

Unochapecó

Vinicius Fortunati

Trabalho Engenharia de Software

**CHAPECÓ
2023**

Crise de Software

Crise de software é um fenômeno que se manifesta em diferentes níveis e contextos da indústria de desenvolvimento de software. O termo foi criado na década de 1960, quando os primeiros projetos de software começaram a encontrar dificuldades significativas para atender aos requisitos de qualidade, prazo e orçamento.

Uma das principais causas da crise do software é o aumento da complexidade de sistemas cada vez mais complexos e interligados. Contribuindo ainda mais para o problema está a falta de padronização na indústria, a ausência de métodos de desenvolvimento eficientes e a dependência de tecnologias que rapidamente se tornam obsoletas.

Outra causa significativa da crise do software é a falta de habilidades e conhecimentos específicos por parte dos desenvolvedores. Os programadores geralmente carecem de treinamento adequado em engenharia de software e não estão familiarizados com técnicas de desenvolvimento mais avançadas.

Várias abordagens foram propostas nas últimas décadas para lidar com a crise do software. Um dos mais difundidos é o desenvolvimento ágil, que enfatiza a colaboração entre os membros da equipe, a entrega incremental de funcionalidade e a flexibilidade para lidar com requisitos em constante mudança.

Outra abordagem que ganhou destaque recentemente é a adoção de práticas de DevOps, que buscam integrar equipes de desenvolvimento e operações para otimizar o processo de desenvolvimento e entrega de software. Isso inclui o uso de ferramentas de automação, implementação de testes automatizados e adoção de uma cultura de melhoria contínua.

Resumindo, a crise do software é um problema complexo e multifacetado que exige ações coordenadas de empresas, governos, universidades e demais atores da indústria. Adotar abordagens mais ágeis e colaborativas aliadas ao uso de tecnologias avançadas e treinamento adequado para desenvolvedores pode ajudar a superar esse problema e melhorar a qualidade e eficiência do software produzido.

Requisitos Funcionais

Os requisitos funcionais referem-se às funções específicas que o sistema ou software deve executar. Esses requisitos descrevem o que o software deve fazer e como deve fazê-lo, ou seja, suas funções. Por exemplo, um requisito funcional pode ser a capacidade de um sistema de vendas online permitir que os usuários façam uma compra segura.

Requisitos não Funcionais

Os requisitos não funcionais referem-se a recursos do software que não estão diretamente relacionados à sua funcionalidade. Esses requisitos descrevem como o software deve funcionar em termos de desempenho, usabilidade, segurança, confiabilidade, capacidade de manutenção e escalabilidade. Por exemplo, um requisito não funcional pode ser a capacidade de um sistema de vendas online lidar com milhares de transações simultâneas.

Requisitos de sistema

Os requisitos de sistema são uma parte fundamental do processo de engenharia de software, pois eles ajudam a definir os parâmetros técnicos e operacionais que um software deve atender para ser considerado viável e funcional em determinado ambiente. Esses requisitos podem ser divididos em duas categorias: requisitos funcionais e requisitos não funcionais.

Os requisitos funcionais referem-se às funcionalidades específicas que o software deve ter, incluindo tarefas que ele deve ser capaz de realizar e dados que ele deve ser capaz de manipular. Por exemplo, um requisito funcional pode ser a capacidade de um sistema de vendas online permitir que os usuários realizem uma compra com segurança e facilidade.

Já os requisitos não funcionais referem-se a aspectos como desempenho, usabilidade, segurança, confiabilidade, manutenção e escalabilidade. Esses requisitos descrevem como o software deve se comportar em termos de desempenho, como a velocidade de processamento e o tempo de resposta, bem como a facilidade de uso para o usuário final.

Além disso, os requisitos de sistema também incluem aspectos técnicos, como a compatibilidade do software com o hardware e o software do sistema operacional

em que ele será executado. Outros requisitos incluem a integração com outros sistemas, como bancos de dados e sistemas de segurança.

É importante que os requisitos de sistema sejam claramente definidos e documentados desde o início do processo de desenvolvimento do software. Isso ajuda a garantir que todos os envolvidos no projeto tenham uma compreensão clara do que deve ser feito e que o software final atenda às expectativas dos usuários e das empresas. Além disso, os requisitos de sistema devem ser atualizados e revisados regularmente ao longo do ciclo de vida do software para garantir que ele continue atendendo às necessidades dos usuários.