

Universidade Comunitária da Região de Chapecó

Unochapecó

Leonardo E. Zuchi

Crise do Software

Chapecó – Santa Catarina

2023

Crise do Software

A Crise do software foi um termo que surgiu nos anos 70, o termo expressava as dificuldades do desenvolvimento de software frente ao rápido crescimento da demanda por software, da complexidade dos problemas a serem resolvidos e da inexistência de técnicas estabelecidas para o desenvolvimento de sistemas que funcionassem adequadamente ou pudessem ser validados. No Início dos anos 70, quando vivia-se a terceira era do software, houveram muitos problemas de prazo e custo no desenvolvimento de software, devido a baixa produtividade, baixa qualidade e difícil manutenção do software. Grande parte dos projetos continuam com estes problemas ainda na atualidade, assim pode-se dizer que a crise continua vigente.

Os problemas mais comuns no desenvolvimento de software.

- Estimativas de prazo e de custo imprecisas .
- Produtividade das pessoas da área de software não acompanha a demanda.
- Prazos ultrapassados.
- Custos acima do previsto.
- A facilidade de manutenção não era enfatizada como um critério importante, gerando assim custos de manutenção elevados.
- Não atendimento dos requisitos do usuário.
- 1/3 dos projetos eram cancelados.
- 2/3 dos projetos extrapolavam o orçamento.

Custo hardware x software

- 1970 = 8:2
- 1991 = 2:8
- Dias Atuais = 1:9

Exemplo de Falha no Desenvolvimento do Software

- Ariane 5 – Projeto Espacial da agência Européia

Solução para a crise do software

- Utilização de técnicas, ferramentas e processos sistematizados para produzir software.
- Treinamento e educação em conjunto com a mudança de paradigma sobre o que é desenvolver software e como deveria ser feito.
- Criação da Engenharia de Software.

A criação da Engenharia de Software surgiu numa tentativa de contornar a crise do software e dar um tratamento de engenharia(mais sistemático e controlado) ao desenvolvimento de sistemas de software complexos. O termo Engenharia de Software tornou-se conhecido após uma conferência em

1968, quando as dificuldades e armadilhas de projetar sistemas complexos foram discutidas francamente. A busca de soluções começou. Ela se concentrou em melhores metodologias e ferramentas. As mais importantes foram as linguagens de programação que refletem os estilos procedimental, modular e, em seguida, orientado a objeto. A engenharia de software está intimamente ligada ao aparecimento e aperfeiçoamento desses estilos. Também importantes foram os esforços de sistematização, automatização da documentação do programa e testes. Por último, a verificação analítica e provas de correção deveriam substituir os testes. só após a conferência as dificuldades foram discutidas abertamente e confessadas com franqueza incomum, e os termos "engenharia de software" e "crise de software" foram criados.

Referências

1. PRESSMAN, Roger. Software Engineering: A Practitioner's Approach, 6ª edição, Mc Graw Hill, 2005 .
2. «Ariane-5G». Gunter's Space Page. Consultado em 6 September 2014.
3. Naur and B. Randell, Eds. Software Engineering. Report on a Conference held in Garmisch, Oct. 1968, sponsored by NATO
4. <https://ariane.cnes.fr/en/5-goodies/decollage-parfait-pour-ariane-5>

Universidade Comunitária da Região de Chapecó

Unochapecó

Leonardo E. Zuchi

Engenharia de Requisitos

Chapecó – Santa Catarina

2023

Requisitos de Sistema

Os requisitos de um sistema são as descrições do que o sistema deve fazer, os serviços oferecem e as restrições a seu funcionamento. Esses requisitos refletem as necessidades dos clientes para um sistema que serve a uma finalidade determinada, como controlar um dispositivo, colocar um pedido ou encontrar informações.

O processo de descobrir, analisar, documentar e verificar esses serviços e restrições é chamado engenharia de requisitos (RE, do inglês requirements engineering). O termo 'requisito' não é usado de forma consistente pela indústria de software. Em alguns casos, o requisito é apenas uma declaração abstrata em alto nível de um serviço que o sistema deve oferecer ou uma restrição a um sistema. No outro extremo, é uma definição detalhada e formal de uma função do sistema.

Requisitos Funcionais

Os requisitos funcionais de um sistema descrevem o que ele deve fazer. Eles dependem do tipo de software a ser desenvolvido, de quem são seus possíveis usuários e da abordagem geral adotada pela organização ao escrever os requisitos. Quando expressos como requisitos de usuário, os requisitos funcionais são normalmente descritos de forma abstrata, para serem compreendidos pelos usuários do sistema. No entanto, requisitos de sistema funcionais mais específicos descrevem em detalhes as funções do sistema, suas entradas e saídas, exceções etc. Requisitos funcionais do sistema variam de requisitos gerais, que abrangem o que o sistema deve fazer, até requisitos muito específicos, que refletem os sistemas e as formas de trabalho em uma organização.

Em princípio, a especificação dos requisitos funcionais de um sistema deve ser completa e consistente. Completude significa que todos os serviços requeridos pelo usuário devem ser definidos. Consistência significa que os requisitos não devem ter definições contraditórias. Na prática, para sistemas grandes e complexos, é praticamente impossível alcançar completude e consistência dos requisitos. Uma razão para isso é que ao elaborar especificações para sistemas complexos é fácil cometer erros e omissões. Outra razão é que em um sistema de grande porte existem muitos stakeholders. Um stakeholder é uma pessoa ou papel que, de alguma maneira, é afetado pelo sistema. Os stakeholders têm necessidades diferentes — e, muitas vezes, inconsistentes. Essas inconsistências podem não ser evidentes em um primeiro momento, quando os requisitos são especificados, e, assim, são incluídas na especificação. Os problemas podem surgir após uma análise mais profunda ou depois de o sistema ter sido entregue ao cliente.

Requisitos Não Funcionais

Os requisitos não funcionais, como o nome sugere, são requisitos que não estão diretamente relacionados com os serviços específicos oferecidos pelo sistema a seus usuários. Eles podem estar relacionados às propriedades emergentes do sistema, como confiabilidade, tempo de resposta e ocupação de área. Uma alternativa a esse cenário seria os requisitos definirem restrições sobre a implementação do sistema, como as capacidades dos

dispositivos de E/S ou as representações de dados usadas nas interfaces com outros sistemas.

Os requisitos não funcionais, como desempenho, proteção ou disponibilidade, normalmente especificam ou restringem as características do sistema como um todo. Requisitos não funcionais são frequentemente mais críticos que requisitos funcionais individuais. Os usuários do sistema podem, geralmente, encontrar maneiras de contornar uma função do sistema que realmente não atenda a suas necessidades. No entanto, deixar de atender a um requisito não funcional pode significar a inutilização de todo o sistema. Por exemplo, se um sistema de aeronaves não cumprir seus requisitos de confiabilidade, não será certificado como um sistema seguro para operar; se um sistema de controle embutido não atender aos requisitos de desempenho, as funções de controle não funcionarão corretamente.

Embora muitas vezes seja possível identificar quais componentes do sistema implementam requisitos funcionais específicos (por exemplo, pode haver componentes de formatação que implementam requisitos de impressão de relatórios), é frequentemente mais difícil relacionar os componentes com os requisitos não funcionais. A implementação desses requisitos pode ser difundida em todo o sistema. Há duas razões para isso:

1. Requisitos não funcionais podem afetar a arquitetura geral de um sistema em vez de apenas componentes individuais. Por exemplo, para assegurar que sejam cumpridos os requisitos de desempenho, será necessário organizar o sistema para minimizar a comunicação entre os componentes.
2. Um único requisito não funcional, tal como um requisito de proteção, pode gerar uma série de requisitos funcionais relacionados que definam os serviços necessários no novo sistema. Além disso, também podem gerar requisitos que restrinjam requisitos existentes.

Os requisitos não funcionais surgem por meio das necessidades dos usuários, devido a restrições de orçamento, políticas organizacionais, necessidade de interoperabilidade com outros sistemas de software ou hardware, ou a partir de fatores externos, como regulamentos de segurança ou legislações de privacidade.

Referências

- <https://www.facom.ufu.br/~william/Disciplinas%202018-2/BSI-GSI030-EngenhariaSoftware/Livro/engenhariaSoftwareSommerville.pdf>