



Gestão e Qualidade de Software

Sistemas de Informação

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

Conceitos de Garantia de Qualidade de Software (QA)

A Garantia de Qualidade de Software (Quality Assurance - QA) é um conjunto de processos, metodologias e práticas destinadas a garantir que o software desenvolvido atenda aos requisitos especificados e funcione de maneira confiável e eficiente. O objetivo principal do QA é prevenir defeitos durante o desenvolvimento do software, em vez de apenas identificá-los após a implementação.

A Garantia de Qualidade de Software (Software Quality Assurance - SQA) consiste em um conjunto de atividades técnicas aplicadas durante todo o processo de desenvolvimento. Seu principal objetivo é garantir que tanto o processo de desenvolvimento quanto o produto de software atinjam os níveis de qualidade especificados. Já o Controle da Qualidade de Software (Software Quality Control - SQC) refere-se à aplicação de processos de qualidade voltados para a identificação e eliminação de produtos que não atingiram o nível de qualidade exigido. Enquanto o SQA foca na prevenção de defeitos ao longo do desenvolvimento, o SQC atua na detecção e correção de falhas nos produtos finais.

Embora muitas vezes usados como sinônimos, há uma diferença fundamental entre os conceitos de teste de software e garantia de qualidade. A Garantia de Qualidade (QA) concentra-se nos processos de desenvolvimento para evitar defeitos e melhorar a qualidade do software desde o início, enquanto o Teste de Software foca na identificação de defeitos através da execução do software.

A Garantia de Qualidade está integrada a todas as fases do Ciclo de Vida do Desenvolvimento de Software (Software Development Life Cycle - SDLC). Na fase de requisitos, realiza-se a revisão e validação para evitar ambiguidades. No design, avalia-se a arquitetura para garantir escalabilidade e segurança. Durante o desenvolvimento, são aplicadas práticas como revisão de código e integração contínua. Nos testes, executam-se testes automatizados e manuais. Por fim, na implantação e manutenção, há o monitoramento contínuo para garantir a estabilidade do software.

Segundo Pressman (2016), a garantia de qualidade deve estar presente em todas as etapas do desenvolvimento de software e deve ser implementada por meio de auditorias, revisões técnicas formais e processos de melhoria contínua. O autor destaca que a qualidade de software não pode ser um elemento isolado, mas deve ser incorporada à cultura organizacional, onde cada membro da equipe tem um papel fundamental na manutenção dos padrões estabelecidos. Além disso, Pressman enfatiza a importância do uso de padrões como ISO 9001 e CMMI (Capability Maturity Model Integration) para estruturar e aperfeiçoar os processos de qualidade dentro das organizações.

Glenford Myers (2011) também contribui para o entendimento da qualidade de software ao enfatizar a importância dos testes como um meio essencial para a detecção de defeitos e falhas no software. Segundo Myers, o objetivo do teste de software não é apenas provar que um sistema funciona corretamente, mas sim revelar falhas que podem comprometer sua confiabilidade. O autor destaca ainda que os testes devem ser planejados e conduzidos de forma sistemática, utilizando técnicas como testes de caixa-preta e caixa-branca para abranger diferentes aspectos da qualidade do software. Além disso, Myers defende a automação de testes como um recurso essencial para aumentar a eficiência e reduzir erros humanos.

Diversos tipos de testes são utilizados para garantir a qualidade do software. Os testes funcionais validam se o software atende aos requisitos especificados, enquanto os testes não funcionais avaliam aspectos como desempenho, usabilidade e segurança. Os testes automatizados utilizam scripts para reduzir o tempo e erros humanos nos testes. Os testes de regressão garantem que novas implementações não afetem funcionalidades existentes. Já os testes de unidade testam pequenos blocos do código individualmente, e os testes de integração avaliam a interação entre diferentes módulos do sistema.

Existem várias metodologias e boas práticas aplicadas ao QA. A metodologia ágil utiliza ciclos iterativos e incrementais para entregar qualidade de forma contínua. O DevOps e CI/CD promovem integração e entrega contínuas para automação de testes e lançamento de versões confiáveis. A revisão de código é uma prática colaborativa para garantir código limpo e eficiente. Além disso, métricas de qualidade, como taxa de falhas, cobertura de testes e tempo médio para correção, são utilizadas para avaliar e melhorar a qualidade do software.

A Garantia de Qualidade de Software é um processo essencial para garantir um produto final confiável, seguro e eficiente. A adoção de boas práticas, automação e metodologias ágeis melhora a qualidade do software e reduz custos com correções tardias. Incorporar QA ao longo de todo o ciclo de vida do desenvolvimento é fundamental para empresas que buscam excelência em seus produtos tecnológicos.

Bibliografia:

PRESSMAN, Roger S.; MAXIM, Bruce R. *Engenharia de Software: Uma Abordagem Profissional*. 8. ed. São Paulo: McGraw-Hill, 2016;

MYERS, Glenford J. *The Art of Software Testing*. 3. ed. Hoboken: Wiley, 2011.