ENGENHEIRO DE QUALIDADE DE SOFTWARE

ANDRÉ DIENES FRIEDRICH

Análise de Qualidade

São Bento do Sul

2023

1. RESUMO

Durante minha transição de carreira em junho de 2022, decidi seguir a área de qualidade e buscar oportunidades no mercado. Foi nesse contexto que descobri o curso profissionalizante oferecido pela EBAC, intitulado "Engenheiro de Qualidade de Software". Neste trabalho, gostaria de compartilhar alguns dos pontos e habilidades que adquiri ao longo dos módulos do curso.

Ao longo do programa, adquiri conhecimentos sobre estratégias de teste, aprendendo a desenvolver critérios de aceite e casos de teste eficazes. Além disso, explorei a criação de um repositório no GitHub para gerenciar e versionar meus testes, bem como a importância dos testes automatizados no ciclo de desenvolvimento de software.

Outro aspecto abordado no curso foi a realização de testes de performance, compreendendo como avaliar a capacidade e a eficiência de um sistema em termos de resposta e escalabilidade. Além disso, aprendi sobre a integração contínua de testes, ou seja, como incorporar os testes de forma contínua e automatizada no processo de desenvolvimento, garantindo a detecção precoce de problemas e a entrega de software de qualidade.

No decorrer deste trabalho, apresentarei os principais insights e aprendizados relacionados a esses tópicos, destacando as habilidades e competências adquiridas ao longo do curso.

2. SUMÁRIO

3. INTRODUÇÃO

O profissional atuante na área de Teste de Software desempenha um papel fundamental na monitoração de cada etapa do desenvolvimento de um software, com o objetivo de assegurar o cumprimento dos resultados esperados. É de sua responsabilidade criar planos de testes, rastrear bugs, desenvolver padrões de qualidade e identificar potenciais problemas que possam afetar o usuário final.

O planejamento é uma atividade essencial em qualquer projeto, desempenhando um papel análogo ao de um "mapa". Sem um plano, um mapa ou qualquer outra fonte de informação semelhante, os objetivos e o destino do projeto permanecem desconhecidos, tornando-se impossível ter a certeza de ter alcançado a meta almejada. Compreender o propósito do planejamento é de suma importância, pois permite a monitoração adequada das atividades em execução. Além disso, é crucial compreender o papel dos riscos no planejamento e a distinção entre estratégias e planos.

O planejamento engloba três atividades principais: a definição de um cronograma de atividades, que estabelece as tarefas a serem realizadas, as etapas a serem seguidas e a ordem cronológica de execução; a alocação de recursos, que define quais pessoas serão responsáveis por cada atividade e quais ferramentas ou recursos serão utilizados; e a definição de marcos de projeto, estabelecendo pontos de referência a serem alcançados para o acompanhamento do progresso.

A atividade de monitoração ou supervisão, que acompanha o planejamento, tem como objetivo avaliar se o progresso alcançado está em conformidade com o que foi estabelecido no plano, respondendo à pergunta: "Quão bem estamos progredindo no projeto?".

No contexto do desenvolvimento de software, vários documentos são necessários, tais como o plano de projeto, o documento de requisitos e o plano de teste. Neste artigo, o foco está no último documento mencionado, o plano de teste. Trata-se de um documento ou mapa que define o escopo, os objetivos, os requisitos, as estratégias e os recursos a serem empregados nas atividades de testes de software. O artigo apresenta os elementos que devem ser incluídos em

um documento de plano de teste, fornecendo exemplos e discussões sobre cada um deles.

O teste de software é uma das atividades do processo de desenvolvimento de sistemas de software, visando executar um programa de forma sistemática com o objetivo de identificar falhas. Essa atividade requer a verificação e validação do software, envolvendo o estabelecimento do momento de início e término das atividades de verificação e validação, a avaliação dos atributos de qualidade e o controle dos releases do software ao longo do processo de desenvolvimento.

Além de identificar falhas, os testes visam aumentar a confiabilidade de um sistema de software, ou seja, a probabilidade de que o sistema continue funcionando sem falhas ao longo do tempo. Embora seja desejável testar um sistema por completo, é importante ter em mente que a atividade de teste apenas assegura a identificação das falhas existentes, mas não garante sua ausência. Portanto, as atividades de teste devem ser conduzidas de forma disciplinada para identificar a maioria dos erros presentes.

Realizar testes de software implica responder a uma série de questões, como quais atributos de qualidade devem ser testados, quem será responsável pelos testes, quais recursos serão utilizados, quais são as dependências entre os atributos de qualidade, quais são as dependências entre as atividades de desenvolvimento e como o processo e a qualidade do sistema de software serão monitorados.

Nas próximas seções deste artigo, serão apresentados três casos práticos, que exemplificam e aprofundam cada um dos tópicos abordados no capítulo 4, proporcionando uma visão mais abrangente do conhecimento adquirido durante o curso.

4. O PROJETO

CASE A SER RESOLVIDO

Para este trabalho de conclusão de curso **Profissão: Engenheiro de Qualidade de software**, você deve utilizar o conhecimento adquirido ao longo do curso para elaborar uma estratégia de testes adequada para validar o e-commerce EBAC Shop (http://lojaebac.ebaconline.art.br/). Você deve considerar as histórias de usuário já refinadas como se você estivesse participando de um time ágil. As funcionalidades devem seguir todo o fluxo de trabalho de um *Quality Engineer* (QE), desde o planejamento até a entrega. Siga as etapas dos subtópicos para se orientar no trabalho.

ATENÇÃO:

- Conforme a sua estratégia, você pode executar os testes no endereço disponibilizado ou utilizando as imagens disponíveis no Docker Hub:
    - Banco de Dados: ernestosbarbosa/lojaebacdb
    - Loja EBAC: ernestosbarbosa/lojaebac

        - Comandos para subir os containers:

```
docker network create --attachable ebac-network

docker run -d --name wp_db -p 3306:3306 --network ebac-network ernestosbarbosa/lojaebacdb:latest

docker run -d --name wp -p 80:80 --network ebac-network ernestosbarbosa/lojaebac:latest
```

Após subir os containers a loja estará em http://localhost:80

4.1 Estratégia de teste

- Faça uma estratégia de testes em um mapa mental, seguindo algumas diretrizes como objetivos, papeis e responsabilidades, fases de testes, padrões, tipos de testes, técnicas de testes, ambientes, ferramentas, abordagem (manual ou automatizado), framework ou ferramenta usados, plataformas (web, api, mobile), etc.;
- Referência: Módulo 5

4.2 Critérios de aceitação

- Considere as histórias de usuário:
  - [US-0001] – Adicionar item ao carrinho
  - [US-0002] – Login na plataforma
  - [US-0003] – API de cupons
- Para cada uma delas crie pelo menos 4 critérios de aceitação usando a linguagem Gherkin;
- Crie histórias de usuário para as funcionalidades:
  - Catálogo de Produtos
  - Painel Minha Conta
  - Meus Pedidos
  - Endereços
  - Detalhes da Conta
- Referência: Módulo 8

4.3 Casos de testes

- Crie pelo menos 4 casos de testes para cada história de usuário, sempre que possível, usando as técnicas de testes (partição de equivalência, valor limite, tabela de decisão etc.).
- Considere sempre o caminho feliz (fluxo principal) e o caminho alternativo e negativo (fluxo alternativo). Exemplo de cenário negativo: “Ao preencher com usuário e senha inválidos deve exibir uma mensagem de alerta...”
- Identifique quais os casos de teste serão automatizados, sendo ao menos 1 caminho feliz e 1 caminho alternativo.
- Referência: Módulos 4 e 5

4.4 Repositório no GitHub

- Crie um repositório no GitHub com o nome TCC-EBAC-QE;
- Deixe o repositório público até a análise dos tutores;
- Neste repositório você deve subir este arquivo e todos os código fontes das automações que criar.
- Referência: Módulo 10
- Link do repositório: https://github.com/Guspex/EBAC-QA/tree/main/Mod34

4.5 Testes automatizados

4.5.1 Automação de UI

- Crie um projeto de automação WEB com o framework e a linguagem que preferir
- Justifique a sua escolha através de um comparativo entre ao menos 3 opções de ferramentas e linguagem.
- Crie uma pasta chamada UI para os testes WEB dos casos de teste que forem automatizados
- Utilize ao menos um *Testing Pattern* (à sua escolha) na implementação dos testes.

4.5.2 Automação de API

- Crie uma pasta chamada API para os testes de API dos casos de teste que forem automatizados
- Você deve utilizar a ferramenta Supertest para criar seus testes de API
- Não esqueça de validar os contratos! ☺

4.5.3 Automação Mobile

- Considere para os APPs apenas a funcionalidade de Catálogo de Produtos
- Você pode encontrar os APPs em:
  - *Android*: https://github.com/EBAC-QE/testes-mobile-ebac-shop/tree/main/app/android
  - *iOS*: https://github.com/EBAC-QE/testes-mobile-ebac-shop/tree/ios-tests/app/ios
- Crie uma pasta chamada Mobile para os testes em aplicativos dos casos de teste que forem automatizados
- Utilize ao menos um *Testing Pattern* (à sua escolha) na implementação dos testes.
- Você deve implementar testes para ao menos uma das plataformas Mobile (*Android* ou *iOS*)

- Observações:
  - Considere todas as boas práticas aprendidas até aqui
  - Não esqueça de implementar a geração de relatórios

- Referência: Módulos 11, 12, 14, 16, 17, 22, 23, 24, 29 e 30

4.6 Integração contínua

- Execute os testes automatizados em integração contínua utilizando o Github Actions

- Referência: Módulo 26

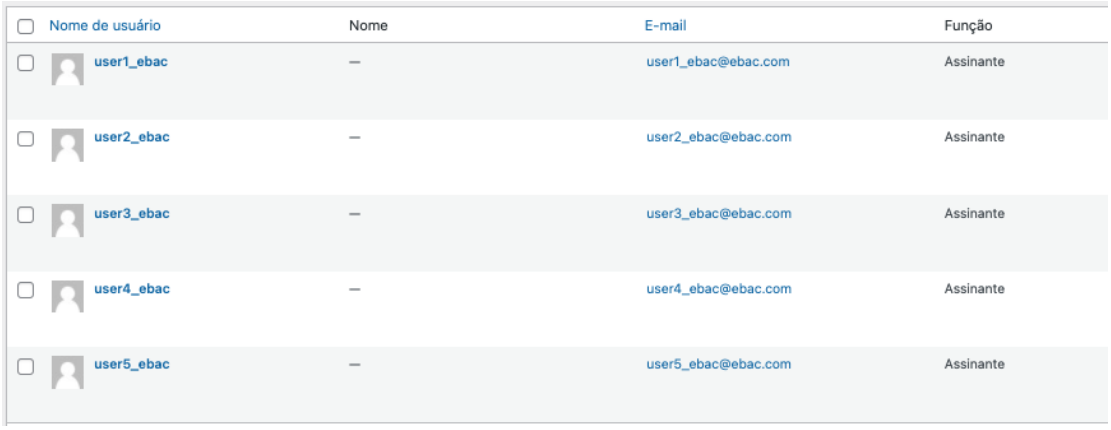
4.7 Testes de performance

- Usando o K6, implemente um teste de performance em ao menos 2 casos de testes
- Referência: Módulo 28

- Configurações do teste de performance:

  -Usuários virtuais: 20
  -Tempo de execução: 2 minutos
  -RampUp: 20 segundos
  -Massa de dados: Usuário / senha:

  user1_ebac / psw!ebac@test
  user2_ebac / psw!ebac@test
  user3_ebac / psw!ebac@test
  user4_ebac / psw!ebac@test
  user5_ebac / psw!ebac@test



Nome de usuário	Nome	E-mail	Função
user1_ebac	–	user1_ebac@ebac.com	Assinante
user2_ebac	–	user2_ebac@ebac.com	Assinante
user3_ebac	–	user3_ebac@ebac.com	Assinante
user4_ebac	–	user4_ebac@ebac.com	Assinante
user5_ebac	–	user5_ebac@ebac.com	Assinante

5. CONCLUSÃO

Durante este curso, pude adquirir conhecimentos e habilidades essenciais para atuar como profissional na área de Teste de Software, especialmente no contexto de desenvolvimento de aplicativos web e mobile. Através do estudo dos processos de desenvolvimento, aprendi a importância do planejamento e da estratégia de testes para garantir a qualidade do software.

Ao longo do curso, compreendi a necessidade de criar um plano de testes adequado, considerando os objetivos, papéis e responsabilidades, fases de testes, padrões, tipos de testes, técnicas, ambientes e ferramentas a serem utilizadas. Também explorei a importância dos critérios de aceitação, que auxiliam na definição e validação das funcionalidades do software.

Além disso, tive a oportunidade de praticar a criação de casos de testes, utilizando técnicas como partição de equivalência e valor limite. Identifiquei os casos de teste que poderiam ser automatizados, tanto no caminho feliz quanto em caminhos alternativos e negativos. Aprendi sobre diferentes abordagens de automação, tanto para testes de interface do usuário (UI) quanto para testes de API e mobile.

Compreendi também a importância da integração contínua, que permite a execução automatizada dos testes em um ambiente de desenvolvimento colaborativo. Essa prática proporciona uma maior confiabilidade e agilidade no processo de entrega do software.

Por fim, explorei a área de testes de performance, utilizando a ferramenta K6 para implementar testes de desempenho em diferentes casos de teste. Isso me permitiu avaliar o comportamento do software em condições de uso mais intensas, garantindo a sua estabilidade e escalabilidade.

Após a conclusão deste curso, não apenas adquiri conhecimentos teóricos, mas também pude aplicá-los na prática. Hoje, como profissional alocado no mercado, posso colocar em prática todas as habilidades e boas práticas que aprendi, contribuindo para a qualidade e sucesso dos projetos de software em que estou envolvido.

Em suma, este curso foi fundamental para o meu crescimento profissional na área de Teste de Software, fornecendo as bases teóricas e práticas necessárias para desempenhar um papel efetivo no desenvolvimento de software de qualidade. Estou confiante de que as habilidades adquiridas neste curso serão valiosas ao longo da minha carreira.

6. REFERÊNCIAS BIBLIOGRÁFICAS

Plano de Teste - Um Mapa Essencial para Teste de Software.

Disponível em: https://www.devmedia.com.br/plano-de-teste-um-mapa-essencial-para-teste-de-software/13824

Curso de Teste de Software online – aprenda com EBAC Online.

Disponível em: https://ebaconline.com.br/qualidade-de-software?utm_source=google&utm_medium=cpc&utm_campaign=ccourse_44_sowtware-testing_google_search_all_conversions_all&utm_content=c_14670107930

Exemplo: Plano de Teste.

Disponível em: https://www.cin.ufpe.br/~gta/rup-vc/extend.formal_resources/guidances/examples/resources/test_plan_v1.htm

Curso de Teste de Software online – aprenda com EBAC Online.
Disponível em: https://ebaconline.com.br/qualidade-de-software?utm_source=google&utm_medium=cpc&utm_campaign=ccourse_44_sowtware-testing_google_search_all_conversions_all&utm_content=c_14670107930

Qualidade de software: o que é e como avaliar o seu resultado?
Disponível em: https://www.monitoratec.com.br/blog/qualidade-de-software/#:~:text=O%20que%20%C3%A9%20qualidade%20de%20software%3F