);
11    }
12 }
13
14
15
```

Figura 32 - Área de edição da classe de teste Java
Fonte: Eclipse IDE (2019)

Observe que neste código de teste há a comparação entre o resultado esperado e o resultado produzido pela execução da classe Java que está sendo testada, deste modo, como a classe Java deve retornar o valor "Hello, World!" e o valor esperado informado é "Hello", iremos simular uma situação em que o teste apresenta a falha na classe Java, como pode ser visto ao executar o teste, clicando com o botão direito do mouse sobre o caso de teste, e demonstrado a seguir:

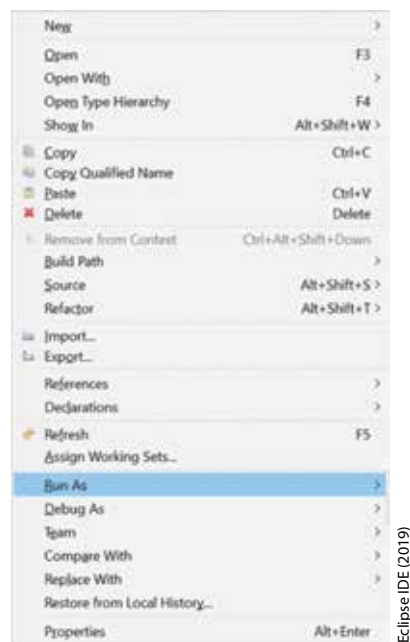


Figura 33 - Menu para executar um item no Eclipse
Fonte: Eclipse IDE (2019)

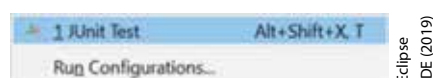


Figura 34 - Seleção para executar o teste pelo JUnit
Fonte: Eclipse IDE (2019)



Eclipse IDE (2019)

Figura 35 - Tela de resultado do teste pelo JUnit
Fonte: Eclipse IDE (2019)

g) A seguir, o código do teste foi alterado para simular uma situação de aprovação do teste executado, como pode ser visto na próxima figura.



Eclipse IDE (2019)

Figura 36 - Área de edição da classe de teste Java
Fonte: Eclipse IDE (2019)

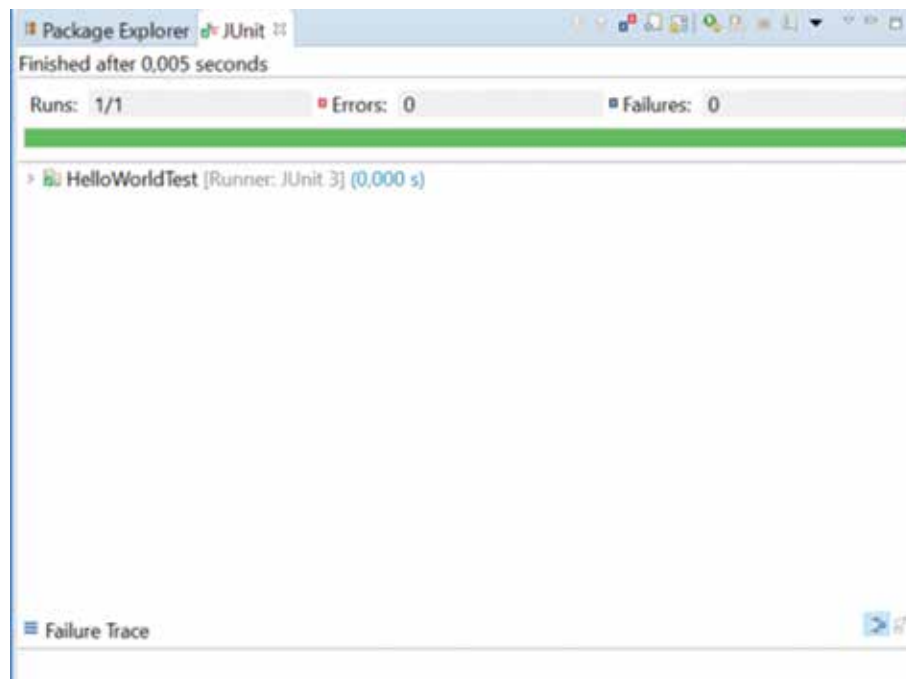


Figura 37 - Tela de resultado do teste pelo JUnit
Fonte: Eclipse IDE (2019)

h) Neste momento, para simular a criação de uma suíte de testes, iremos duplicar o caso de teste existente, no qual um deles terá o código que produzirá como resultado uma falha e outro que produzirá um resultado de teste sem falhas. Com o mesmo procedimento para criar um caso de teste, iremos criar a suíte de teste. Veja!

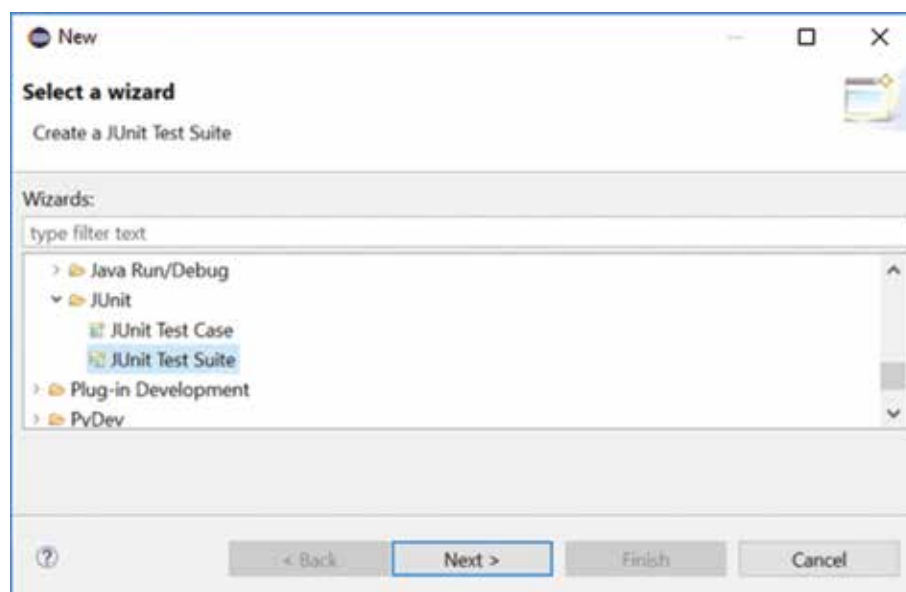


Figura 38 - Tela de busca do novo item a ser criado no Eclipse
Fonte: Eclipse IDE (2019)

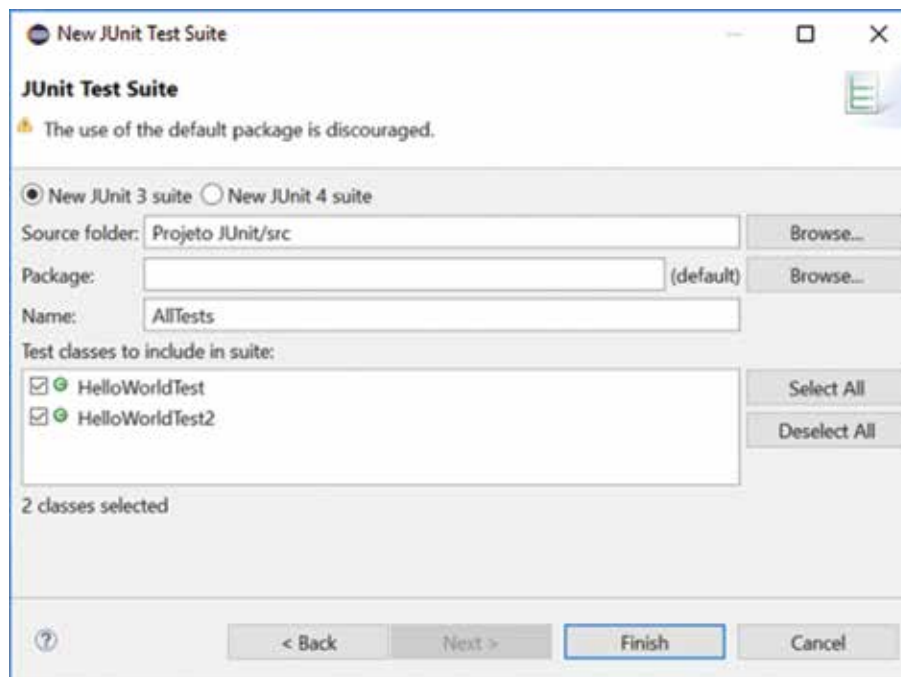


Figura 39 - Tela de cadastro da suíte de teste JUnit
Fonte: Eclipse IDE (2019)

```
1 import junit.framework.Test;
2
3 public class AllTests {
4     public static Test suite() {
5         TestSuite suite = new TestSuite(AllTests.class.getName());
6         // $JUnit4-BEGIN$
7         suite.addTestSuite(HelloWorldTest.class);
8         suite.addTestSuite(HelloWorldTest2.class);
9         // $JUnit4-END$
10        return suite;
11    }
12 }
13
14
15
16
```

Fonte: Área de edição da classe da Suíte de teste Java
Fonte: Eclipse IDE (2019)

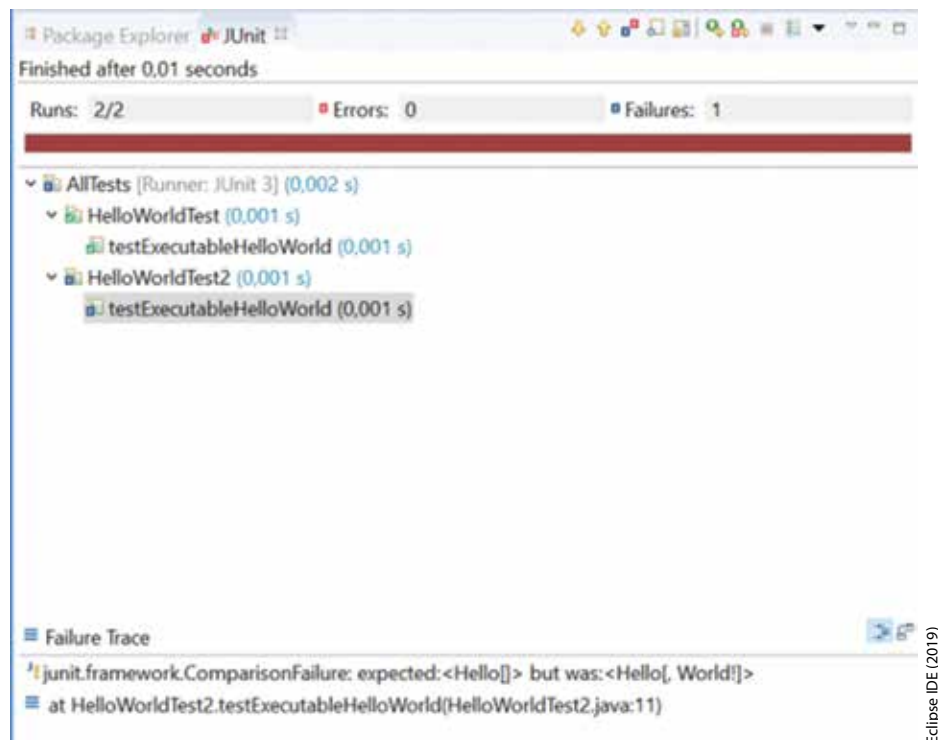


Figura 40 - Tela de resultado do Teste pelo JUnit
Fonte: Eclipse IDE (2019)

Esta foi uma demonstração de teste unitário automatizado, com o uso da ferramenta JUnit, para aplicação de forma dinâmica ao código do sistema em desenvolvimento.

A seguir você poderá acompanhar outros exemplos de aplicação de automação de testes. Vamos lá!

4.3 AUTOMAÇÃO DE GERÊNCIA E PLANEJAMENTO DE TESTES

Você sabia que a automação também pode ser aplicada no gerenciamento e planejamento de testes? É verdade! Seu intuito principal é gerir os casos de teste, mantendo um repositório da descrição desses casos para consulta ou, ainda, para alterações ou inserções de novos casos, tendo em vista que o processo de teste é dinâmico e pode surgir a necessidade de novos testes durante a fase de execução.

Outro aspecto importante neste tipo de automação de teste é gerenciar, via ferramenta, o processo de teste ou, mais precisamente, os resultados chave e ações a serem realizadas durante o andamento do processo.

A principal vantagem de automatizar a gestão dos testes é a melhoria da qualidade dos testes, pelo fato de o processo de teste estar suportado por uma ferramenta que disponibilizará insumos para análises mais certeiras sobre os testes já executados no processo.

No entanto, a implantação dessa automação precisa observar alguns critérios e contextos para que se possa obter os resultados esperados. Confira!

- Verificar a possibilidade de ajustes do processo de teste, para possível adequação à ferramenta utilizada na automação.
- Realizar a capacitação dos envolvidos, observando a curva de aprendizado necessária.
- Iniciar a automação em projeto piloto de menor porte, após a devida capacitação dos envolvidos.
- Buscar o apoio da direção da organização.
- Identificar as métricas que devem ser apresentadas pela ferramenta.

Para o exemplo prático desta forma de automação de teste, será apresentada a ferramenta TestLink, que possui como principal objetivo o controle do ciclo de vida do teste, ou seja, seu planejamento, execução e resultados obtidos. Saiba que essa ferramenta é extremamente robusta e possui a capacidade de gerir milhares de casos de teste, tendo como pré-requisitos um banco de dados dedicado e um servidor web com suporte a PHP. Ou seja, diante deste cenário, há diferentes formas de realizar a instalação, mas para efeitos didáticos, a instalação pode ser realizada por meio de ferramentas que simulam o ambiente necessário, como é o caso do XAMPP.

A instalação do XAMPP busca a simplicidade e facilidade, criando o ambiente necessário em poucas etapas do processo de instalação, as quais podem ser visualizadas no link: https://www.apachefriends.org/pt_br/index.html.

Após instalação deverá ser possível visualizar o painel do XAMPP com os módulos Apache e MySQL em execução, como pode ser observado na figura a seguir.



Figura 41 - Painel do XAMPP com os módulos Apache e MySQL em execução
Fonte: XAMPP Apache (2019)

Para realizar o download do TestLink para Windows, um dos caminhos é o link <<https://bitnami.com/stack/wamp/modules#testlink>>, que disponibilizará um arquivo executável de instalação. Também é possível acessar o link <<http://testlink.org/>> e escolher outras formas de obter a ferramenta.

No procedimento de instalação que segue, é utilizado o arquivo executável de instalação, o qual possui como pré-requisito a instalação do pacote XAMPP, conforme mencionado anteriormente.

As figuras a seguir demonstram todos os passos da instalação e uso inicial da ferramenta:

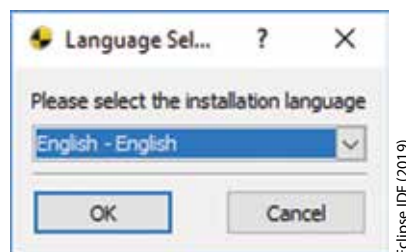


Figura 42 - Seleção da linguagem para instalação
Fonte: TestLink (2019)

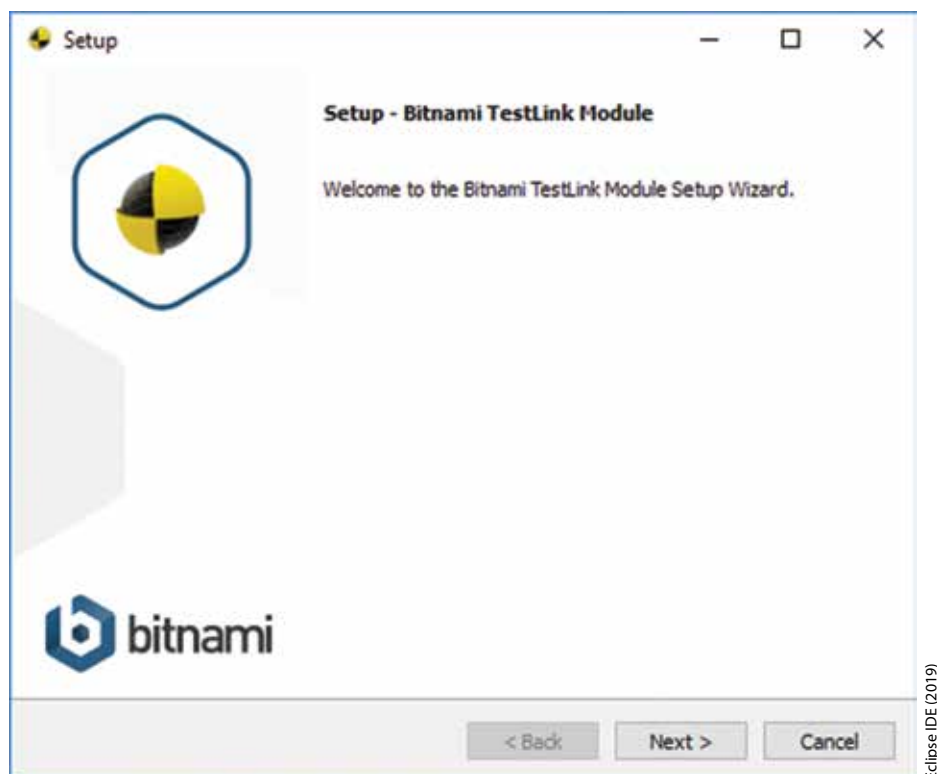
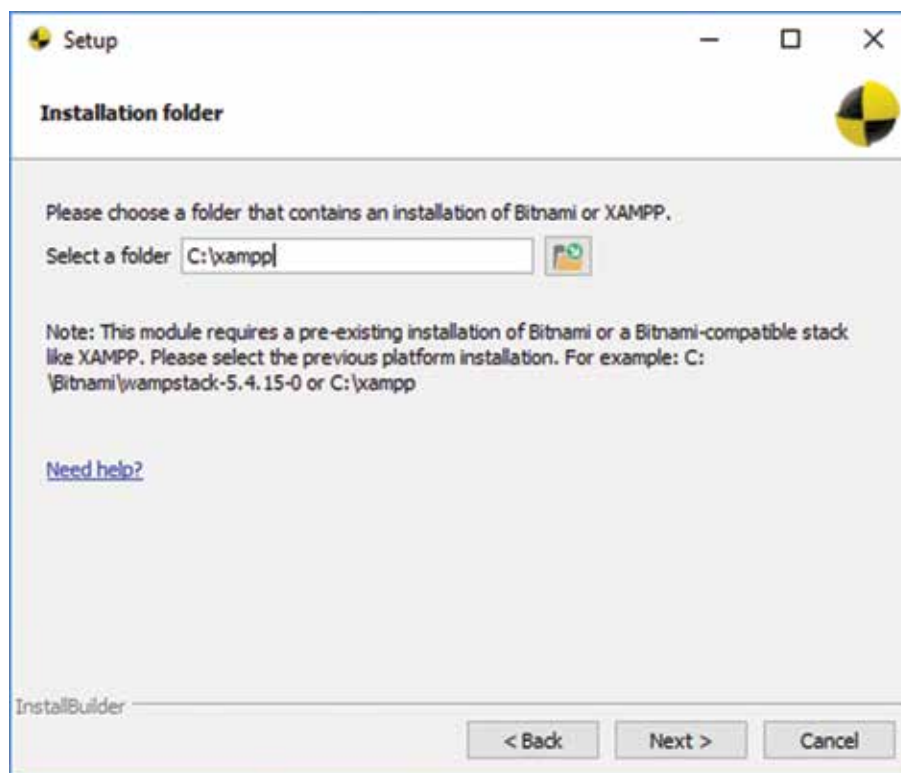


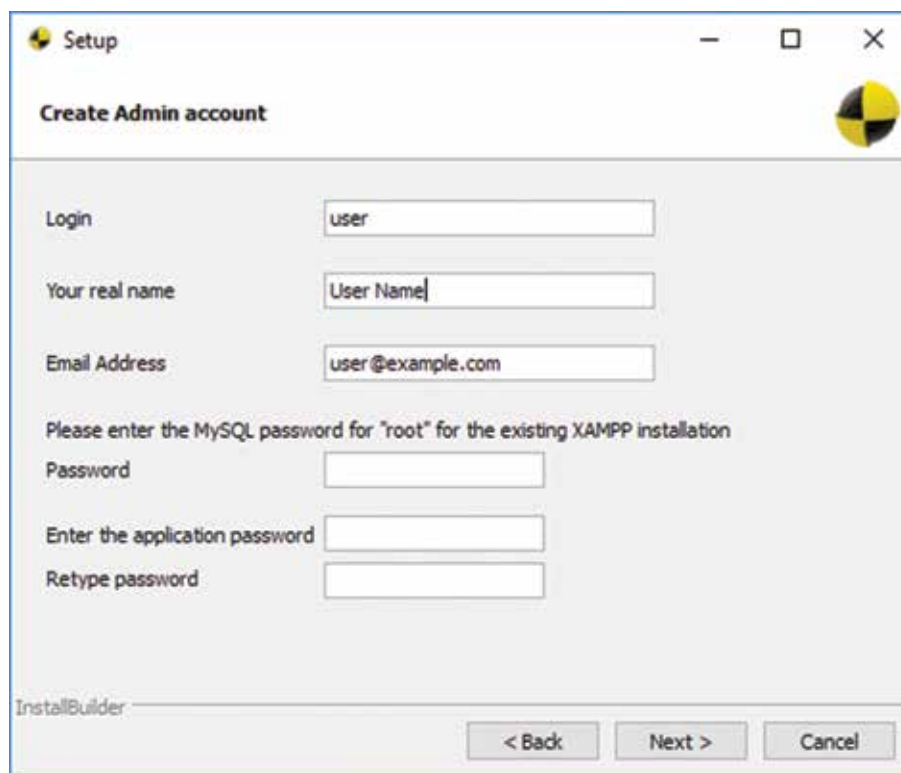
Figura 43 - Tela inicial do procedimento de instalação
Fonte: TestLink (2019)



Eclipse IDE (2019)

Figura 44 - Tela para selecionar diretório de instalação do XAMPP

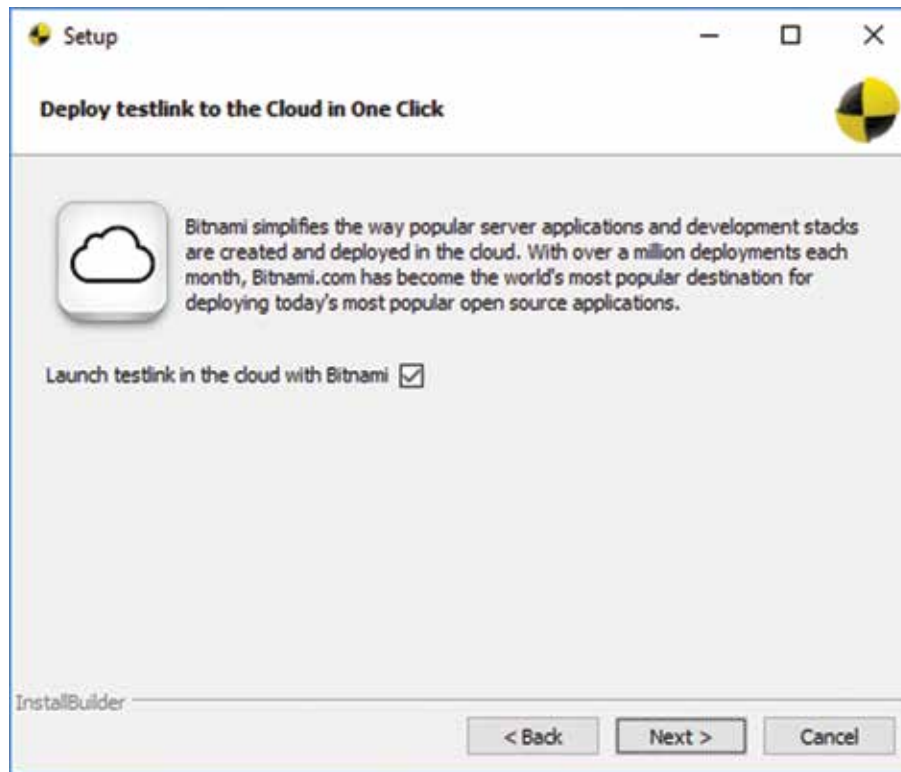
Fonte: TestLink (2019)



Eclipse IDE (2019)

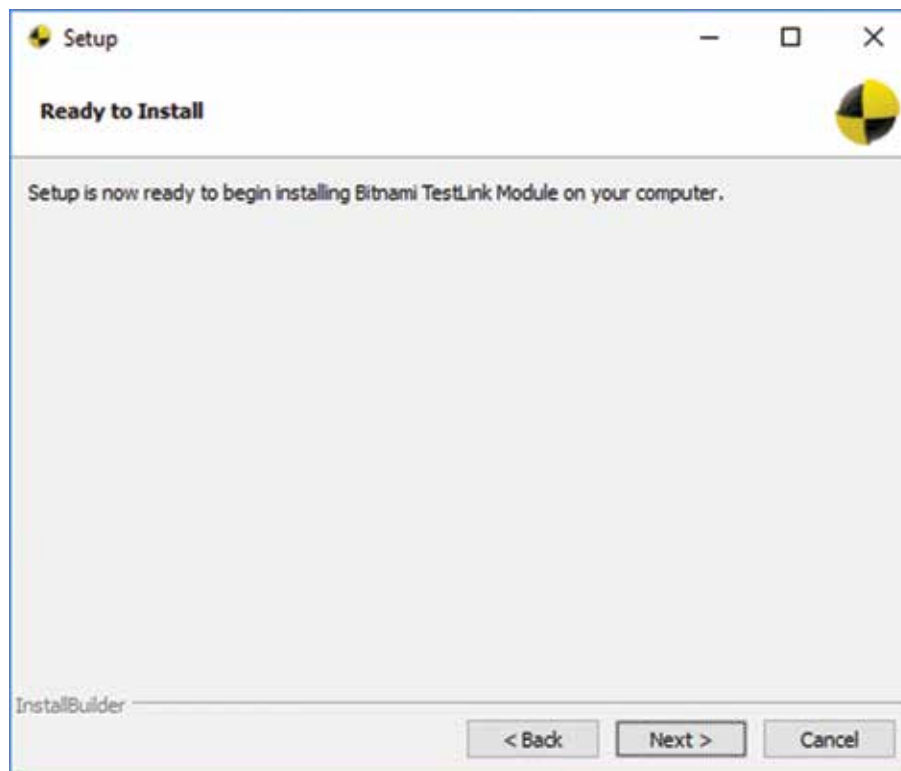
Figura 45 - Tela para definir usuário e senha da ferramenta e banco de dados

Fonte: TestLink (2019)



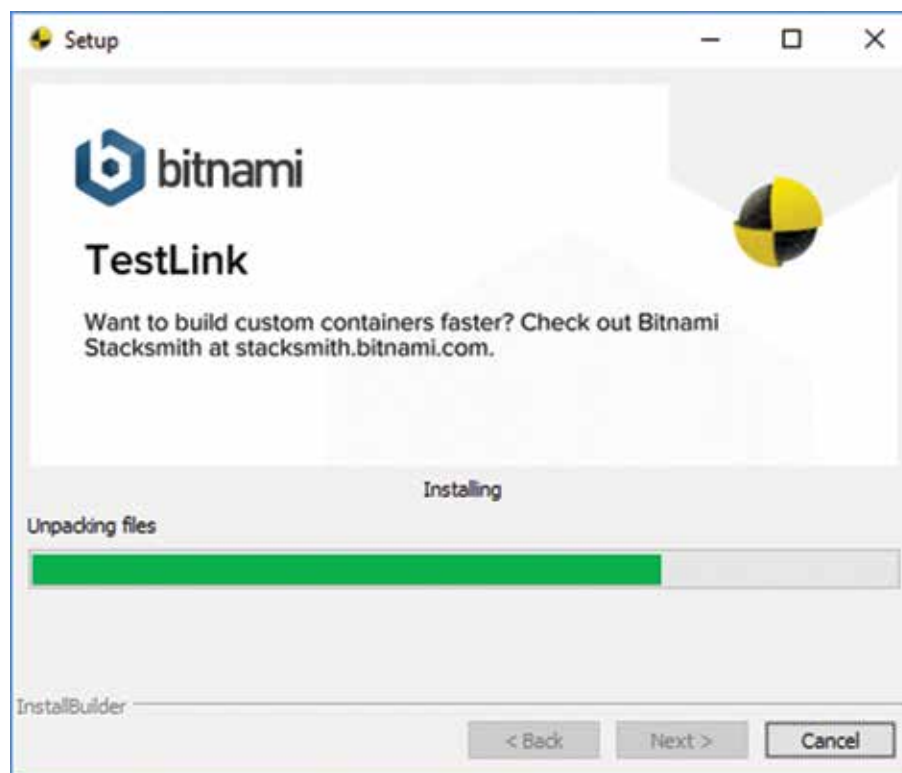
Eclipse IDE (2019)

Figura 46 - Tela com opção de instalação da ferramenta em nuvem
Fonte: TestLink (2019)



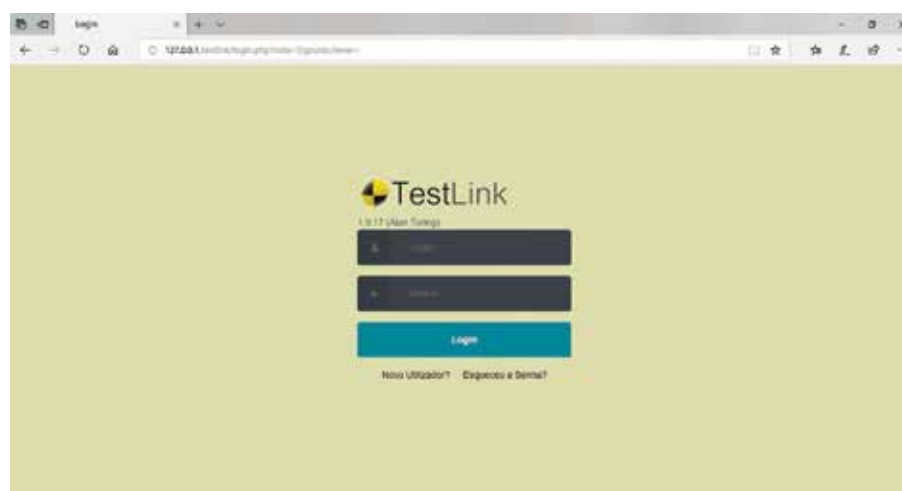
Eclipse IDE (2019)

Figura 47 - Tela para iniciar a execução da instalação
Fonte: TestLink (2019)



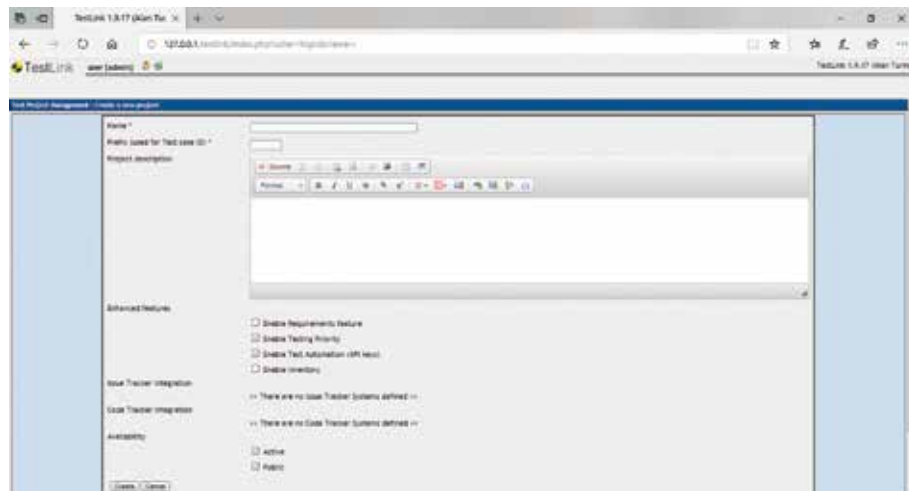
Eclipse IDE (2019)

Figura 48 - Tela para mostrar o progresso da instalação
Fonte: TestLink (2019)



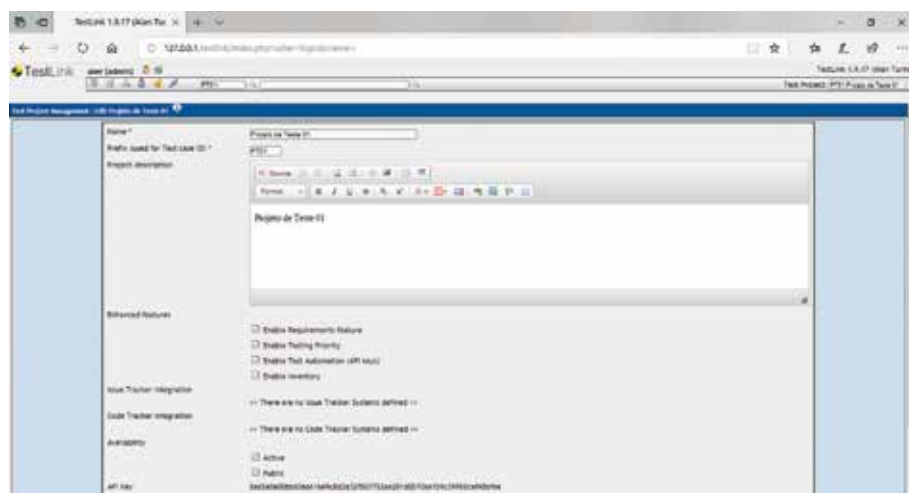
TestLink (2019)

Figura 49 - Tela inicial da ferramenta a partir de acesso local
Fonte: TestLink (2019)



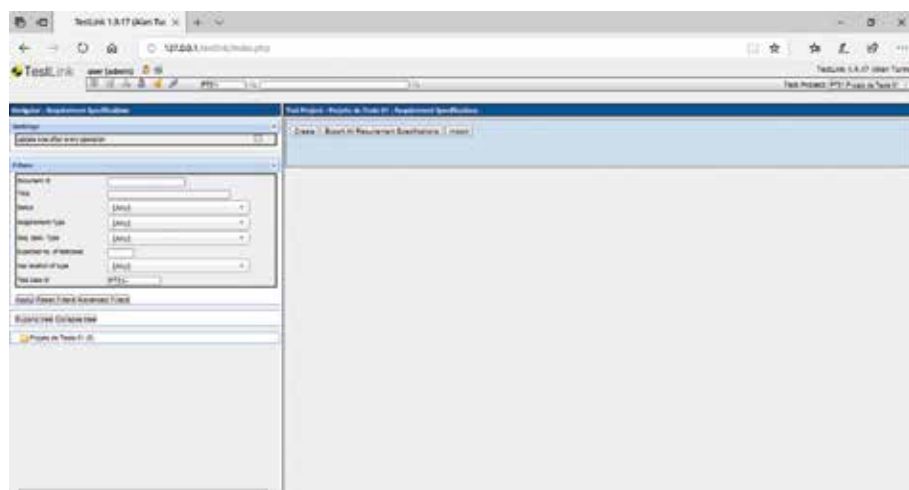
TestLink (2019)

Figura 50 - Tela da ferramenta para cadastro de projeto de teste
Fonte: TestLink (2019)



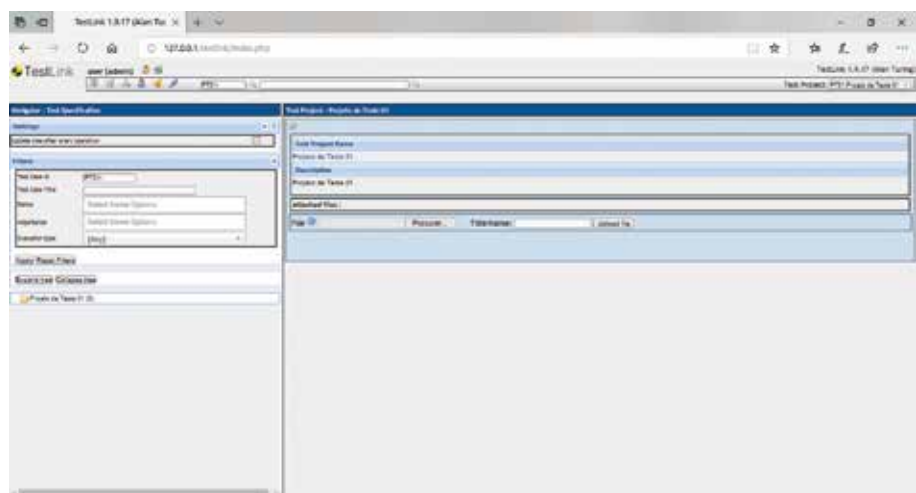
TestLink (2019)

Figura 51 - Tela da ferramenta com exemplo de projeto de teste
Fonte: TestLink (2019)



TestLink (2019)

Figura 52 - Tela da ferramenta para cadastro de requisito de teste
Fonte: TestLink (2019)



TestLink (2019)

Figura 53 - Tela da ferramenta para cadastro de caso de teste
Fonte: TestLink (2019)

O próximo tipo de teste automatizado está diretamente relacionado com as funções do sistema, ou seja, com os requisitos funcionais que determinam seu comportamento.

Acompanhe e continue aprendendo sobre automação de testes!



**SAIBA
MAIS**

O BSTQB (Brazilian Software Testing Qualification Board), além de certificações para analista de teste, também disponibiliza a certificação para engenheiro de automação de teste. Visite o site, a seguir, e confira mais detalhes a respeito!

<<https://www.bstqb.org.br/sobre-ctal-tae>>.

4.4 AUTOMAÇÃO DE TESTES FUNCIONAIS

A automação de testes funcionais tem o compromisso de automatizar os casos de testes que buscam a validação das principais funcionalidades do sistema testado.

As ferramentas de automação de teste neste contexto irão produzir scripts de testes que irão simular as ações de teste, nas quais o funcionamento interno da ferramenta não importa, sendo relevante o resultado produzido, ou seja, os scripts, que terão grande importância.

As principais vantagens deste tipo de automação consistem na aplicação de casos de teste que devem ser repetidos muitas vezes, execução de testes de regressão com poucos recursos e, conseqüente, aumento de produtividade da equipe de testes.

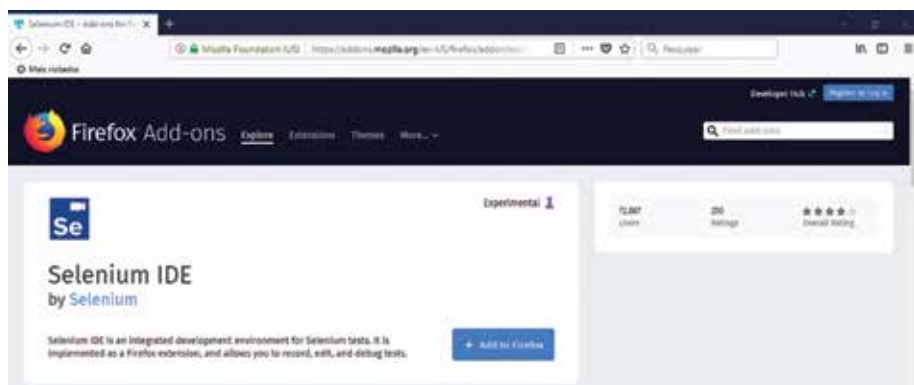
Para realizar a implantação desta automação com sucesso é importante seguir algumas práticas que você verá a seguir.

- a) Fornecer treinamento sobre a ferramenta de automação adotada.
- b) Definir os responsáveis das atividades inerentes à ferramenta.

- c) Estruturar uma equipe para a manutenção e uso da ferramenta.
- d) Definir o repositório de scripts de teste.
- e) Definir o processo de versionamento dos scripts de teste.
- f) Aplicar a implantação inicialmente em um projeto piloto.
- g) Produzir scripts pequenos, para facilitar a análise sobre estes.
- h) Realizar análise sobre os resultados obtidos no teste automatizado.
- i) Manter o estudo e acompanhamento de evoluções da ferramenta de automação adotada.

Neste tópico você vai ver uma aplicação prática da automação de teste funcional com a ferramenta Selenium IDE. Essa aplicação consiste em um plugin a ser instalado no navegador Firefox, que irá gravar as ações de teste sobre um sistema Web, e tais ações serão salvas em um script que pode ser reproduzido posteriormente.

Esta demonstração será iniciada com a instalação da ferramenta, que pode ser realizada por meio do acesso ao link: <<https://addons.mozilla.org/en-US/firefox/addon/selenium-ide/>> com o navegador Mozilla Firefox.



Selenium IDE (2019)

Figura 54 - Tela de instalação do Selenium IDE
Fonte: Selenium IDE (2019)



Selenium IDE (2019)

Figura 55 - Tela de administração de extensões do Firefox
Fonte: Selenium IDE (2019)

Após a instalação, será possível observar o ícone do Selenium IDE no canto superior direito do navegador, onde a ferramenta pode ser invocada, como mostrado a seguir.

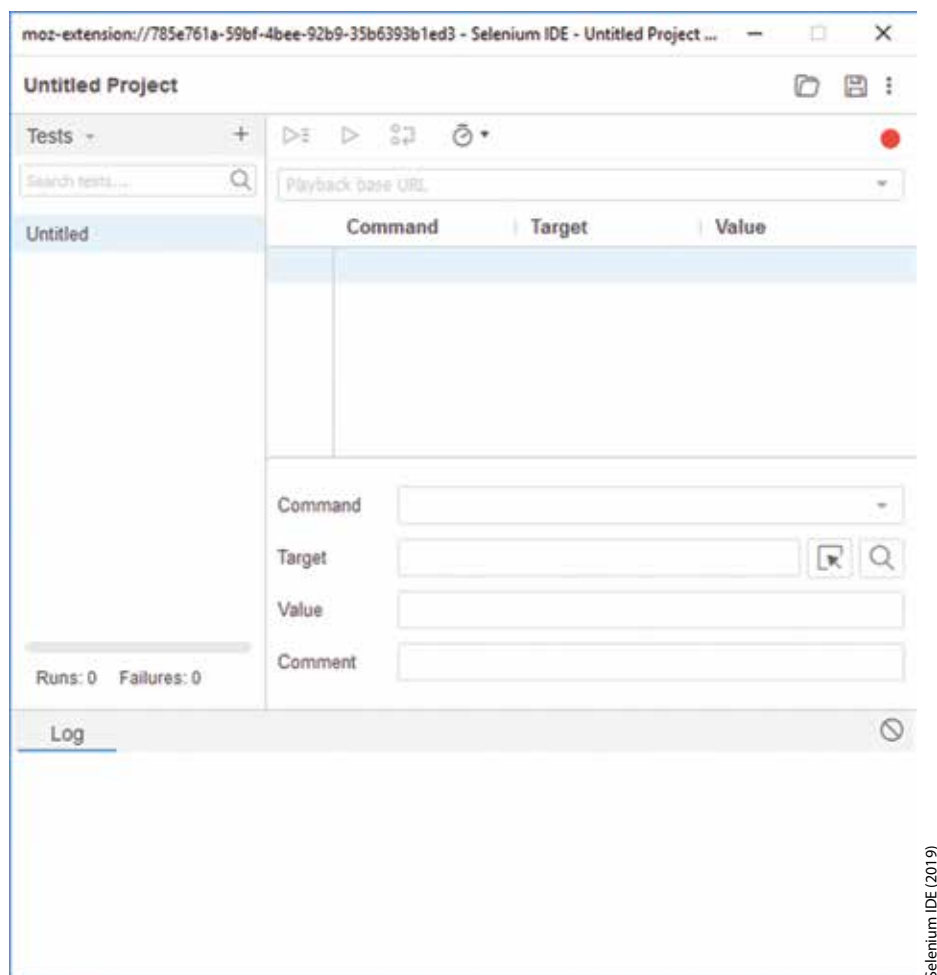
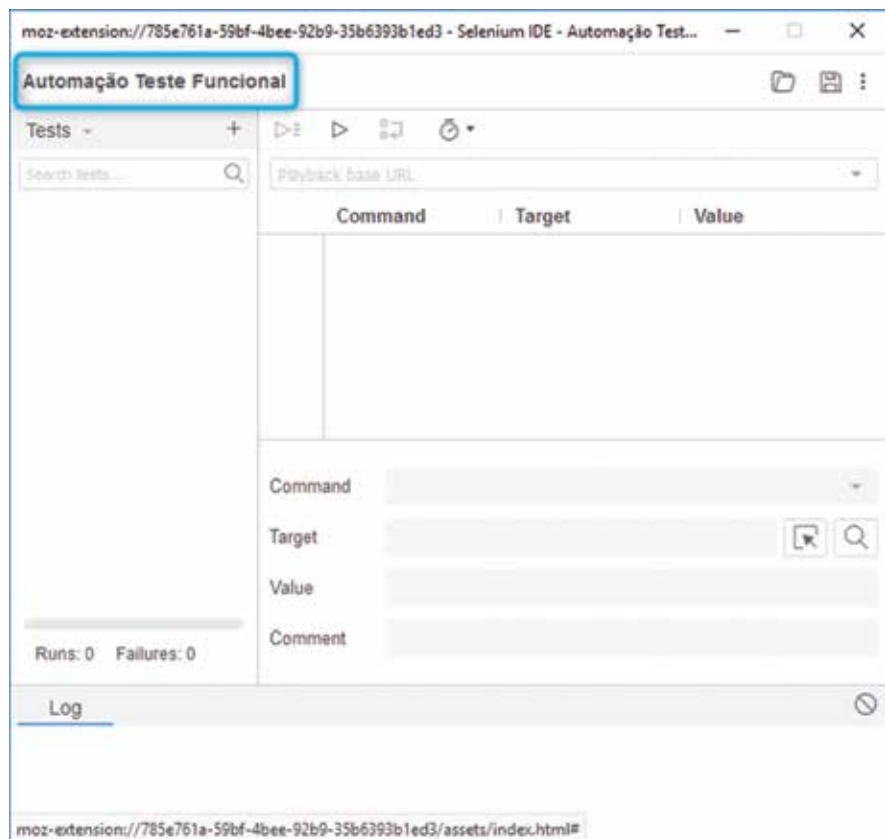


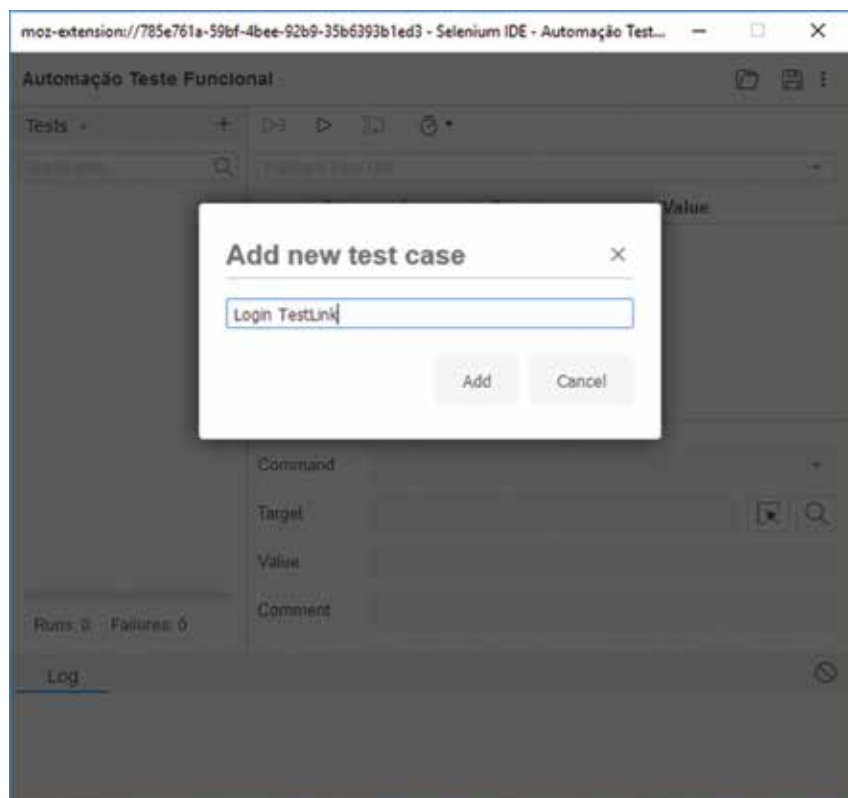
Figura 56 - Tela inicial do Selenium IDE
Fonte: Selenium IDE (2019)

Após esta execução inicial, será possível cadastrar o projeto de teste e adicionar os casos de teste ao projeto, como ilustrado na próxima figura.



Selenium IDE (2019)

Figura 57 - Tela inicial do Selenium com a definição do projeto de teste
Fonte: Selenium IDE (2019)



Selenium IDE (2019)

Figura 58 - Tela do Selenium para cadastro de novo caso de teste
Fonte: Selenium IDE (2019)

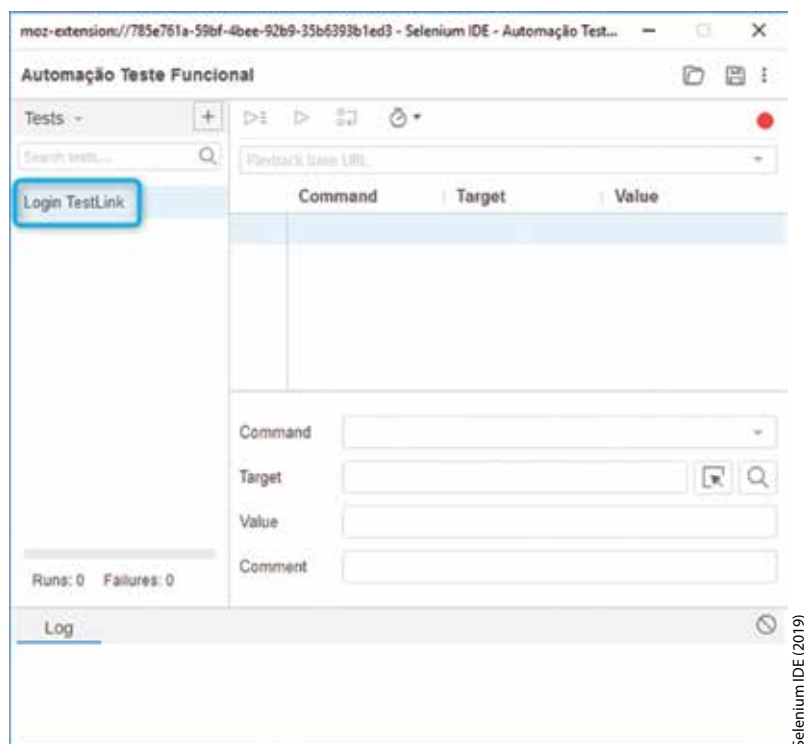


Figura 59 - Tela inicial do Selenium com a definição do caso de teste
Fonte: Selenium IDE (2019)

Após as ações iniciais, é necessário informar o endereço web do sistema que será testado, e iniciar a gravação do caso de teste, com as ações realizadas pelo testador, como pode ser observado a seguir.

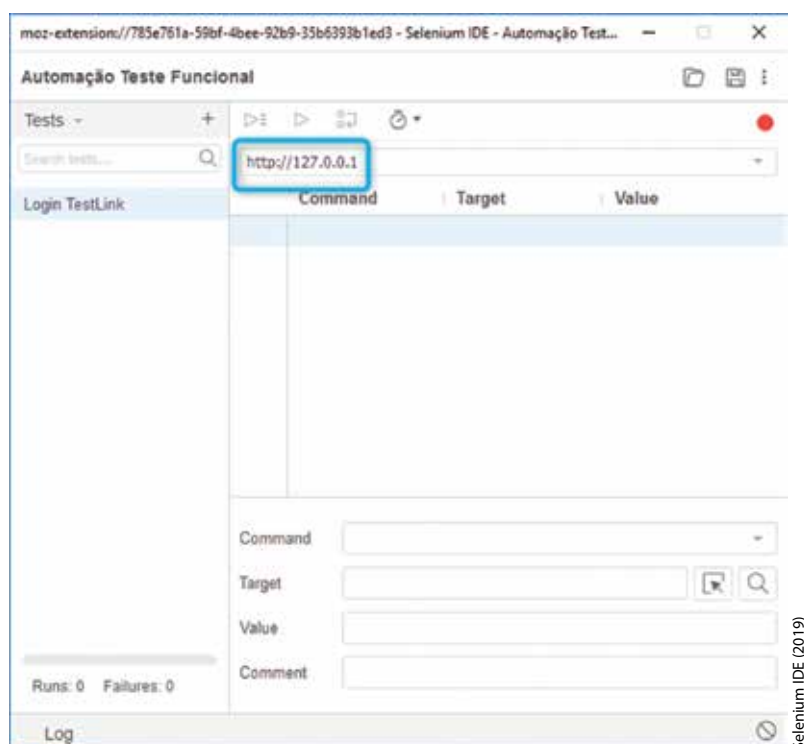
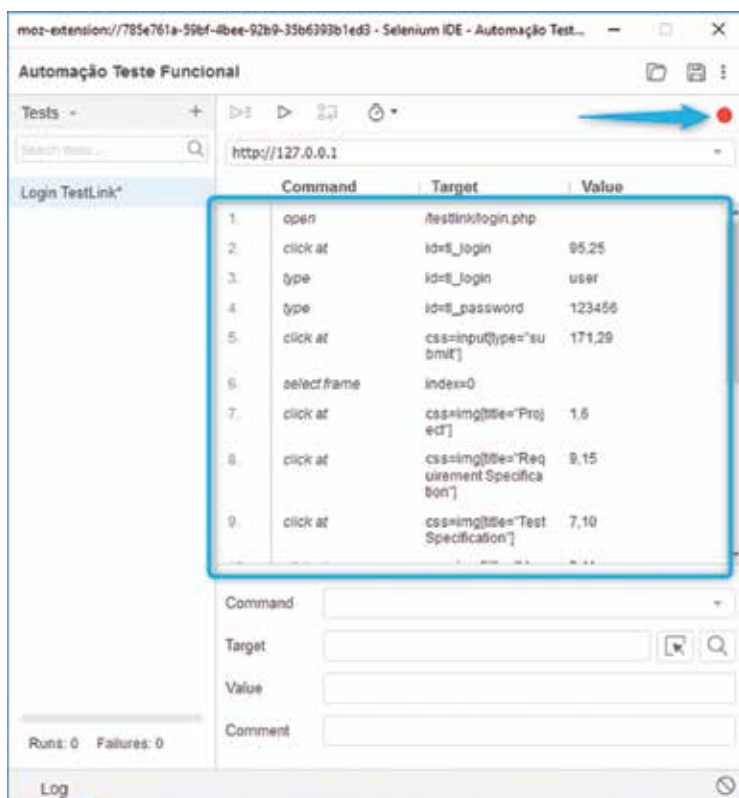
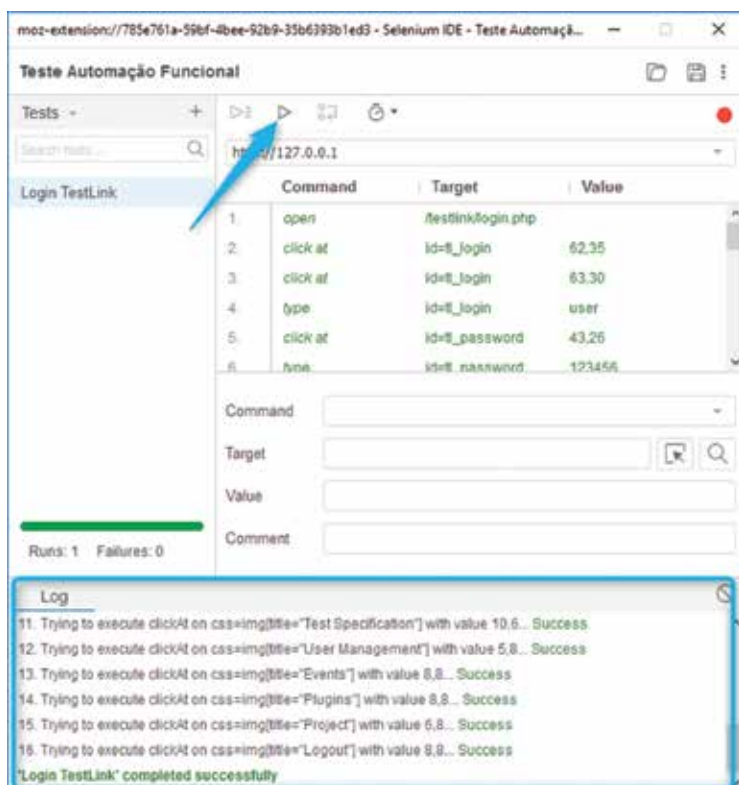


Figura 60 - Definição do endereço do sistema que será testado
Fonte: Selenium IDE (2019)



Selenium IDE (2019)

Figura 61 - Script gerado após gravação do teste com as ações do testador
Fonte: Selenium IDE (2019)



Selenium IDE (2019)

Figura 62 - Resultado da execução do Script de teste
Fonte: Selenium IDE (2019)

Nesta demonstração foi validado o login na ferramenta TestLink, apresentada no tópico anterior, no qual, após a gravação da ação de realizar o login e gerar o script de teste, foi executado pelo Selenium IDE, repetindo todos os passos de login e reportando o resultado do teste, como visto na figura anterior.

Chegou a hora de conhecer a automação de testes de performance, muito útil pelas características deste tipo de teste. Vamos avançar para a próxima etapa.

4.5 AUTOMAÇÃO DE TESTES DE PERFORMANCE

A automação de testes de performance é praticamente inerente a este tipo de teste, ou seja, realizar testes de performance de forma manual normalmente é muito difícil ou altamente custoso. Sendo assim, a automação nesse cenário é extremamente importante.

Assim como na automação de teste funcional, no teste de performance automatizado, existe uma gravação das ações do usuário. Porém, o foco dessa gravação são os comandos enviados entre a aplicação e o servidor desta aplicação, onde no contexto Web são essencialmente requisições do protocolo HTTP (HyperText Transfer Protocol).

Neste perfil de automação são utilizados os conceitos VU (Virtual User), para emular diversos usuários virtuais encaminhando e recebendo requisições HTTP, e Throughput, que significa a vazão ou tráfego destas requisições HTTP trocadas entre a aplicação e seu servidor. Com esses conceitos serão medidos os limites da aplicação para a quantidade máxima de acessos simultâneos e o tempo de resposta das funcionalidades do sistema testado.

O teste de performance ainda pode ser classificado em uma granularidade menor, como segue:

Teste de Carga

Tem foco na quantidade máxima de acessos simultâneos no sistema e o seu comportamento com esta carga de usuários.

Teste de Desempenho

Possui o interesse em identificar o tempo de resposta das funcionalidades do sistema, independentemente da quantidade de usuários.

Teste de Estresse

O objetivo é identificar o limite de carga sobre o sistema, sem degradação do seu desempenho.

D'Imite Camargo Martins (2019)

Figura 63 - Teste de performance em granularidade menor
Fonte: Do autor (2019)

As ferramentas de automação para testes de performance possuem a capacidade de contemplar todas as faces deste tipo de teste, além de possibilitar o monitoramento de outras características de consumo de recursos, como o uso de memória alocada, informações sobre as transações do banco de dados e demais recursos do ambiente em que o teste é realizado.

As principais vantagens sobre este tipo de automação estão relacionadas à identificação do comportamento do sistema em cenários críticos de utilização, definição, com maior embasamento, da infraestrutura necessária para a execução do sistema e descoberta de possíveis gargalos de desempenho antes de desmobilizar o sistema em ambiente de produção.

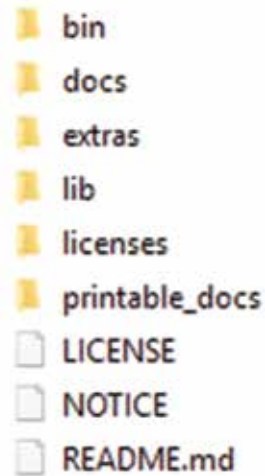
Para possibilitar a implantação e aplicação desta automação é importante observar algumas práticas. São elas:

- a) Identificar as necessidades de testes, de acordo com as especificações não funcionais sobre desempenho do sistema.
- b) Definir os cenários de teste com as especificações dos testes que serão aplicados, como a quantidade de VU e entre outras.
- c) Garantir que o sistema testado esteja disponível e com acesso para o monitoramento.
- d) Usar computadores com recursos suficientes para não limitar os testes realizados.
- e) Evitar processamento desnecessário nos computadores usados para aplicação dos testes, garantindo que os recursos estarão destinados para os testes.
- f) Realizar o monitoramento dos testes durante a sua execução.
- g) Analisar os resultados obtidos com as especificações de desempenho do sistema.

Neste tópico você também vai ver uma aplicação prática da automação de teste de performance com a ferramenta JMeter, que consiste na gravação das ações de teste, essencialmente as requisições HTTP sobre um sistema Web, e estas ações serão armazenadas em um script que poderá ser reproduzido posteriormente.

Esta demonstração será iniciada com a execução da ferramenta, que pode ser obtida por meio do link <https://jmeter.apache.org/download_jmeter.cgi>.

Após concluir o download, será necessário descompactar o pacote baixado e acessá-lo, como é demonstrado na figura a seguir.



Apache JMeter (2019)

Figura 64 - Estrutura de arquivo do JMeter
Fonte: Apache JMeter (2019)

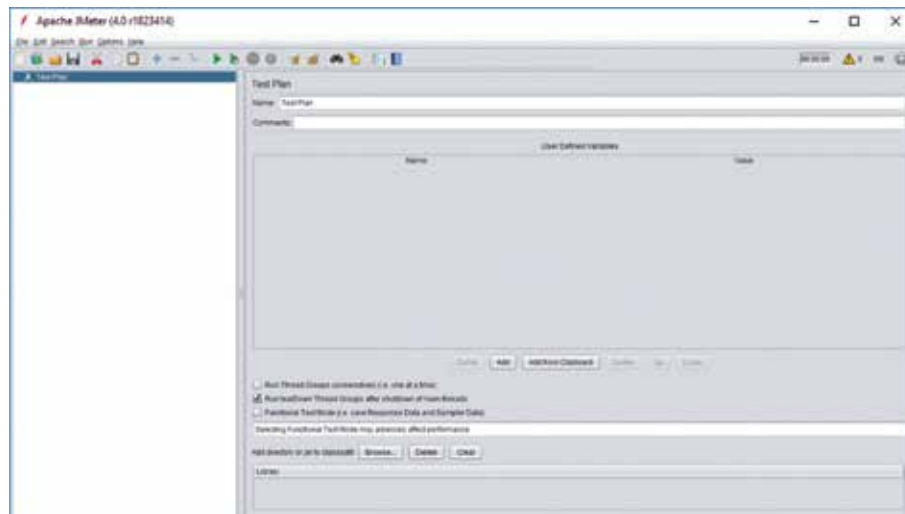
o diretório “bin” haverá o arquivo de execução para SO Windows, identificado como “jmeter.bat”, e, ao executá-lo com um duplo clique do mouse, deverão ser exibidas as telas ilustradas nas figuras que seguem.

A imagem mostra a interface de linha de comando (CLI) do JMeter em uma janela de terminal. O título da janela é 'C:\WINDOWS\system32\cmd.exe'. O conteúdo da tela é o seguinte:

```
Don't use GUI mode for load testing !, only for Test creation and Test debugging.  
For load testing, use NON GUI Mode:  
jmeter -n -t [jmx file] -l [results file] -e -o [Path to web report folder]  
& increase Java Heap to meet your test requirements:  
Modify current env variable HEAP="-Xms1g -Xmx1g -XX:MaxMetaspaceSize=256m" in the  
jmeter batch file  
Check : https://jmeter.apache.org/usermanual/best-practices.html  
=====
```

Apache JMeter (2019)

Figura 65 - Tela com interface CLI do JMeter
Fonte: Apache JMeter (2019)

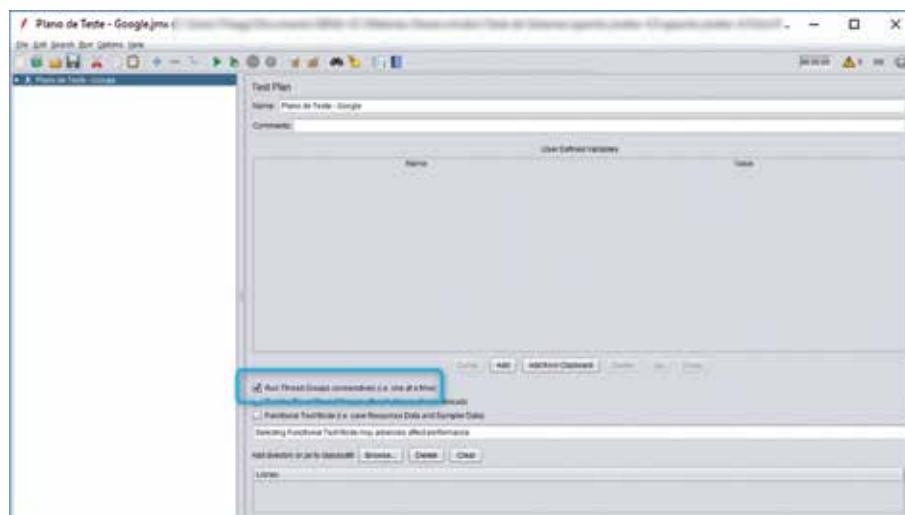


Apache JMeter (2019)

Figura 66 - Tela com interface gráfica do JMeter
Fonte: Apache JMeter (2019)

Após a execução do JMeter, a demonstração será iniciada com o objetivo de testar o acesso ao Google com vinte usuários simultâneos e medir o tempo de resposta do acesso a este site.

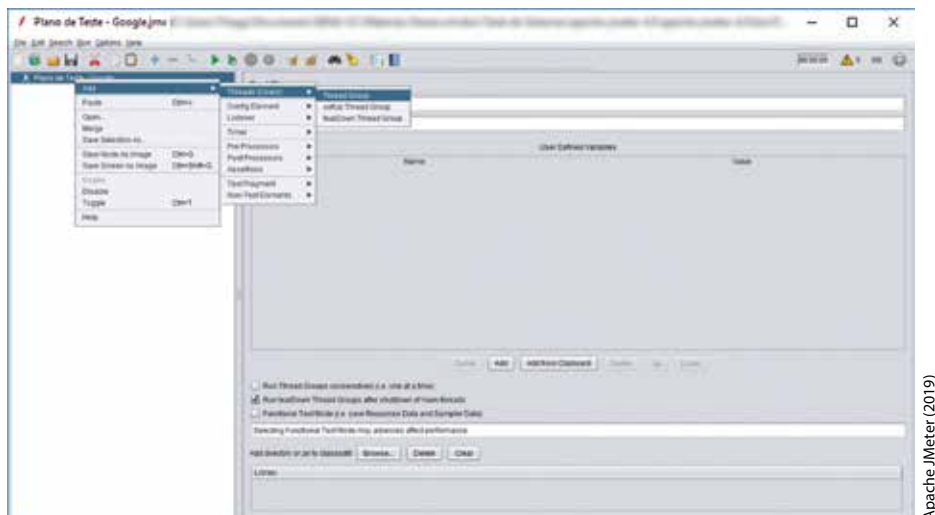
O primeiro passo é definir um plano de teste na ferramenta, como pode ser observado nas figuras seguintes.



Apache JMeter (2019)

Figura 67 - Definição do plano de teste
Fonte: Apache JMeter (2019)

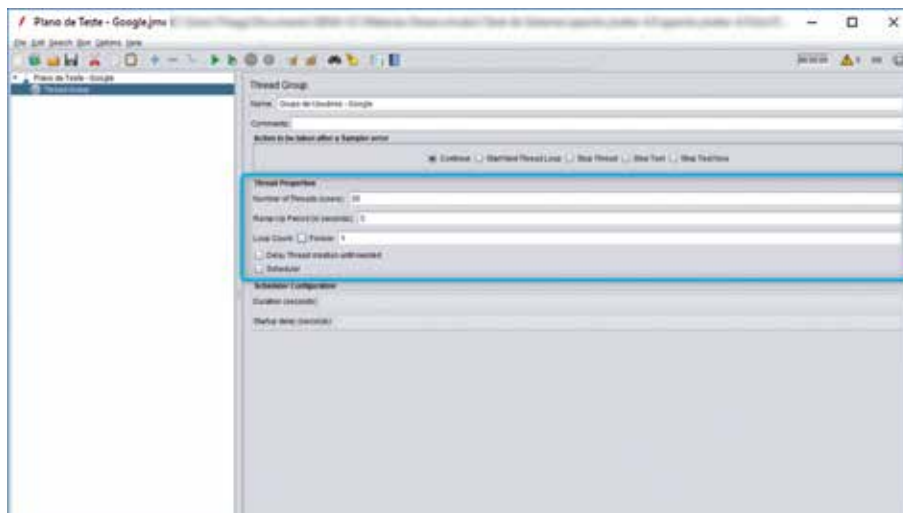
Na figura anterior, você pôde ver que foi renomeado o plano de teste e selecionada sua forma de execução, conforme o destaque da figura.



Apache JMeter (2019)

Figura 68 - Cadastro de grupo de usuários
Fonte: Apache JMeter (2019)

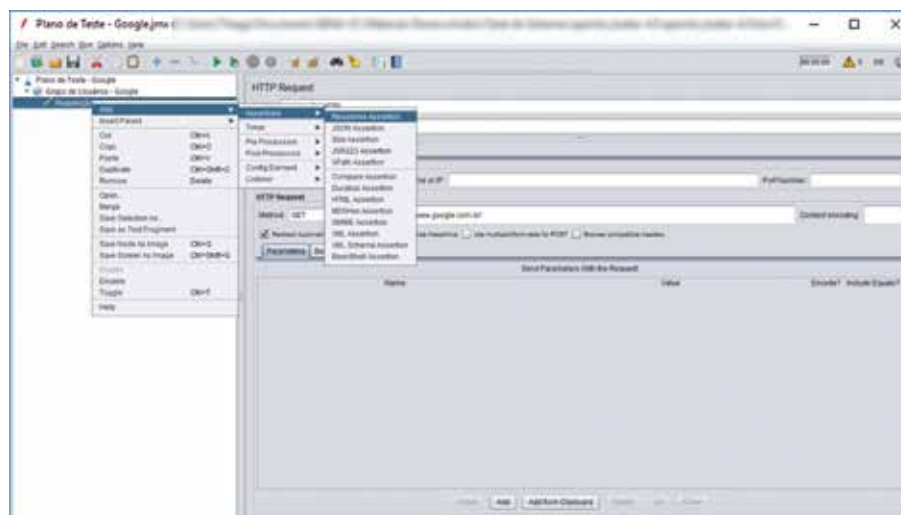
Na sequência, foi criado o grupo de usuários que irá simular os acessos simultâneos no site do Google, como é demonstrado na figura anterior.



Apache JMeter (2019)

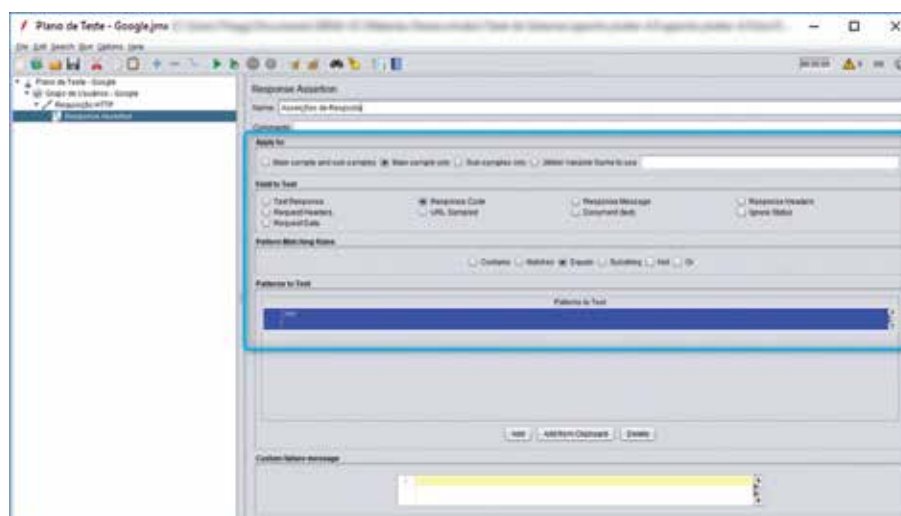
Figura 69 - Configuração do grupo de usuários
Fonte: Apache JMeter (2019)

Para verificar se o acesso dos usuários simulados ao site do Google foi realizado com sucesso, é criada uma verificação ou asserção deste teste, como mostrado nas figuras que seguem.



Apache JMeter (2019)

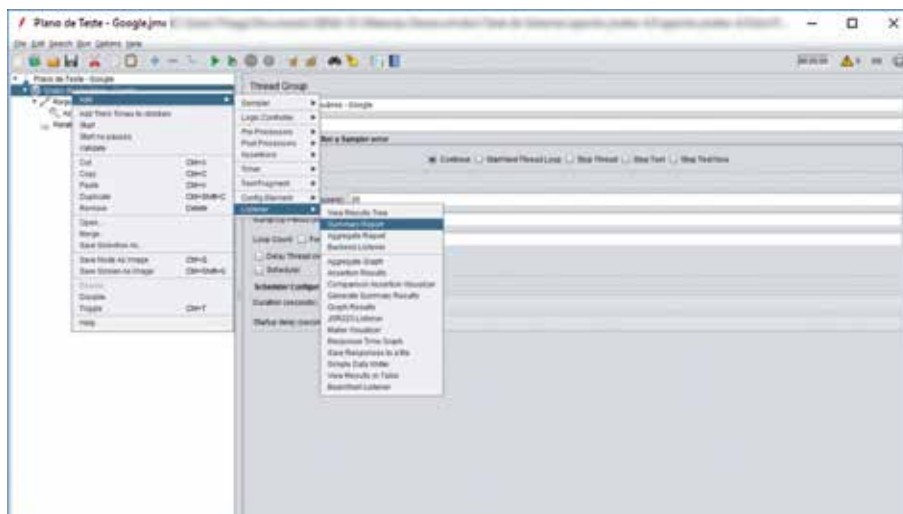
Figura 72 - Cadastro de asserção de resposta
Fonte: Apache JMeter (2019)



Apache JMeter (2019)

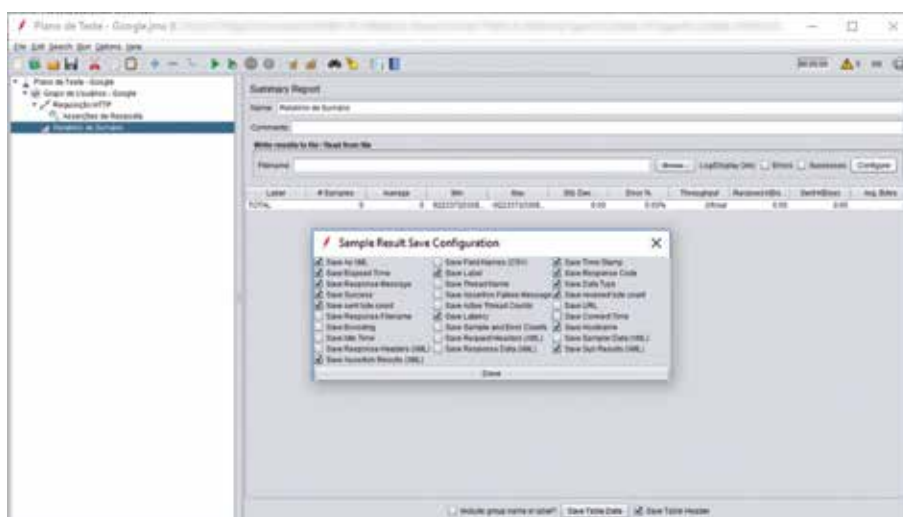
Figura 73 - Configuração da asserção de resposta
Fonte: Apache JMeter (2019)

Assim que as configurações da realização do teste forem concluídas, é preciso configurar a coleta de dados e o monitoramento deste teste por meio de relatórios, ilustrados nas próximas figuras.



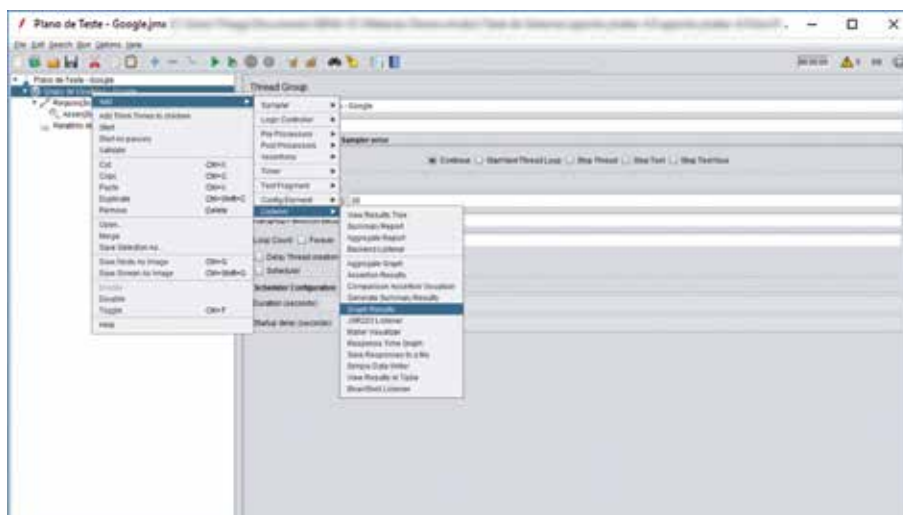
Apache JMeter (2019)

Figura 74 - Criação do relatório de sumário
Fonte: Apache JMeter (2019)



Apache JMeter (2019)

Figura 75 - Configuração do relatório de sumário
Fonte: Apache JMeter (2019)



Apache JMeter (2019)

Figura 76 - Criação do gráfico de resultados
Fonte: Apache JMeter (2019)

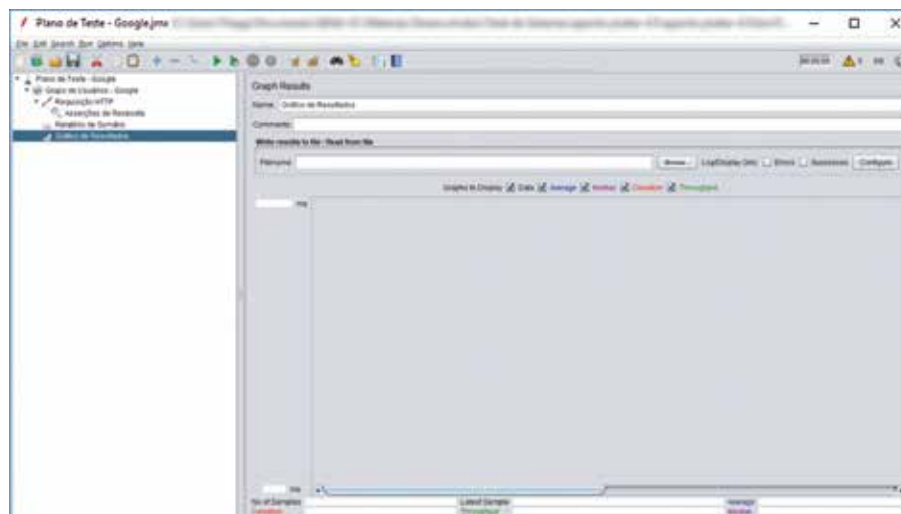


Figura 77 - Gráfico de resultados
Fonte: Apache JMeter (2019)

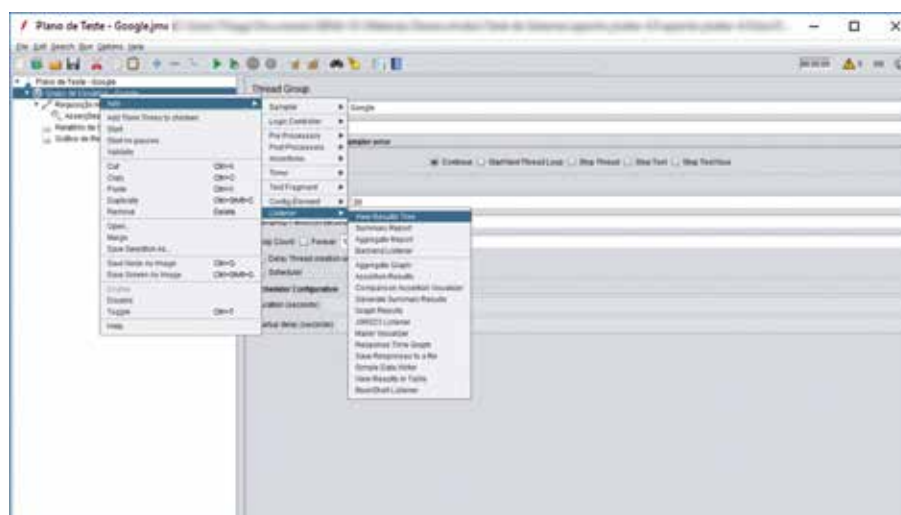


Figura 78 - Criação da árvore de resultados
Fonte: Apache JMeter (2019)

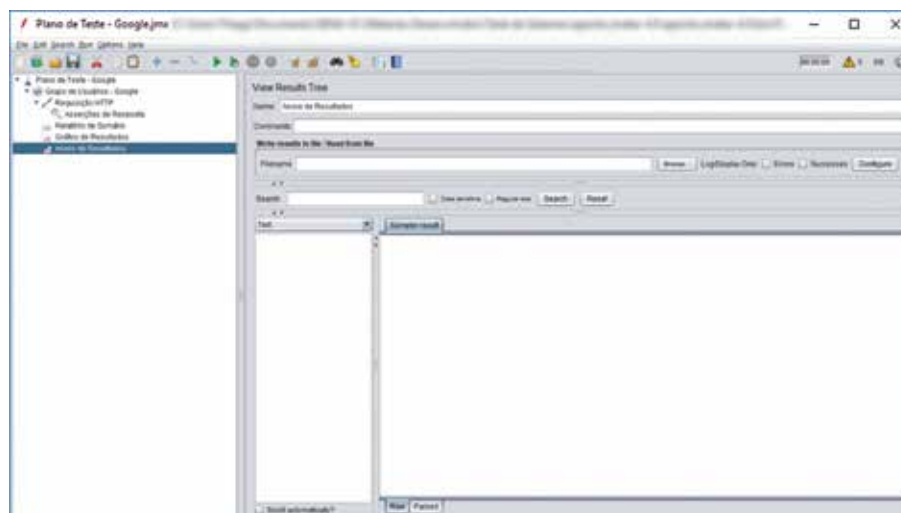


Figura 79 - Árvore de resultados
Fonte: Apache JMeter (2019)

Neste ponto da demonstração, todas as configurações foram concluídas. Sendo assim, o teste pode ser executado clicando no botão de “start” da ferramenta, como ilustrado a seguir.



Figura 80 - Execução do teste no JMeter
Fonte: Apache JMeter (2019)

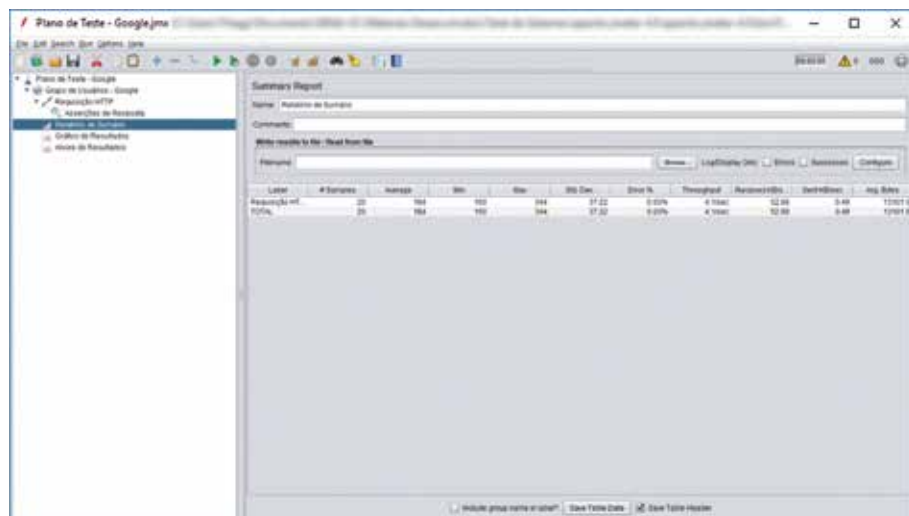


Figura 81 - Resultado do teste no relatório de sumário
Fonte: Apache JMeter (2019)

Assim que a execução do teste é concluída, é possível observar os resultados obtidos, analisando os relatórios configurados no plano de teste, de acordo com as figuras que seguem.

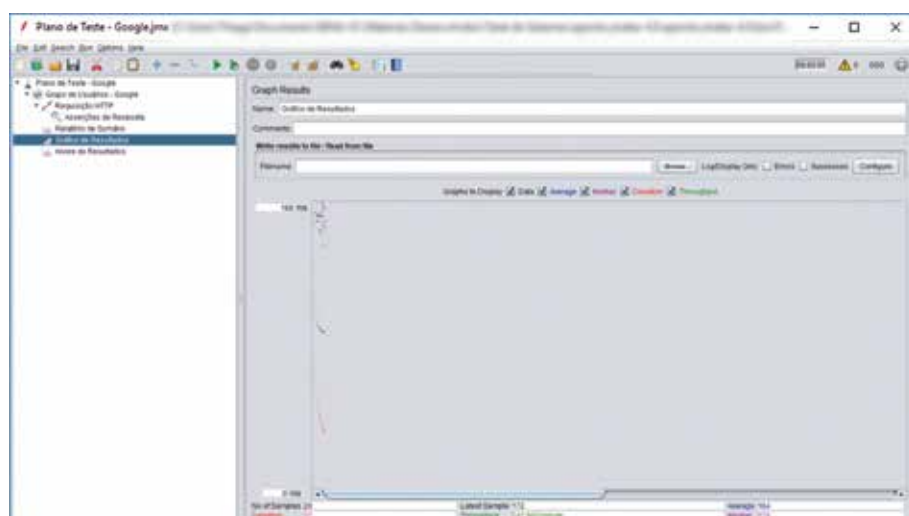
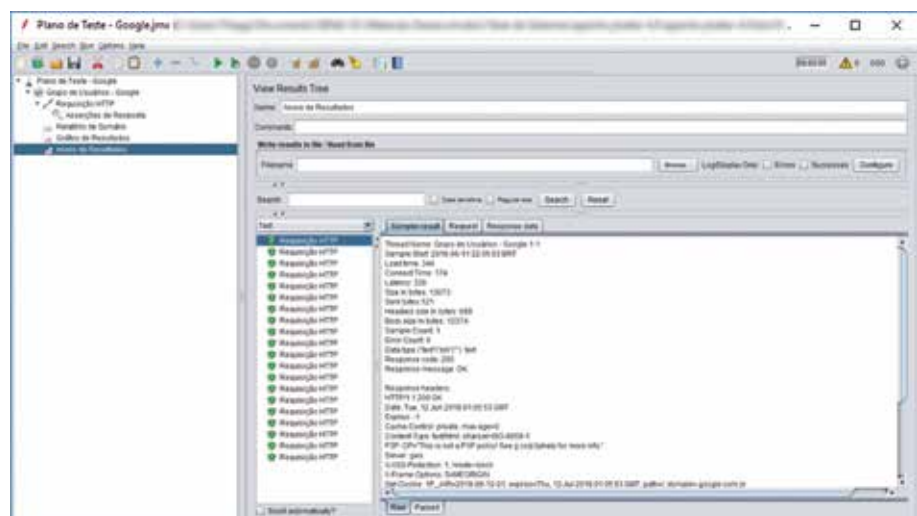


Figura 82 - Resultado do teste no gráfico de resultados
Fonte: Apache JMeter (2019)



Apache JMeter (2019)

Figura 83 - Resultado do teste na árvore de resultados
Fonte: Apache JMeter (2019)

Este tópico foi concluído com a abordagem e demonstrações das principais formas de automação de testes e das ferramentas mais utilizadas para essas automações, revelando-se uma referência interessante para a aplicação prática das atividades de teste.

No próximo capítulo você verá os registros e análises dos resultados de teste, para que o ciclo de teste possa ser concluído com resultados estruturados que irão contribuir para a qualidade dos sistemas aferidos. Pronto para continuar? Vamos lá!



CASOS E RELATOS

Automação de teste e alto desempenho

No desenvolvimento de software, realizado em uma renomada empresa da região, o processo de desenvolvimento e de teste eram estruturados e integrados, porém os projetos eram geralmente entregues com atraso, gerando insatisfação dos clientes e aumento de custos. Após a análise da gestão desta empresa, foi identificado que a realização de testes era o motivo destes atrasos, no entanto, era notória a importância dos testes para conferir qualidade aos sistemas. Desse modo foi adotado a automação dos testes que eram executados com maior frequência e que exigiam a produção de massa de dados. Nos projetos realizados após esta automação foi percebido grande aumento no desempenho da execução de testes e em poucos projetos o custo de implantação da automação foi compensado pela eficiência e qualidade dos sistemas produzidos.



RECAPITULANDO

Neste capítulo, você viu que a automação do processo de teste vai aumentar a eficiência e diminuir o tempo na hora de realizar os testes. Você também estudou os conceitos de automação de testes, os tipos de testes automatizados, a automação de testes unitários, que consiste em realizar testes em pequenas partes do sistema, além de identificar como é realizada a automação de gerência de planejamento de testes, dos testes funcionais e de performance, observando a importância da automação desses testes.



Neste capítulo você irá conhecer os seguintes tipos de relatório de testes: Relatório de Log de Teste, Relatório de Incidente de Teste e Relatório de Sumário de Teste, assim como seus diferentes padrões e estruturas. Sendo assim ao final dos estudos você terá subsídios para:

- a) documentar os resultados de testes em seus respectivos formatos de relatórios;
- b) compreender os relatórios de testes previstos na Norma 829 do IEEE (Institute of Electrical and Electronics Engineers).

Preparado para iniciar este último capítulo? Vamos lá!

5.1 TIPOS DE RELATÓRIOS DE TESTES

Os resultados obtidos no decorrer da execução dos testes, ou seja, durante todo o processo de teste, devem ser documentado para que estejam estruturados e acessíveis para diversas finalidades do projeto de teste, como o monitoramento dos testes, defeitos que precisam ser corrigidos, testes de regressão necessários, entre outros.



YakobchukOlena(2017)

A maneira mais adequada, e prevista em normas e boas práticas, é o registro destes resultados por meio de relatórios específicos de teste. Tais documentos estão presentes em diferentes momentos do processo de teste.

O instituto IEEE (Institute of Electrical and Electronics Engineers) estabelece a norma 829, que sugere três tipos de relatórios de testes específicos, sendo estes:

- a) Relatório de Log de Teste.
- b) Relatório de Incidente de Teste.
- c) Relatório de Sumário de Teste.

Em algumas literaturas também é definido o Relatório de Defeito, além dos citados anteriormente pela norma. No entanto, é possível e até usual que este relatório esteja suprimido no Relatório de Incidente de Teste.



CURIOSIDADES

Você sabia que o IEEE foi criado em 1963 com o objetivo de disseminar conhecimento sobre engenharia elétrica, computação e telecomunicação, mas que atualmente sua atuação compreende a robótica, bioengenharia, nanotecnologia, entre outras áreas voltadas para a tecnologia.

No tópico seguinte, serão abordados os relatórios previstos pela norma 829. Porém, para aplicações do teste em que o Relatório de Defeitos seja necessário, é importante apresentar a estrutura deste documento, como segue:

- a) Número do relatório.
- b) Código do projeto de desenvolvimento.
- c) Nome do projeto de desenvolvimento.
- d) Código do projeto de teste.
- e) Nome do projeto de teste.
- f) Código do requisito.
- g) Nome do requisito.
- h) Identificação da etapa de teste.
- i) Código do caso de teste.
- j) Nome do caso de teste.
- k) Código do script de teste (se houver automação).
- l) Nome do script de teste (se houver automação).

Com essa disposição, é possível identificar e estruturar a informação sobre o defeito encontrado no teste, ressaltando o requisito afetado e o momento e condições do teste.

Os relatórios de teste descritos aqui são documentos estruturados, que possuem uma organização pré-definidas das informações, com o intuito de tornar os resultados mais claros, objetivos e precisos.

Para identificar o analista e as atividades de teste realizadas por este, também há um padrão definido, como você vai ver descrito a seguir.

- a) Nome do analista.
- b) Tipo de teste.
- c) Data e hora da identificação do defeito.
- d) Evidências do defeito.
- e) Descrição do defeito.
- f) Severidade do defeito (Alta, Média ou Baixa).
- g) Tipo de defeito (Indefinido, Defeito de requisito, Defeito Funcional).

Veja, a partir de agora, os relatórios propriamente ditos, observando o objetivo e a estrutura de cada um.

Relatório de Log de Teste

Neste relatório são reportadas as principais ou mais relevantes atividades ocorridas durante todo o processo de teste. Assim, ele pode abranger todas as atividades de testes envolvidas no projeto.

**FIQUE ALERTA**

O reporte de forma clara e consistente dos resultados do teste é essencial para o sucesso do projeto de teste.

O formato deste documento está previsto na estrutura descrita a seguir. Acompanhe!

Identificação Identificador ou código único para o item do relatório.
Descrição Descrever o teste e ambiente de execução deste teste.
Atividades e eventos Registro dos momentos em que as atividades ocorreram.
Descrição da execução Descrever a execução do teste e identificar o analista responsável.
Resultados Descrever os resultados do teste. Ambiente de teste: Informações complementares do ambiente de teste.
Eventos imprevistos Descrever possíveis imprevistos na execução do teste.

D'Imire Camargo Martins (2019)

Figura 84 - Estrutura do relatório de log de teste
Fonte: Do autor (2019)

Relatório de Incidente de Teste

O principal objetivo deste relatório é demonstrar a necessidade de maiores verificações em incidentes encontrados nos testes. Como mencionado anteriormente, a partir dele pode ser derivado o relatório de defeitos, que deverá realizar uma análise mais aprofundada sobre o incidente detectado no teste.

**SAIBA MAIS**

O Syllabus do ISTQB aborda os principais conceitos para a realização de testes. Visite o site, a seguir, e faça download do documento, conferindo mais detalhes a respeito!
<https://www.bstqb.org.br/uploads/syllabus/syllabus_ctfl_Eclipse_8br.pdf>.

A estrutura é simples, com maior detalhamento para a descrição do incidente, de acordo com o formato apresentado:

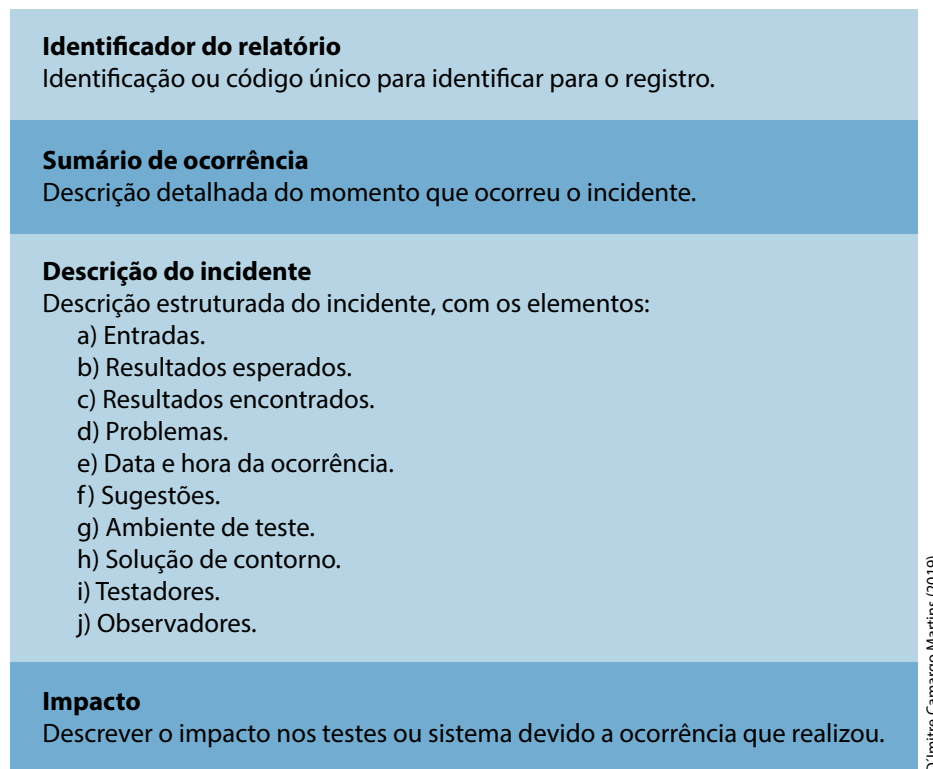


Figura 85 - Estrutura do relatório de incidente de teste
Fonte: Do autor (2019)

Relatório de Sumário de Teste

A intenção deste relatório é disponibilizar um sumário com as atividades de testes realizadas em um projeto de teste e correlacionar estas atividades com uma breve descrição das ocorrências identificadas.

Do mesmo modo que os relatórios anteriores, este também possui uma estrutura pré-definida:

- a) Identificador:** Identificação ou código único para o relatório.
- b) Sumário:** Descrição resumida das atividades de teste realizadas no projeto de teste.
- c) Variações:** Relacionar procedimentos adotados que não estavam previstos no planejamento inicial do projeto de teste.
- d) Avaliações do processo:** Descrição de eventuais ocorrências que possam ter impactado no processo de teste e na qualidade do sistema.
- e) Sumário dos resultados:** Relacionar as métricas do projeto de teste, como a quantidade de casos de teste, incidentes e defeitos encontrados, entre outros.
- f) Avaliação do teste:** Descrição sobre a avaliação do projeto de teste e de possíveis riscos não mitigados no projeto.
- g) Sumário das atividades:** Informação dos recursos humanos consumidos e relacionados com o tempo investido em atividades de teste e no projeto de teste.
- h) Aprovações:** Identificação das pessoas responsáveis pela aprovação do projeto de teste.

Este relatório também representa o final do projeto de teste e da conclusão das atividades relacionadas aos testes do sistema. Assim, você já tem os conhecimentos essenciais para ser um futuro analista de testes. Portanto, lembre-se de prezar pela qualidade dos sistemas testados!



CASOS E RELATOS

A comunicação do teste

A comunicação é fundamental em todas as relações profissionais, sendo assim, na atividade de testes de sistemas sua importância também é expressiva, principalmente na divulgação dos resultados obtidos com a aplicação dos testes.

José, que atua como analista de teste, reconhece essa importância e aplica as devidas normas para estruturar e tornar mais clara essa comunicação, desse modo, garantindo que os gestores estejam cientes dos resultados, que os incidentes possam ser resolvidos, que os defeitos encontrados sejam corrigidos e que as demais providências e ações possam ser realizadas até a conclusão final dos testes, na qual também deverá ser reportada.

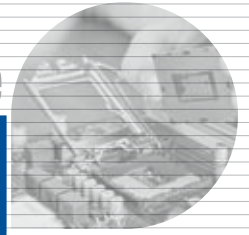


RECAPITULANDO

Finalizando seu estudo, você viu que os resultados obtidos durante a realização dos testes devem ser documentados nos relatórios de testes. Esses relatórios são documentos com informações estruturadas que visam deixar os resultados mais claros e objetivos. Você conheceu os relatórios previstos na norma 829 e os relatórios normatizados pelo instituto IEEE, verificando o padrão e a estrutura de cada um deles.

Virtudes Profissionais, Organização e Qualidade

6



Aqui você vai entender o que é profissionalismo e a sua importância para o ambiente de trabalho, irá reconhecer os princípios da organização aplicáveis aos ambientes de trabalho, vai entender como realizar a organização do seu tempo, dos seus compromissos, das suas atividades e do seu local de trabalho. Por fim você entenderá ainda o que é o sistema de gestão da qualidade e sua importância nas atividades profissionais. Sendo assim ao final deste capítulo você estará apto a:

- a) planejar e administrar seu tempo e as atividades sob sua responsabilidade;
- b) aplicar princípios da qualidade em suas atividades profissionais;
- c) adotar atitudes de respeito e compromisso na execução de suas atividades profissionais.

Preparado? Então é hora de iniciar o processo de construção de novos conhecimentos e saberes. Vamos lá!

6.1 VIRTUDES PROFISSIONAIS

Profissionalismo é a característica central dos profissionais que exercem suas atividades de acordo com as regras estabelecidas pela sociedade. Essas regras ou virtudes profissionais dizem respeito às atitudes morais e éticas e às características clássicas de respeito e compromisso que o profissional adota para conduzir a execução de suas atividades.

Você se considera um bom profissional? Que tal se aprofundar nesse conceito?

Após entender o conceito de profissionalismo, na sequência você estudará itens importantes para o seu dia a dia no trabalho. Acompanhe!

6.1.1 ORGANIZAÇÃO

A organização do trabalho atua precisamente para conseguir um objetivo que se define como qualidade. Organizar qualquer tipo de trabalho é tarefa para técnicos preparados e que tem como função determinar um nível qualificativo para qualquer tipo de tarefa.

A organização do trabalho é uma engrenagem perfeita para realizar todos os tipos de tarefas as quais você se propõe com uma realização final mais satisfatória. Se um trabalho é organizado, possivelmente todos os objetivos propostos por uma empresa serão cumpridos.

Confira a seguir como planejar e administrar seu tempo no trabalho.

6.1.2 PLANEJAMENTO E ADMINISTRAÇÃO DO TEMPO

Não se perder em uma rotina de trabalho intensa não é fácil. Com muitas demandas chegando, fica difícil não se perder e acabar deixando de lado atividades importantes. Para que isso não aconteça, é necessário criar uma organização de trabalho que lhe possibilite realizar todas as atividades sem se perder. Cada rotina exige uma organização própria, sendo necessário analisar o cenário específico e criar um processo que o atenda. Não existe uma fórmula padrão perfeita, existe aquela que melhor atende a sua necessidade.



Tatomm (2018)

É necessário observar os pontos que mais falham na rotina. Se tarefas importantes estão ficando de fora, é importante definir um horário do dia para se dedicar a elas. Se tarefas de rotina importantes estão ficando para trás, uma lista de tarefas (*checklist*) ou o uso de lembretes eletrônicos (como os do Google ou do *Outlook*) pode ser a solução. É importante que a organização não seja muito complicada, dificultando o processo de organização. O objetivo é que seja fácil de se fazer e administrar, sem consumir muito tempo e esforço.

Administre seu tempo, estabeleça horários e vença suas responsabilidades. Esteja sempre atento às suas atividades e seus compromissos, evite distrações, conversas paralelas e não se alongue nas conversas pelo telefone. Tenha em mente que a concentração é sua maior aliada.

Se você perde prazos com frequência, esquece compromissos e leva trabalho para casa, provavelmente não está fazendo uma boa administração de seu tempo. A seguir, algumas dicas importantes para se organizar e melhorar a produtividade.

- a) **Limpeza e organização:** Manter seu local de trabalho limpo e seus materiais de trabalho e documentos organizados (inclusive os de computador) vão ajudar muito na execução das tarefas. Encontrar facilmente as coisas vai poupar tempo.
- b) **Utilize a tecnologia:** Cadastre seus compromissos na agenda do celular e coloque alertas antecipados para que possa se planejar e não esquecer compromissos e atividades importantes.
- c) **Estimativa e priorização:** Calcule o tempo estimado para a realização de tarefas e de compromissos para conseguir planejar o que poderá ser feito durante o dia. Depois disso, organize as atividades por ordem de prioridade colocando o que for mais urgente primeiro.
- d) **Foco e compromisso:** Não perca o foco de suas atividades com distrações como redes sociais, vídeos, longos intervalos ou bate-papo com outros colegas. Práticas como essas podem comprometer o planejamento de um dia inteiro. Manter o foco e o comprometimento vai possibilitar o cumprimento das atividades e a adaptação a imprevistos.
- e) **Pausa para relaxar:** As pausas para descanso devem existir, mas precisam ser planejadas. Definir pequenos intervalos para descanso e alimentação. Manter um ritmo acelerado pode levar à exaustão e reduzir a produtividade.

Acompanhe o “Casos e relatos” descrito a seguir para entender melhor como planejar seu tempo e organizar seu trabalho.



CASOS E RELATOS

Celular como aliado no dia a dia

Um consultor de projetos de TI tinha muitos problemas na sua rotina, pois frequentemente se esquecia de reuniões e de atividades importantes que deveriam ser executadas em datas e horários específicos. A quantidade de coisas para se lembrar era muito grande, e como ele era muito esquecido acabava tendo vários problemas.

Um dia ele teve a ideia de passar a cadastrar na agenda do celular cada item que precisasse se lembrar no futuro. Reuniões, tarefas, temas que precisaria cobrar de outras pessoas, e-mails que deveria responder, entre outras coisas que não poderiam ser esquecidas. A partir de então tudo mudou. Passou a ter a agenda como forte aliado na organização da rotina. Sempre que gerava uma atividade futura ou se lembrava de algo, cadastrava na agenda e era lembrado no momento certo.

Com essa simples atitude, o consultor passou a cumprir suas atividades de forma pontual, melhorando sua reputação entre os colegas e sua relação com seus líderes e com as áreas que atendia. Às vezes precisamos reconhecer nossos pontos fracos para criar meios de superá-los.

Confira a seguir a importância da iniciativa e da autonomia no ambiente de trabalho.

6.1.3 INICIATIVA E AUTONOMIA

A definição da palavra iniciativa, segundo o dicionário de português Michaelis on-line (2009), é apresentada a seguir:

sf (de iniciativo) 1 Qualidade de quem concebe e executa espontaneamente. 2 Direito ou prerrogativa de ser o primeiro a propor ou a começar uma coisa. 3 Ação ou efeito de ser o primeiro a pôr em prática uma ideia, a lembrá-la, enunciá-la ou propagá-la. 4 Atividade, diligência.

A iniciativa, dentro do ambiente de trabalho, refere-se às práticas proativas e empreendedoras individuais, e por isso é definida e caracterizada por cinco aspectos: ser consistente com a missão da organização; estar focada no longo prazo; ser direcionada para os objetivos e orientada pela ação; ser persistente frente às dificuldades (FRESE *et al.*, 1996).

Quando a organização se mostra aberta à iniciativa de sua equipe, falhas e desvios podem ser prevenidos, pois ideias e ações são mais facilmente estimuladas e produzidas para intervi-las. Assim, o desempenho da organização, em geral, tende a ser mais positivo quanto maior for o clima para as iniciativas pessoais.

Agora, após reconhecer a definição e o conceito da palavra iniciativa, você irá conhecer a definição da palavra autonomia, segundo o dicionário de português Michaelis on-line (2009):

sf (gr autonomía) 1 Qualidade de autônomo. 2 Autodeterminação político-administrativa de que podem gozar, relativamente, grupos (partidos, sindicatos, corporações, cooperativas etc.), em relação ao país ou comunidade política dos quais fazem parte. 3 Liberdade moral ou intelectual.

Quando você pensar em autonomia, lembre-se dos conceitos de independência e liberdade. Mas você não deve confundir esse significado com poder ilimitado quando falamos em atuação em uma empresa, afinal existem limites, mesmo que você venha a exercer uma autoridade formal por conta da hierarquia da estrutura organizacional.

Agora reflita como você tem internalizado esses dois conceitos tão fundamentais e de que forma os tem aplicado na sua rotina de trabalho.

Considere pensar em autonomia como sendo a sua independência moral e intelectual, o que lhe permite tomar decisões e agir nos momentos adequados, ou seja, ter iniciativa. Dentro do ambiente de trabalho, você deve levar em consideração a opinião e a predisposição em ajudar dos demais colegas. Compartilhar dúvidas para construir soluções de forma integrada não implica perda de autonomia, pois você faz parte de um todo maior. Quanto maior o engajamento nas atividades do trabalho e, conseqüentemente, quanto maior autonomia houver, será observada uma crescente tendência à criatividade, que é característica fundamental do conceito de inovação.



peshkov (2019)

Vários autores indicam que, quando os indivíduos têm um nível de autonomia moderadamente elevado em seu ambiente de trabalho, a criatividade e a conseqüente inovação são estimuladas.

A seguir você irá entender a importância da organização do trabalho.

6.2 ORGANIZAÇÃO DO TRABALHO

No mundo corporativo, o planejamento é a identificação dos objetivos de uma organização a fim de traçar metas para atingi-lo, identificando os recursos, dados e a melhor forma de utilizar os meios disponíveis para o fim desejado. Pensando em você como profissional, poderá realizar um planejamento pessoal, no qual você pode estabelecer suas próprias metas pessoais e/ou profissionais.

Para que você possa realizar um planejamento que seja eficaz, você deve considerar as prioridades. Assim, o essencial é ordenar as prioridades segundo alguns critérios, como:

- a) urgência;
- b) praticidade;
- c) menor custo;
- d) grau de dificuldade ou facilidade;
- e) prazer.

Você deve, ainda, priorizar critérios nesse processo, sejam eles de tempo, de execução e/ou custo de execução.



Jat306 (2018)

É muito importante que o local de trabalho esteja sempre limpo e organizado, evitando perda de documentos, atraso na realização de tarefas, entre outros. Descarte tudo o que não é necessário e organize de forma que facilite o seu acesso quando necessário. Dessa forma, a tendência é que seu trabalho seja realizado com mais qualidade.

A seguir, acompanhe o “Casos e Relatos” que demonstra a importância do planejamento e da organização no ambiente de trabalho.



CASOS E RELATOS

A importância do planejamento e organização

Franciele, estagiária de uma grande empresa de desenvolvimento de software, está um pouco atrapalhada com suas atividades e percebeu que suas tarefas diárias começaram a acumular e ela não conseguia mais cumprir os prazos de entregas solicitadas pelo seu gestor. A última entrega que ela precisava fazer era o levantamento de alguns fornecedores, porém Franciele não localizou todos os contratos e demorou muito para entregar a relação dos fornecedores para o seu gestor.

Como sua situação estava ficando crítica, Franciele resolveu pesquisar sobre como ela poderia planejar e se organizar para executar suas tarefas diárias.

Durante a pesquisa, Franciele descobriu uma ferramenta chamada 5S e resolveu utilizá-la para organizar seu posto de trabalho. Os colegas de Franciele observaram o movimento de limpeza das gavetas, eliminação de papéis não utilizados, enfim, e resolveram “entrar no clima”.

Foi um dia divertido, em que todos trabalharam em equipe organizando seu ambiente de trabalho como um todo. Franciele ficou animada, pois além do apoio dos seus colegas, pode ver sua mesa mais limpa e organizada, assim ela poderá localizar os documentos necessários com mais facilidade.

Dando continuidade aos seus estudos, a seguir você irá entender o que envolve a qualidade no trabalho.

6.3 QUALIDADE

A qualidade está relacionada à cultura, a um produto ou serviço, e à percepção do indivíduo, suas expectativas e necessidades, ou seja, o atendimento completo das necessidades de um cliente depende de implantação de processos para uma melhoria contínua.

Hoje, fala-se muito sobre vários tipos de qualidade: qualidade de um produto, qualidade de um serviço, qualidade de ensino, qualidade de vida.

Para todos esses casos, os objetivos são bem parecidos, pois a preocupação está em:

- a) satisfazer as necessidades dos clientes – utilizar a qualidade de forma a atender o que o cliente necessita com o que se produz na empresa;
- b) aumentar a produtividade – utilizar a qualidade para diminuir custo e diminuir todas as falhas internas do produto ou serviço;
- c) promover a realização socioprofissional dos trabalhadores – utilizar a qualidade para promover a motivação dos profissionais.

A qualidade pode, ainda, ser definida sob várias perspectivas que, apesar de diferentes, são complementares, como:

- a) qualidade quanto ao desempenho do produto – o aumento da qualidade está relacionado ao aumento de custo, porém, gera satisfação sob a ótica do consumidor;
- b) qualidade quanto à existência de deficiências – o objetivo é reduzir desperdícios, encargos após a venda e melhoria da imagem da empresa, por meio do aumento da qualidade em todas as fases da produção, motivando a diminuição do custo.

Segundo a ISO – International Organization for Standardization (Organização Internacional para Padronização) –, o conceito de qualidade é visto como “o grau no qual um conjunto de características inerentes satisfaz a requisitos”. (ABNT NBR ISO 9000:2005)

Qualidade pode, então, estar relacionada à conformidade do produto com relação às normas técnicas, com a adequação do produto ao uso a que ele se destina, ou ainda a um produto com maior desempenho e durabilidade.

Alguns atributos da qualidade de um produto são, por exemplo:

- a) desempenho;
- b) conveniência de uso;
- c) disponibilidade;
- d) confiabilidade;
- e) manutenibilidade;
- f) conformidade;
- g) assistência técnica;
- h) imagem, entre outros.

No próximo tópico, você irá estudar sobre a qualidade total. Acompanhe!

6.3.1 QUALIDADE TOTAL

Você já ouviu falar no termo Qualidade Total? Pois bem, a qualidade total é decorrente da aplicação da melhoria contínua (Filosofia Kaizen) nos processos de uma organização e tem como objetivo implicar toda a empresa, bem como fornecedores, distribuidores e demais parceiros.

Ela é um conjunto de ferramentas, programas e métodos aplicados no controle do processo de produção das empresas, a fim de obter bens e serviços pelo menor custo e melhor qualidade, objetivando atender as exigências e a satisfação dos clientes.

A gestão pela qualidade total nada mais é do que uma estratégia que tem como objetivo central maximizar a competitividade da empresa, ou seja, ampliar sua eficiência e eficácia. Essa gestão é vista por muitos como uma filosofia permeada por um conjunto de fundamentos, sustentada por um conjunto de ferramentas e técnicas.



Dimitre Camargo Martins (2018);

Quando se trata de qualidade, alguns termos acabam tendo o seu significado confundido, como é o caso de “eficiência” e “eficácia”, que muitas vezes são empregados de forma incorreta. Confira a diferença entre esses termos.

EFICIÊNCIA

A eficiência preocupa-se com os métodos, procedimentos e os meios para a melhor execução das coisas, a fim de otimizar recursos (máquinas, matéria-prima, pessoas) e aplicá-los de forma racional. Eficiência é realizar as coisas de maneira adequada, resguardar os recursos aplicados e resolver problemas.

EFICÁCIA

A eficácia indica o grau de sucesso obtido nos resultados. Ela visa aos resultados alcançados independentemente dos recursos utilizados, ou seja, conseguir atingir o objetivo. É possível atingir o ótimo e conseguir a eficácia e eficiência juntas, de forma a obter o resultado almejado utilizando o mínimo de recurso.

Você pode atingir um resultado (eficácia), mas os recursos utilizados para atingir o resultado foram desproporcionais (falta de eficiência).



FIQUE ALERTA

Lembre-se de que a qualidade não é mais um diferencial, e sim essencial para a sustentabilidade das empresas.



RECAPITULANDO

Neste capítulo você pôde perceber como atenção, organização, comprometimento são virtudes profissionais essenciais. E que organizar, e planejar as atividades, administrando da melhor forma o seu tempo é fundamental para vencer suas atividades. Você pôde ver ainda como compreender e aplicar os princípios da qualidade contribuem para o alcance do sucesso profissional.

REFERÊNCIAS

- BASTOS, Aderson et al. **Base de conhecimento em teste de software**. 3. ed. São Paulo (SP): Martins Fontes, 2012. 258 p. ISBN 9788580630534.
- MOLINARI, Leonardo. **Inovação e automação de testes de software**. São Paulo (SP): Érica, 2010. 140 p. ISBN 9788536502694.
- MOLINARI, Leonardo. **Testes de software**: produzindo sistemas melhores e mais confiáveis. 4. ed. São Paulo (SP): Érica, 2008. 228 p. ISBN 9788571949591.
- DELAMARO, Márcio; MALDONADO, José Carlos; JINO, Mário. **Introdução ao teste de software**. Rio de Janeiro (RJ): Elsevier: Campus, c2007. xi, 394 p. (Sociedade brasileira de computação). ISBN 9788535226348.
- PEZZÈ, Mauro; YOUNG, Michal. **Teste e análise de software**: processos, princípios e técnicas. Porto Alegre: Bookman, 2008. x, 512 p. ISBN 9788577802623 (broch.).
- ISTQB. **Certified Tester**: Foundation Level Syllabus, 77p.

MINICURRÍCULO DO AUTOR

THIAGO ROBERTO MENDES

Thiago Roberto Mendes é pós-graduando em Engenharia de Projeto de Software pela Universidade do Sul de Santa Catarina (Unisul), graduado em Gerenciamento de Redes de Computadores pelo Instituto de Ensino Superior da Grande Florianópolis (IES) e técnico em Telecomunicações com ênfase em Redes de Computadores pelo Instituto Federal de Santa Catarina (IFSC), certificado em gerenciamento de serviços de TI ITIL V3, certificado em governança de TI COBIT 4.1 e certificado em teste de software CTFL. No SENAI/SC, em Florianópolis, atua como professor do curso superior de Análise de Sistemas. Ao longo dos anos atuou como analista de rede e infraestrutura na empresa Cianet Networking, envolvendo atividades relacionadas a tecnologias e meios de transmissão de dados e VoIP. Atualmente também atua como analista de sistemas na empresa Softplan/Poligraph.

ÍNDICE

D

Documento de Teste, 9, 13, 67, 143

Drivers, 9, 75, 79, 80, 143

E

Estratégia de Teste, 9, 17, 22, 24, 27, 45, 51, 55, 62, 63, 64, 65, 66, 73, 143

G

Gerência e Planejamento de Testes, 9, 93, 143

P

Plano de Caso de Teste, 9, 68, 143

R

Relatórios de Testes, 9, 119, 120, 121, 124, 143

S

Stubs, 9, 75, 80, 143

T

Teste de Software, 9, 13, 15, 19, 48, 49, 65, 139, 141, 143

Testes de Performance, 9, 106, 107, 143

Testes Funcionais, 9, 75, 100, 117, 143

Testes Progressivos, 9, 75, 80, 143

Testes Regressivos, 80, 143

Testes Unitários, 9, 15, 24, 35, 57, 71, 75, 81, 82, 117, 143

SENAI – DEPARTAMENTO NACIONAL

UNIDADE DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA – UNIEP

Felipe Esteves Morgado
Gerente-Executivo

Luiz Eduardo Leão
Gerente de Tecnologias Educacionais

Catarina Gama Catão
Coordenação Geral do Desenvolvimento dos Livros Didáticos

SENAI – DEPARTAMENTO REGIONAL DE SANTA CATARINA

Fabrizio Machado Pereira
Diretor do SENAI/SC

Claudemir José Bonatto
Diretor de Educação do SENAI/SC

Daniel Thiesen Horongoso
Gerente Executivo da Regional Sudeste SENAI/SC

Ricardo Maximo Anzolin
Gerente de Operações da Regional Sudeste SENAI/SC

Priscila Carneiro Gallasse Cesconetto
Supervisora de Consultoria em Educação

Thiago Roberto Mendes
Elaboração

Felipe Demarchi
Revisão técnica

Michele Antunes Corrêa
Design educacional

Davi Leon Dias
D'imitre Camargo Martins
Ilustrações e tratamento de imagens

IStock
SENAI/SC
Banco de imagens

Tatiana Daou Segalin
Diagramação

Tatiana Daou Segalin
Revisão e Fechamento de Arquivos

Luciana Effting Takiuchi
CRB – 14/937
Ficha Catalográfica

i-Comunicação
Projeto Gráfico

Editorar Multimídia Ltda.
Revisão Ortográfica e Gramatical

Editorar Multimídia Ltda.
Normalização



*Iniciativa da CNI - Confederação
Nacional da Indústria*

ISBN 978-855052423-8



9

788550

524238