

Universidade São Judas Tadeu - Butantã
Noturno

Nomes: Gabriel Carvalho dos Santos - 821159957
Gustavo Mendes de Lima - 821139542
Matheus dos Santos de Souza - 821133576

Turma: GQS-CCP1AN-BUE1

Professor: Robson Calvetti

Conceitos de Técnicas de Revisão de Software.

As técnicas de revisão de software são essenciais para garantir a qualidade e a confiabilidade dos sistemas de software. Cada técnica tem seu propósito e contexto específico, e a escolha da técnica correta depende das necessidades do projeto e da equipe. Implementar uma abordagem equilibrada e bem estruturada para revisões pode levar a um software mais robusto e de alta qualidade.

Revisão de Código (Code Review):

É o processo onde desenvolvedores revisam o código-fonte de colegas para identificar e corrigir erros, melhorar a qualidade e garantir conformidade com padrões. Pode ser feita de forma informal (revisão por pares) ou formal (revisão estruturada com reuniões e documentação, como a Fagan Inspection).

Inspeções (Inspections):

São revisões formais onde um grupo analisa sistematicamente um artefato de software. O processo inclui planejamento, preparação, revisão, correção e follow-up para identificar e resolver defeitos.

Walkthroughs:

Revisões informais onde o autor do software guia os revisores através do código ou documentação, explicando decisões e buscando feedback. Incluem walkthroughs de código e documentação.

Auditorias (Audits):

Revisões formais realizadas por uma equipe independente para verificar conformidade com normas e avaliar a eficácia dos processos de desenvolvimento.

Testes de Revisão (Review Testing):

Envolve criar e executar testes para validar que os problemas identificados foram corrigidos e garantir que as correções não introduzem novos defeitos.

Análise de Métricas e Relatórios:

Métricas como densidade de defeitos e taxa de revisão ajudam a avaliar a eficácia das técnicas de revisão, enquanto relatórios fornecem insights para melhorar o processo e a qualidade do software.

Benefícios das Técnicas de Revisão de Software:

Incluem a detecção precoce de defeitos, melhoria da qualidade do código, facilitação do conhecimento entre a equipe e redução dos custos de correção.

Desafios e Considerações:

Podem incluir o tempo e recursos necessários para revisões, resistência a críticas e a necessidade de padronização das práticas de revisão entre os membros da equipe.

Impacto de Defeitos de Software nos Custos:

Defeitos de software se tornam progressivamente mais caros de corrigir conforme o software avança no ciclo de vida. Corrigir erros na fase inicial é mais barato do que após testes ou lançamento, onde correções podem exigir retrabalho e ter impacto na reputação.

Amplificação e Eliminação de Defeitos:

Defeitos podem se amplificar, afetando outras partes do sistema. A eliminação precoce desses problemas é crucial para evitar sua propagação e reduzir o impacto. Técnicas de revisão ajudam a detectar e corrigir erros cedo.

Análise de Métricas de Revisão:

Métricas como densidade de defeitos e taxa de revisão avaliam a eficácia das revisões e a qualidade do desenvolvimento. Essas métricas fornecem insights sobre o processo e áreas para melhoria.

Eficácia dos Custos de Revisões:

A eficácia dos custos de revisões é medida pela relação entre os custos das revisões e os benefícios obtidos, como a redução de defeitos. Revisões bem executadas podem resultar em economias ao corrigir problemas antes que se tornem mais complexos e caros.

Revisões e seu Espectro de Formalidade:

Revisões variam de informais a formais. Revisões informais, como revisões por pares, são rápidas e colaborativas, enquanto revisões formais, como inspeções, seguem processos estruturados e detalhados.

Revisões Informais:

São práticas rápidas e flexíveis, como revisões por pares, onde um desenvolvedor revisa o código de outro. Essas revisões facilitam a identificação rápida de problemas e o compartilhamento de conhecimento.

Revisões Técnicas Formais:

Incluem reuniões de revisão estruturadas, relatórios detalhados documentando defeitos e ações corretivas, diretrizes para garantir consistência e eficácia, e revisões por amostragem para análise representativa do trabalho.

Avaliações Post-mortem:

Realizadas após a conclusão de um projeto, essas avaliações analisam o que funcionou e o que pode ser melhorado, ajudando a identificar lições aprendidas e a aprimorar processos para futuros projetos.

Atividade 2

Conceitos de Garantia de Qualidade de Software

1. Plano de Fundo

Contexto histórico e motivação para a garantia da qualidade de software, explicando a importância das práticas de qualidade na evolução da indústria de software.

2. Elementos de Garantia da Qualidade de Software

- Processos e Procedimentos: Métodos e técnicas para assegurar que o software atende aos padrões de qualidade.
- Ferramentas: Software para automação e monitoramento da qualidade.
- Pessoas: Equipes responsáveis pela implementação das práticas de qualidade.
- Documentação: Planos, relatórios e procedimentos de controle de qualidade.

3. Processos da Software Quality Assurance (SQA) e Características do Produto

- Processos da SQA: Atividades para garantir que o software esteja conforme os requisitos, como planejamento e execução de testes.
- Características do Produto: Qualidades desejadas, como funcionalidade, confiabilidade e usabilidade.

4. Tarefas, Metas e Métricas de SQA

- Tarefas da SQA: Criação de planos de testes, execução de testes, e revisão de requisitos.

- Metas, Atributos e Métricas:
 - Metas: Objetivos de qualidade a serem alcançados.
 - Atributos: Qualidades específicas do software.
 - Métricas: Medidas quantitativas para avaliar a qualidade, como densidade de defeitos.

5. Abordagens Formais da SQA

Métodos rigorosos para garantir qualidade, incluindo análise e modelagem formal, revisões e inspeções sistemáticas.

6. Estatística da Garantida da Qualidade de Software

- Exemplo Genérico: Uso de técnicas estatísticas para analisar dados de qualidade.
- Seis Sigma: Metodologia que visa reduzir defeitos e variabilidade no software.

7. Confiabilidade de Software

- Medidas de Confiabilidade e Disponibilidade: Taxa de falhas e tempo de operação do software.
- Segurança de Software: Proteção contra ataques e vulnerabilidades.

8. Os Padrões de Qualidade ISO 9000

Normas internacionais que definem requisitos para sistemas de gestão da qualidade, garantindo que produtos atendam às expectativas dos clientes.

9. Plano de SQA

Documento que define a abordagem, atividades e recursos necessários para garantir a qualidade do software, incluindo objetivos, métodos e cronograma.