Resumo Detalhado sobre Qualidade de Software

A qualidade de software é uma disciplina essencial dentro da Engenharia de Software, utilizando métodos, técnicas e ferramentas para garantir um controle de qualidade contínuo em produtos e processos. Seu objetivo vai além da simples correção de erros no código, abrangendo desde a análise de requisitos até a implantação e manutenção do software, com o intuito de satisfazer as necessidades dos clientes e alcançar a excelência.

1. Princípios e História da Qualidade

A área de qualidade tem suas raízes no setor industrial, com pioneiros como **Joseph M. Juran** e **William E. Deming**. Juran definiu qualidade como "apto a ser usado", enquanto Deming focava na "conformidade com as exigências". A norma ISO 8402 define qualidade como a capacidade de uma empresa satisfazer as necessidades explícitas e implícitas dos clientes. A busca pela qualidade permite que as organizações aumentem a satisfação do cliente, reduzam erros, desperdícios e custos, e melhorem sua posição no mercado.

A gestão da qualidade, conforme a Trilogia de Juran, é composta por três processos gerenciais:

- **Planejamento da qualidade:** Estabelecimento de metas e processos para atender às necessidades dos clientes.
- **Controle de qualidade:** Medição e comparação do desempenho real com as metas estabelecidas.
- **Melhoria da qualidade:** Diagnóstico de causas de problemas e estabelecimento de controles para manter os ganhos.

2. Evolução da Qualidade de Software

A Engenharia de Software é uma disciplina jovem, que teve seu início entre as décadas de 1950 e 1970. Nos anos 70, surgiu a chamada **"crise do software"**, caracterizada por projetos que extrapolavam custos e prazos, além de apresentarem baixa qualidade e difícil manutenção. Um exemplo clássico foi a falha no software do foguete Ariane 5, que custou bilhões de dólares.

Para superar esses desafios, a Engenharia de Software buscou inspiração em outras áreas, adaptando conceitos como Kanban e Seis Sigma. Esse movimento levou ao surgimento de normas e modelos como a família ISO 9000, CMMI e MPS.BR, que trouxeram mais maturidade e padrões para o desenvolvimento de software.

Organizações como o **IEEE** (Institute of Electrical and Electronics Engineering) e a **ACM** (Association for Computing Machinery) foram fundamentais nesse processo, colaborando na criação do **Swebok**, um guia de melhores práticas para a área.

3. Gestão da Garantia da Qualidade de Software (SQA)

A Gestão da Garantia da Qualidade de Software (SQA) é um campo que envolve diversas siglas e processos para assegurar a qualidade. O Gerenciamento de Qualidade de Software (SQM) é composto por três subcategorias principais:

1. **Planejamento da Qualidade de Software (SQP):** Define metas de qualidade, gerencia riscos e estima o esforço necessário para as atividades de qualidade.
2. **Garantia da Qualidade de Software (SQA):** Conjunto de atividades planejadas para garantir que o processo de desenvolvimento e o produto final atendam aos requisitos técnicos.
3. **Controle de Qualidade de Software (SQC):** Atividades que examinam os artefatos do projeto (código, documentos) para verificar a conformidade com os padrões estabelecidos.

A **Melhoria de Processo de Software (SPI)** é uma subcategoria que visa aprimorar a eficácia e a eficiência dos processos de desenvolvimento, partindo do princípio de que um processo bem definido impacta positivamente a qualidade do software.

4. "Bugs": Erros, Defeitos e Falhas

No contexto de software, os termos "bug", erro, defeito e falha possuem significados distintos:

- **Engano:** Uma ação humana acidental que introduz um problema no código.
- **Defeito:** A consequência do engano no código, resultando em saídas inesperadas.
- **Erro:** O resultado incorreto gerado pelo software devido a um defeito.
- **Falha:** A consequência visível de um erro, como a interrupção da capacidade do software de executar uma função.

Os erros podem ter diversas origens, como requisitos incorretos, falhas de comunicação, desvios de requisitos, erros de projeto lógico, de codificação, e documentação inadequada.

5. Verificação e Validação (V&V)

Verificação e validação (V&V) são atividades centrais na garantia da qualidade. Embora pareçam similares, possuem finalidades distintas e complementares:

- **Verificação:** Garante que o software está sendo desenvolvido da forma correta, ou seja, implementando as funcionalidades conforme especificado. Inclui atividades como *code review*, testes unitários e de integração.
- **Validação:** Assegura que o software desenvolvido é o produto certo, ou seja, atende às necessidades reais do cliente. Um exemplo clássico é o teste de aceitação.

O Modelo V ilustra a relação entre essas duas fases, onde a verificação ocorre durante as etapas de projeto e implementação, e a validação ocorre nas etapas de teste.

6. Modelos e Normas de Qualidade

Para padronizar e melhorar os processos, existem diversos modelos e normas:

- **ISO/IEC/IEEE 12207:** Estabelece uma estrutura comum para os processos do ciclo de vida do software, abrangendo desde a aquisição até a manutenção.
- **ISO/IEC 25010:** Define um modelo de qualidade de produto de software, com oito características principais: funcionalidade, confiabilidade, usabilidade, eficiência de desempenho, manutenibilidade, portabilidade, segurança e compatibilidade.
- **CMMI (Capability Maturity Model Integration):** Um modelo de maturidade que ajuda as organizações a melhorar seus processos de desenvolvimento através de cinco níveis, do "Incompleto" ao "Otimizado".
- **IEEE 1028:** Descreve procedimentos para a execução de diferentes tipos de revisões e auditorias de software.

7. Revisões, Auditorias e Usabilidade

- **Revisões e Inspeções:** Atividades que analisam artefatos como requisitos, código e planos de teste para encontrar erros e garantir a conformidade com os padrões.
- **Auditoria de Software:** Um processo mais formal que verifica se o projeto está em conformidade com normas, políticas e procedimentos estabelecidos.
- **Testes de Usabilidade:** Técnica de avaliação que envolve usuários reais para verificar a eficácia, eficiência e satisfação no uso de um software. É um aspecto crucial da qualidade, pois afeta diretamente a experiência do usuário.

Em suma, a garantia da qualidade de software é um processo contínuo e multifacetado, que exige o envolvimento de toda a equipe de desenvolvimento. A adoção de boas práticas, modelos de maturidade e normas internacionais é fundamental para criar softwares que não apenas funcionem corretamente, mas que também encantem e satisfaçam os usuários finais.