

Estratégia de Testes e Gestão de Configuração

PetSpeed

Recife, 2019.

Estratégia de Testes

Teste de software é um processo sistemático que tem por objetivo identificar prováveis defeitos, verificando se o software realiza suas tarefas de forma correta (isto é, conforme os requisitos fornecidos pelos stakeholders) e também se eventualmente faz o que não deveria fazer.

O desenvolvimento de software costuma ser uma atividade bastante complexa que muitas vezes leva à ocorrência de falhas no sistema resultante. Estas falhas podem estar presentes em todas as etapas do processo de desenvolvimento e este fato pode conduzir uma equipe à geração de um produto diferente do que foi inicialmente especificado. Um bug gerado por um programador pode ocasionar um erro que irá gerar um estado de inconsistência em uma determinada funcionalidade, ou seja, erros podem causar uma falha, que é um comportamento inesperado em um software e isso pode afetar o usuário final. Quanto mais tarde um erro for descoberto dentro um software, maiores serão o tempo e a complexidade para a correção. Em diversos casos, muitos pontos do software podem precisar ser refeitos.

O teste de software aplicado que vai garantir a qualidade é o teste de aceitação. Este acontece na camada da interface de usuário e normalmente envolvem todas as camadas da aplicação testada para validar um cenário de negócio. São testes formais com relação às necessidades dos usuários, requisitos e processos de negócios conduzidos para determinar se um sistema satisfaz ou não os critérios de aceitação. O cenário em teste de aceitação será, login e logout, cadastro de usuários, cadastro de animais, agendamento, entre outros.

Os testes de aceitação são muito mais sensíveis a alterações no sistema do que qualquer outro tipo de teste, justamente, por testar o fluxo de informações entre várias camadas de uma aplicação distribuída, precisam de um trabalho constante para que fiquem sempre atualizados. Podem ser criados para qualquer tipo de aplicação, seja web, mobile, desktop, watch e até aplicações de background.

Gestão de Configuração

1 Ferramentas, Ambiente e Infra-estrutura

Tipo	Ferramenta
Sistema Operacional	Linux/Windows
Controle de Versão	GIT
Ambiente de Desenvolvimento (IDE)	Android Studio
Editor de Diagramas	draw.io
Comunicação	Discord/Hangouts

Requisitos mínimos de Hardware do Ambiente de Desenvolvimento

- Microprocessador dual-core 1.8GHz com suporte ao quemu;
- Memória RAM 4GB;
- Espaço livre em disco 16GB;
- Conexão com a Internet.

Requisitos mínimos de Hardware do Ambiente de Execução (smartphone)

- Microprocessador dual-core 1GHz (com sistema Android 4.0 ou superior instalado);
- Memória RAM 512MB;
- Espaço livre em memória interna 2MB.

2 O Programa de Gerenciamento de Configuração

2.1 Estrutura do Repositório de Versões

O GitHub é uma plataforma de hospedagem de projetos de software que oferece, além de armazenamento de repositórios GIT e uma área de repositório específica de documentação. A fim de se ter melhor aproveitamento da plataforma e agilidade no ciclo de vida Scrum do PetSpeed, todos os artefatos de software e o gerenciamento do projeto será feito no GitHub e/ou em ferramentas integradas ao GitHub.

Repositório principal

O repositório do PetSpeed, podendo ser controlado pelo sistema de controle de versão GIT, deve ter os seguintes artefatos nos respectivos diretórios relacionados abaixo:

Diretório	Conteúdo
Diagramas-PetSpeed	Diagramas do App PetSpeed
PetSpeed-App	Código-fonte com Java – projeto do Android Studio
petspeedufrpe.github.io	Link para acesso ao site do projeto
Prototipo	README do protótipo do aplicativo

2.2 Estimativa do Status de Configuração

2.2.1 Processo de Armazenamento de Mídia e Liberação do Projeto

O repositório deverá ser clonado por todos os integrantes da equipe em um diretório monitorado por algum aplicativo de sincronização e caso algum imprevisto aconteça o código-fonte estará salvo e pode ser recuperado no qual estava o diretório do repositório Git.

3 Marcos

O Plano de Gerência de Configuração é alterado nos seguintes casos:

- O repositório de versões não está atendendo as necessidades dos integrantes da equipe;
- O Plano não foi aprovado pelo cliente e precisa ser refatorado.