



ENGENHEIRO DE QUALIDADE DE SOFTWARE

MARIA CAROLINA LOPES BRAULINO

Análise de Qualidade

Curitiba

2023

1. RESUMO

No final de 2021 me interessei pelo profissional de qualidade e estava em busca de adentrar neste mercado. Assim, encontrei o curso profissionalizante presente na EBAC, o de “Engenheiro de qualidade de software” e neste trabalho irei trazer alguns pontos e skills que adquiri ao longo dos módulos, como estratégias de teste, como é a criação dos critérios de aceite, casos de testes, a criação de um repositório no GitHub, como são os testes automatizados, testes de performance e a realização de integração continua destes testes.

2. SUMÁRIO

1. RESUMO.....	2
2. SUMÁRIO	3
3. INTRODUÇÃO	4
4. O PROJETO	6
4.1 Estratégia de teste	6
4.2 Critérios de aceitação	7
4.3 Casos de testes	8
4.4 Repositório no Github	8
4.5 Testes automatizados	8
4.6 Integração contínua.....	9
4.7 Testes de performance	9
5. CONCLUSÃO	10
6. REFERÊNCIAS BIBLIOGRÁFICAS	10

3. INTRODUÇÃO

O profissional da área de Teste de Software monitora cada fase do desenvolvimento de um software para garantir que os resultados esperados sejam cumpridos. Fica sob a responsabilidade do profissional da área de Teste de Software criar planos de testes, rastrear bugs, desenvolver padrões de qualidade e identificar potenciais problemas para o usuário.

Uma atividade essencial no desenvolvimento de todo e qualquer projeto é o planejamento. Um plano tem o papel semelhante ao de um 'mapa'. Sem um mapa, um plano ou qualquer outra fonte de informação similar, você não conhecerá seus objetivos, nem aonde quer chegar e jamais terá a certeza de ter alcançado sua meta. Perceba que entender o propósito do planejamento é de suma importância a fim de monitorar a execução de atividades, sendo também importante conhecer o papel dos riscos no planejamento, bem como diferenciar estratégias de planos. Planejamento engloba três atividades principais:

- Definir um cronograma de atividades: estabelecer as atividades que devem ser realizadas, as etapas a serem seguidas e a ordem cronológica de execução;
- Fazer alocação de recursos: definir quem realiza as atividades e quais ferramentas/recursos a serem utilizados;
- Definir marcos de projeto – estabelecer os marcos, ou milestones, a serem alcançados com objetivo de se fazer o acompanhamento.

Perceba que o planejamento é acompanhado da atividade de monitoração ou supervisão que visa avaliar se o progresso que tem sido alcançado está em conformidade com o que foi estabelecido no plano ou, em outras palavras, responder a questão: quão bem estamos indo no projeto?

Agora, dentro do contexto do desenvolvimento de software, você necessitará de vários documentos como, por exemplo, plano de projeto, documento de requisitos e plano de teste. Neste artigo, o foco recai sobre o último, isto é, plano de teste. Trata-se de um documento ou mapa no qual se definem escopo e objetivos, além de requisitos, estratégias e recursos a serem empregados nas atividades de testes de software. Nesse sentido, o artigo apresenta os itens que

devem fazer parte de um documento de plano de teste, exemplificando e discutindo esses itens.

Teste de software é uma das atividades do processo de desenvolvimento de sistema de software que visa executar um programa de modo sistemático com o objetivo de encontrar falhas. Perceba que isto requer verificação e validação de software. Nesse sentido, definir quando as atividades de verificação e validação iniciam e terminam, como os atributos de qualidade serão avaliados e como os releases do software serão controlados, são questões que devem ser acompanhadas ao longo do processo de software.

E, além de encontrar falhas, testes objetivam aumentar a confiabilidade de um sistema de software, isto é, aumentar a probabilidade de que um sistema continuará funcionando sem falhas durante um período de tempo.

Embora seja desejável testar um sistema por completo, deve-se ter em mente que a atividade de teste assegura apenas encontrar falhas se ela(s) existirem, mas não asseguram sua ausência. Portanto, as atividades devem ser disciplinadas a fim de identificar a maioria dos erros existentes. Note que realizar os testes de software implica em responder às questões:

1. Quais atributos da qualidade deverão ser testados?
2. Quem realizará os testes?
3. Quais recursos serão utilizados?
4. Quais as dependências entre os atributos de qualidade?
5. Quais as dependências entre as atividades de desenvolvimento?
6. Como o processo e a qualidade do sistema de software serão acompanhados?

Portanto, nas sessões seguintes estarei mostrando um pouco mais do conhecimento que consegui adquirir durante o curso mostrando a resolução de 3 cases e trazendo um pouco mais sobre cada tópico do capítulo 4.

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 sub-tópicos 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](https://hub.docker.com/r/ernestosbarbosa/lojaebacdb)
 - Loja EBAC: [ernestosbarbosa/lojaebac](https://hub.docker.com/r/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
```

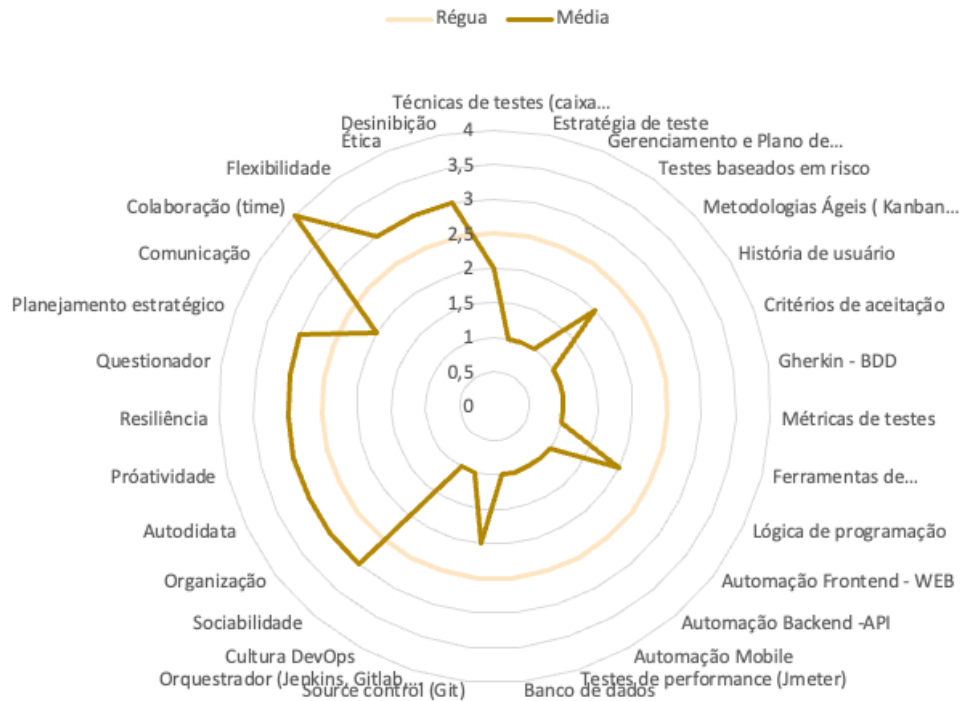
Após subir os containers a loja estará em <http://localhost:80>

4.1 Estratégia de teste

- Faça uma estratégia de testes em um mapa mental, seguindo algumas diretrizes como objetivos, papéis 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

Radar de Habilidades



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 publico 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/CarolBraulino/exercicios_ebac/tree/main/M34%20-%20Projeto%20final

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

<input type="checkbox"/> Nome de usuário	Nome	E-mail	Função
<input type="checkbox"/>  user1_ebac	—	user1_ebac@ebac.com	Assinante
<input type="checkbox"/>  user2_ebac	—	user2_ebac@ebac.com	Assinante
<input type="checkbox"/>  user3_ebac	—	user3_ebac@ebac.com	Assinante
<input type="checkbox"/>  user4_ebac	—	user4_ebac@ebac.com	Assinante
<input type="checkbox"/>  user5_ebac	—	user5_ebac@ebac.com	Assinante

5. CONCLUSÃO

Com este curso pude aprender, além das habilidades para ser uma profissional na área, os processos básicos de como funciona o desenvolvimento de software para aplicativos web e mobile. Pude aprender sobre os princípios de testes, como pôr em pratica o modo ágil e ferramentas para automatização de testes.

Após longos meses de aprendizado e me ver chegando até aqui, posso dizer que aprendi muito mais do que eu esperava, hoje sou uma profissional já alocada no mercado e consigo pôr em prática tudo que aprendi aqui.

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>>. Acesso em: 03 abr. 2023.

Curso de Teste de Software online – aprenda com EBAC Online. Disponível em: <https://ebaonline.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>. Acesso em: 04 abr. 2023.

Exemplo: Plano de Teste. Disponível em: <https://www.cin.ufpe.br/~gta/rup-vc/extend.formal_resources/guidances/examples/resources/test_plan_v1.htm>. Acesso em: 03 mar. 2023.

Curso de Teste de Software online – aprenda com EBAC Online. Disponível em: <https://ebaonline.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>. Acesso em: 10 mar. 2023.

TEAM, M. **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>>. Acesso em: 11 abr. 2023.