

NOME: Jandiele dos santos
PROFESSOR: Radames Pereira
DICIPLINA: Engenharia de software 1

Crise de software

O Crash do software de 1970 foi um momento crucial na história do desenvolvimento de software. Na época, o software estava se tornando cada vez mais importante para muitas organizações, mas a qualidade e a confiabilidade do software geralmente eram baixas. Isso levou a uma série de falhas de software notáveis, incluindo a queda do sistema de reservas da American Airlines em 1965 e a falha do sistema de controle de tráfego aéreo do país em 1967. Essas falhas de software atraíram a atenção do governo, da indústria e da academia, e muitas pessoas começaram a estudar o problema. Um relatório digno de nota é o "Relatório de Crise de Software" de 1970 emitido pelo Departamento de Defesa dos EUA. O relatório destaca vários problemas comuns no desenvolvimento de software, incluindo falta de padronização, baixa qualidade de código, falta de documentação adequada e gerenciamento de projeto ruim.

Para resolver esses problemas, muitas soluções foram propostas, incluindo o uso de metodologias de desenvolvimento de software mais rigorosas, como o desenvolvimento em cascata, e a criação de padrões e ferramentas de desenvolvimento. Essas soluções ajudam a melhorar a qualidade e a confiabilidade do software e continuam a influenciar a forma como o software é desenvolvido atualmente. No geral, a crise de software da década de 1970 foi um momento importante na história do desenvolvimento de software que ajudou a aumentar a conscientização sobre a importância da qualidade e confiabilidade do software e incentivou o desenvolvimento de métodos e ferramentas de desenvolvimento mais rigorosos.

Os requisitos funcionais são: uma parte essencial do processo de desenvolvimento de software e, portanto, também desempenham um papel importante na prevenção e gestão de crises de software. Os requisitos funcionais descrevem as funcionalidades que um sistema deve ter para atender às necessidades dos usuários e do negócio.

Quando os requisitos funcionais não são adequadamente definidos ou implementados, pode ocorrer uma crise de software. Por exemplo, se o sistema não tiver a funcionalidade correta ou se não puder ser usado facilmente pelos usuários, pode haver problemas de produtividade ou de usabilidade, levando a perdas financeiras e de reputação.

Além disso, é importante que os requisitos funcionais sejam continuamente monitorados e atualizados ao longo do ciclo de vida do software, para garantir que o sistema continue atendendo às necessidades dos usuários e do negócio e para evitar problemas de crise de software no futuro.

Os requisitos não funcionais também desempenham um papel importante na prevenção e gestão de crises de software. Os requisitos não funcionais são aqueles que não estão relacionados às funcionalidades específicas do sistema, mas que definem as características e qualidades do sistema em termos de desempenho, confiabilidade, segurança, usabilidade, entre outros.

Para evitar esses problemas, é importante que os requisitos não funcionais sejam bem definidos e testados antes da implementação. Isso inclui a definição de padrões de desempenho, segurança e usabilidade que o sistema deve atender e a realização de testes para garantir que esses padrões sejam cumpridos.

Em resumo, os requisitos não funcionais são importantes para garantir que o sistema tenha um desempenho adequado, seja seguro e fácil de usar. Definir e monitorar esses requisitos pode ajudar a prevenir crises de software.

Os requisitos de sistemas: são outro aspecto importante na prevenção e gestão de crises de software. Os requisitos de sistemas definem as necessidades e expectativas do sistema em termos de hardware, software e infraestrutura.

Quando os requisitos de sistemas não são adequadamente definidos ou implementados, podem ocorrer problemas de incompatibilidade, escalabilidade, desempenho e disponibilidade, levando a uma crise de software. Por exemplo, se um sistema não for projetado para suportar um grande volume de usuários simultâneos, o sistema pode falhar quando muitos usuários tentarem acessá-lo ao mesmo tempo.

É importante que os requisitos de sistemas sejam bem definidos e testados antes da implementação para evitar problemas. Isso inclui a definição dos requisitos de hardware, software e infraestrutura que o sistema deve atender, bem como a realização de testes para garantir que o sistema funcione de forma adequada em diferentes ambientes e configurações.

