Felipe dos Santos Gamboa - 821137220

Atividade 02 - Garantia de Qualidade de Software

A **Garantia de Qualidade de Software** é um segmento dentro da engenharia de software que se concentra em garantir que o processo de desenvolvimento e o produto final atendam aos requisitos de qualidade estabelecidos. Ela abrange uma série de atividades e metodologias que são aplicadas ao longo do ciclo de vida do software, com o objetivo de prevenir defeitos, assegurar a conformidade com os requisitos e melhorar continuamente os processos de desenvolvimento.

Definição e Objetivos

A garantia de qualidade é definida como um conjunto de atividades organizadas para monitorar e melhorar os processos envolvidos na criação de software. Seu principal objetivo é garantir que o software desenvolvido seja confiável, seguro, eficiente e que atenda tanto aos requisitos funcionais quanto não funcionais estabelecidos. Isso envolve a adoção de práticas que minimizem a ocorrência de defeitos, a realização de revisões periódicas, testes e a implementação de melhorias contínuas nos processos de desenvolvimento.

Processos de Garantia de Qualidade

A Garantia de Qualidade de Software é frequentemente integrada em metodologias de desenvolvimento como o Waterfall, Agile, Scrum, entre outras. Independentemente da metodologia, os processos geralmente incluem as seguintes etapas:

1. **Planejamento da Qualidade**: Definição de padrões, métodos e métricas que serão utilizados para assegurar a qualidade do software. Envolvendo o estabelecimento de critérios de aceitação, normas a serem seguidas e ferramentas a serem utilizadas. Além de verificar prazos, requisitos de pessoas, orçamentos e etc. Essa etapa é onde se concentra grande parte do sucesso do software ao longo de seu desenvolvimento, um bom planejamento da qualidade aumenta muito as chances do projeto dar certo.
2. **Análise e Revisão de Requisitos**: Verificar se todos os requisitos e necessidades ficaram claras para todas as pessoas do time e também envolvidas no projeto indiretamente, é o momento de tirar dúvidas e não deixar pontas soltas. Revisões e inspeções são realizadas para identificar possíveis ambiguidades ou erros na fase inicial do desenvolvimento, evitando problemas durante fases mais complexas do projeto.
3. **Revisões de Design e Código**: Revisões formais de design e código ajudam a identificar erros antes que se estendam para fases mais avançadas do ciclo de vida do software. Isso inclui revisões por pares e inspeções de código, que são fundamentais para assegurar que o design e a implementação estejam de acordo com os requisitos e padrões estabelecidos. Além de alinhar todos os integrantes com as necessidades que o software precisa.

4. **Testes de Software**: Os testes são uma parte crítica, abrangendo testes unitários, de integração, de sistema e de aceitação. Eles têm como objetivo detectar defeitos no software e garantir que ele funcione conforme esperado em diferentes cenários e condições.
5. **Revisões de Processo**: Auditorias internas e externas são realizadas para assegurar que os processos de desenvolvimento estão sendo seguidos conforme planejado e que eles são eficazes na produção de software de qualidade.
6. **Métricas e Análise de Qualidade**: O uso de métricas ajuda a quantificar a qualidade do software e a eficácia dos processos de desenvolvimento. Métricas comuns incluem densidade de defeitos, tempo médio para correção de defeitos, cobertura de teste, entre outras. Essas métricas auxiliam na análise de comportamento da equipe, vendo onde precisa ser melhorado, onde há um “excesso de atenção”, conseguindo assim direcionar para qual lado deve-se focar.
7. **Melhoria Contínua**: Também envolve a análise contínua dos processos e a implementação de melhorias baseadas em lições aprendidas, novas práticas de mercado e feedback de usuários. Modelos como o CMMI (Capability Maturity Model Integration) são frequentemente utilizados para guiar essa melhoria contínua. Isso também pode envolver o pós-lançamento, com atualizações e melhorias de problemas que não foram identificados durante o desenvolvimento (segurança, interface, disponibilidade e etc).

Importância

A adoção de práticas de gerenciamento mais “robustas” e a “rédea curta” é crucial para garantir que o software seja desenvolvido com alta qualidade, minimizando riscos de falhas e aumentando a satisfação dos usuários. Não apenas contribui para a criação de produtos confiáveis e seguros, mas também pode reduzir custos a longo prazo, evitando retrabalho e correções pós-implementação (patches, atualizações e etc).

Além disso, com a crescente complexidade dos sistemas de software e a demanda por ciclos de desenvolvimento mais rápidos, esses métodos ajudam a assegurar que a qualidade não seja comprometida, mesmo em ambientes de desenvolvimento ágeis e dinâmicos. Que, muitas vezes, exigem muita produção em um curto período de tempo, e frases como “Não precisa passar por QA” ou “Testa em produção” se tornam comúns durante o projeto.

Referências:

https://www.tecnisys.com.br/garantia-da-qualidade-de-software-sqa/
https://homepages.dcc.ufmg.br/~rodolfo/dcc823-2-07/Entrega4/Damazio4.pdf
https://ciriusquality.com.br/glossario/o-que-e-sqa-garantia-da-qualidade-de-software/
https://www.devmedia.com.br/introducao-a-garantia-de-qualidade-de-software-e-ferramentas-para-teste/28027