

Crise de Software

Uma crise de software é um problema que ocorre quando os métodos e técnicas utilizados para o desenvolvimento de software não conseguem lidar efetivamente com a complexidade e escala crescentes dos projetos de software.

A crise de software foi reconhecida na década de 1960, quando o crescimento exponencial da complexidade dos sistemas de software começou a sobrecarregar os métodos tradicionais de desenvolvimento.

As principais causas da crise de software incluem a falta de padronização e qualidade no processo de desenvolvimento de software, a falta de compreensão adequada das necessidades dos usuários e a falta de colaboração efetiva entre os membros da equipe de desenvolvimento. Além disso, a complexidade crescente dos sistemas de software também contribui para a crise, tornando o desenvolvimento mais difícil e aumentando o risco de erros e problemas de desempenho.

Para enfrentar a crise de software, os engenheiros têm desenvolvido novas metodologias, como a engenharia de software ágil, que enfatiza a colaboração e a interação constante entre os membros da equipe de desenvolvimento, bem como a melhoria contínua do processo de desenvolvimento. Além disso, novas tecnologias e ferramentas, como as linguagens de programação de alto nível e os ambientes de desenvolvimento integrados, também ajudaram a tornar o processo de desenvolvimento mais eficiente e efetivo.

Requisitos Funcionais

Requisitos funcionais são descrições das funcionalidades e comportamentos que um sistema de software deve possuir para atender às necessidades dos usuários e das partes interessadas. Eles definem o escopo e as expectativas do sistema.

São aqueles que são desenvolvidos utilizando técnicas de engenharia de software, como a análise de requisitos, a modelagem de processos de negócios e a prototipagem.

Existem vários tipos de requisitos funcionais baseados em engenharia de software, segue abaixo dois exemplos:

1. Requisitos de processo: descrevem como o sistema deve se comportar em relação às entradas, processamento e saídas.
2. Requisitos de interface: descrevem como o sistema deve interagir com outros sistemas e usuários.

Os requisitos funcionais são importantes porque eles definem as expectativas do sistema e ajudam a garantir que o software atenda às necessidades e expectativas dos usuários e partes interessadas. Eles também ajudam a orientar o processo de desenvolvimento, fornecendo uma base sólida para a tomada de decisões de design e desenvolvimento.

Requisitos Não Funcionais

Os requisitos não funcionais são aqueles que não estão relacionados diretamente com a funcionalidade do sistema, mas sim com as suas características, como desempenho, usabilidade, segurança, confiabilidade, interoperabilidade, entre outros. Segue exemplos abaixo:

Aluno: Yuri Steffani dos Santos

Data: 05/04/23

1. **Manutenibilidade:** este requisito se refere à facilidade de manutenção do sistema, incluindo correção de erros, atualização de versões e implementação de novas funcionalidades. A manutenibilidade pode ser medida através de métricas como tempo médio para correção de erros, tempo médio para implementação de novas funcionalidades, entre outros.
2. **Portabilidade:** este requisito se refere à capacidade do sistema de ser executado em diferentes plataformas. A portabilidade pode ser medida através de métricas como número de plataformas suportadas, tempo médio para portabilidade entre plataformas, entre outros.

Requisitos de Sistema

Os requisitos de sistema são as especificações detalhadas das funcionalidades e comportamentos esperados do sistema a ser desenvolvido. Eles são elaborados com base nas necessidades dos usuários e das partes interessadas e servem como guia para o processo de desenvolvimento de software. A seguir, estão dois exemplos:

1. **Requisitos funcionais:** estes requisitos definem as funcionalidades e comportamentos esperados do sistema. Eles incluem casos de uso, cenários de uso, fluxos de trabalho, entre outros.
2. **Requisitos de desempenho:** estes requisitos definem o desempenho esperado do sistema em termos de tempo de resposta, tempo de processamento, uso de recursos, entre outros.

Os requisitos de sistema são essenciais para o sucesso do desenvolvimento de software, pois garantem que o sistema desenvolvido atenda às expectativas e necessidades dos usuários e das partes interessadas. Eles devem ser cuidadosamente elaborados, documentados e gerenciados ao longo de todo o ciclo de vida do software.