



Gestão e Qualidade de Software

Integrantes do grupo:

Kayky Cerquiaro Prado - 822155538

Enrico Aguiar Vrunski - 82210618

Thiago Ferreira Lima Gonçalves - 824156179

Matheus Tognon Siqueira - 824141731

Felipe Soares de Oliveira - 824156311

A revisão de software é um conjunto de atividades realizadas com o objetivo de melhorar a qualidade de um sistema, detectando falhas e melhorias antes de sua entrega final. Ela é fundamental para garantir que o software atenda aos requisitos especificados e funcione corretamente. Diversas técnicas podem ser empregadas no processo de revisão de software, e cada uma delas se foca em diferentes aspectos do código ou do sistema como um todo. A seguir, farei uma explicação mais detalhada sobre as principais técnicas de revisão de software, destacando também referências bibliográficas que podem ser úteis, incluindo o livro de Persman. A seguir segue algumas técnicas de revisão de software

Revisão de Código (Code Review)

A revisão de código é uma prática central nas equipes de desenvolvimento de software. Consiste na análise do código-fonte do sistema por parte de outros programadores, com o objetivo de identificar problemas de lógica, segurança, desempenho e aderência aos padrões de codificação. Essa técnica não só ajuda a detectar falhas técnicas, mas também facilita a troca de conhecimento entre os membros da equipe e promove a consistência no código. A revisão de código pode ser feita de diversas formas, como revisões informais ou mais estruturadas, com o uso de ferramentas específicas de integração contínua que automatizam parte do processo.

Em termos de bibliografia, uma boa fonte sobre o assunto é o livro "Software Engineering: A Practitioner's Approach" de Roger S. Pressman. O autor discute amplamente a importância da revisão de código, destacando como ela contribui para a melhoria contínua do software.

Teste de Caixa-Branca

O teste de caixa-branca (ou teste estrutural) é uma técnica em que o testador tem conhecimento total do código-fonte do sistema e testa o software verificando seu comportamento interno. O objetivo é garantir que todas as partes do código sejam executadas e que todos os caminhos lógicos sejam verificados. Este tipo de teste é útil para detectar erros em níveis mais profundos, como falhas de lógica, condições não verificadas e erros de fluxo de controle.

Para mais informações sobre testes de software, o livro de Pressman também aborda o teste de caixa-branca, explicando como ele pode ser usado para garantir uma cobertura completa do código.

Teste de Caixa-Cinza

O teste de caixa-cinza combina elementos tanto do teste de caixa-preta quanto do teste de caixa-branca. Embora o testador tenha conhecimento da estrutura interna do sistema, o foco continua sendo o comportamento do software, ou seja, a análise de como ele responde a diferentes entradas e condições de operação. Esse tipo de teste é útil para

validar tanto a funcionalidade quanto o desempenho do sistema de forma mais equilibrada.

O teste de caixa-cinza é discutido em "Software Engineering" por Pressman, que detalha como a combinação dessas abordagens pode fornecer uma visão mais completa do comportamento do sistema.

Teste de Regressão

O teste de regressão envolve a execução de testes repetidos após modificações ou atualizações no software para garantir que as funcionalidades que estavam funcionando corretamente não foram prejudicadas pelas alterações. Esse tipo de teste é crucial durante o ciclo de vida de um software, especialmente em sistemas complexos, onde mudanças em uma parte do código podem afetar outras áreas de maneira inesperada.

No livro de Pressman, o conceito de teste de regressão é amplamente abordado, e são fornecidos exemplos de como organizar esses testes de forma eficaz para garantir a estabilidade do sistema ao longo do tempo.

Refatoração

Refatoração é o processo de modificar o código-fonte de um sistema para melhorar sua estrutura interna sem alterar seu comportamento externo. Ela visa melhorar a legibilidade, reduzir a complexidade e facilitar a manutenção. Refatorar o código é uma prática contínua que ajuda na adaptação do software às novas necessidades, sem a necessidade de reescrever grandes partes do sistema.

O conceito de refatoração é detalhado no livro de Pressman, que discute como essa técnica pode ser implementada de forma eficaz durante o ciclo de vida do desenvolvimento de software.

Análise Estática de Código

A análise estática é o processo de examinar o código sem executá-lo, utilizando ferramentas automatizadas para identificar erros, vulnerabilidades de segurança, violações de padrões de codificação e outros problemas. A principal vantagem dessa técnica é que ela pode ser realizada nas fases iniciais do desenvolvimento, ajudando a detectar problemas antes que o software seja executado ou distribuído.

No contexto do livro de Pressman, a análise estática de código é vista como uma prática importante para garantir a qualidade do software desde o início, antes mesmo de os testes dinâmicos serem realizados.

Bibliografia

- **Pressman, Roger S..** *Software Engineering: A Practitioner's Approach*. 9ª edição. McGraw-Hill, 2014. Este livro é uma das referências mais completas sobre engenharia de software, abordando técnicas de revisão, testes e outras práticas de qualidade no desenvolvimento de software.
- **Perry, William E..** *Software Engineering: A Practitioner's Approach*. Este livro complementa a leitura de Pressman, oferecendo uma visão detalhada sobre práticas de teste e revisão, com exemplos práticos.
- **Somerville, Ian..** *Software Engineering*. 10ª edição. Pearson, 2016. Outro texto fundamental para quem busca entender mais sobre práticas de engenharia de software, incluindo técnicas de revisão e testes.