



FUNDAMENTOS DA ENGENHARIA DE SOFTWARE

ENGENHARIA DE SOFTWARE I
Análise e Desenvolvimento de Sistemas

Prof. Evandro Zatti, M. Eng.

ENGENHARIA DE SOFTWARE

“A Engenharia de Software engloba um processo, uma coleção de métodos (práticas) e uma matriz de ferramentas que permitem que profissionais construam software de computador de alta qualidade.”

(PRESSMAN e MAXIM, 2015, p. 14)

HISTÓRICO

- O termo **Engenharia de Software** foi criado na década de 1960 e utilizado oficialmente em 1968 na *NATO Conference on Software Engineering* (Conferência sobre Engenharia de Software da OTAN – Org. do Tratado do Atlântico Norte).
- Sua criação 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.

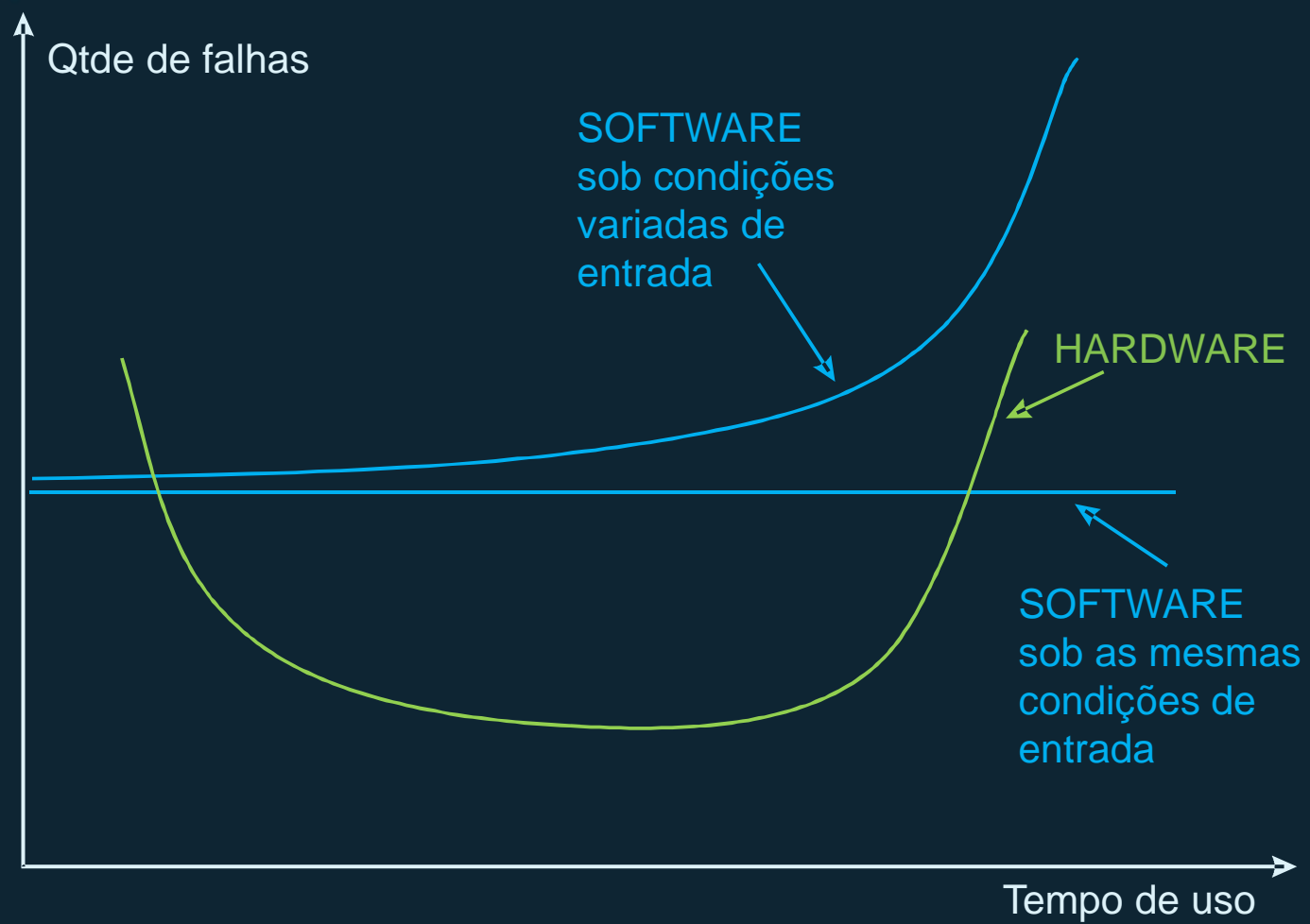
COMPOSIÇÃO DO SOFTWARE

- **Instruções** – que quando executadas produzem a função e o desempenho desejados;
- **Estruturas de Dados** – que possibilitam que os programas manipulem adequadamente a informação;
- **Documentos** – que descrevem a operação e o uso dos programas.

CARACTERÍSTICAS DO SOFTWARE

- Desenvolvido ou projetado por engenharia, não manufaturado no sentido clássico;
- Não se desgasta, mas se deteriora;
- A maioria é feita sob medida em vez de ser montada a partir de componentes existentes.

NATUREZA DO SOFTWARE



fonte: PRESSMAN e MAXIM, 2015
(adaptado)

NATUREZA DO SOFTWARE



fonte: PRESSMAN e MAXIM, 2015
(adaptado)

APLICAÇÕES DE SOFTWARE

- **De sistema:** coleção de programas escritos para dar apoio a outros programas;
- **Comerciais:** sistemas de operações comerciais e tomadas de decisões administrativas;
- **Científicos e de Engenharia:** caracterizados por algoritmos de processamento de números;

APLICAÇÕES DE SOFTWARE

- **Embarcados:** usados para controlar produtos e sistemas para os mercados industriais e de consumo;
- **De linha de produto:** projetados para permitir o uso por diversos consumidores, focados em mercado;
- **Aplicações Mobile e Web:** considerando o uso em rede, vão desde aplicações baseadas em navegador ou aplicações independentes;
- **De Inteligência Artificial:** fazem uso de algoritmos não numéricos para resolver problemas que não sejam favoráveis à computação ou à análise direta.

EVOLUÇÃO DO SOFTWARE

- **1950 – 1965**

- ✓ O hardware sofreu contínuas mudanças;
- ✓ O software era uma arte "secundária" para a qual havia poucos métodos sistemáticos;
- ✓ O hardware era de propósito geral;
- ✓ O software era específico para cada aplicação;
- ✓ Não havia documentação.

EVOLUÇÃO DO SOFTWARE

- **1965 – 1975**

- ✓ Multiprogramação e sistemas multiusuários;
- ✓ Técnicas interativas;
- ✓ Sistemas de tempo real;
- ✓ 1ª geração de SGBDs;
- ✓ Produto de software – *software houses*;
- ✓ Bibliotecas de Software;
- ✓ Cresce número de sistemas baseado em computador;
- ✓ Manutenção quase impossível.

... Crise do Software

EVOLUÇÃO DO SOFTWARE

- **1975 – 2000**

- ✓ Sistemas distribuídos;
- ✓ Redes locais;
- ✓ Uso generalizado de microprocessadores – produtos inteligentes;
- ✓ Hardware de baixo custo;
- ✓ Impacto de consumo;
- ✓ Tecnologias orientadas a objetos;

EVOLUÇÃO DO SOFTWARE

- **2000 – atualmente**

- ✓ Sistemas especialistas e software de inteligência artificial usados na prática;
- ✓ Computação paralela;
- ✓ Computação em nuvem;
- ✓ Aplicações IoT.

CRISE DO SOFTWARE

- Dificuldade em desenvolver programas complexos de forma segura e previsível;
- Dificuldade em cumprir os cronogramas e especificações de grandes projetos de software;
- Termo Engenharia de Software = necessidade de trazer mais Engenharia para o Software.

PROBLEMAS ASSOCIADOS À CRISE DO SOFTWARE

- As estimativas de prazo e de custo frequentemente são imprecisas:
 - ✓ “Não dedicamos tempo para coletar dados sobre o processo de desenvolvimento de software”;
 - ✓ “Sem nenhuma indicação sólida de produtividade, não podemos avaliar com precisão a eficácia de novas ferramentas, métodos ou padrões”.

PROBLEMAS ASSOCIADOS À CRISE DO SOFTWARE

- A produtividade das pessoas da área de software não tem acompanhado a demanda por seus serviços:
 - ✓ “Os projetos de desenvolvimento de software normalmente são efetuados apenas com um vago indício das exigências do cliente”.

PROBLEMAS ASSOCIADOS À CRISE DO SOFTWARE

- A qualidade de software às vezes é menos que adequada
 - ✓ Só recentemente começam a surgir conceitos quantitativos sólidos de garantia de qualidade de software.
- O software existente é muito difícil de manter:
 - ✓ A tarefa de manutenção devora o orçamento destinado ao software;
 - ✓ A facilidade de manutenção não foi enfatizada como um critério importante.

PROBLEMAS ASSOCIADOS À CRISE DO SOFTWARE

- Estimativas de prazo e de custo ↑
- Produtividade das pessoas ↓
- Qualidade do software ↓
- Dificuldade de manter o software ↑

CAUSAS DOS PROBLEMAS ASSOCIADOS À CRISE

- O próprio caráter do software:
 - ✓ O software é um elemento de sistema lógico e não físico;
 - ✓ Consequentemente o sucesso é medido pela qualidade de uma única entidade e não pela qualidade de muitas entidades manufaturadas;
 - ✓ O software não se desgasta, mas se deteriora.

CAUSAS DOS PROBLEMAS ASSOCIADOS À CRISE

- Falhas das pessoas responsáveis pelo desenvolvimento de software:
 - ✓ Gerentes sem nenhum *background* em software;
 - ✓ Os profissionais da área de software recebendo pouco treinamento formal em novas técnicas para o desenvolvimento de software;
 - ✓ Resistência a mudanças.

CAUSAS DOS PROBLEMAS ASSOCIADOS À CRISE

- Mitos do software (propagaram desinformação e confusão):
 - ✓ mitos administrativos;
 - ✓ mitos do cliente;
 - ✓ mitos do profissional.

MITOS DO SOFTWARE

- **Administrativos:**

- ✓ Mito: *“Já temos um manual repleto de padrões e procedimentos para a construção de software. Isso não oferecerá ao meu pessoal tudo o que eles precisam saber?”*
- ✓ Realidade: Será que o manual é usado? Os profissionais sabem que ele existe? Ele reflete a prática moderna de desenvolvimento de software? Ele é completo?

MITOS DO SOFTWARE

- **Administrativos:**

- ✓ Mito: *“Meu pessoal tem ferramentas de desenvolvimento de software de última geração; afinal lhes compramos os mais novos computadores.”*
- ✓ Realidade: É preciso muito mais do que os mais recentes computadores para se fazer um desenvolvimento de software de alta qualidade.

MITOS DO SOFTWARE

- **Administrativos:**

- ✓ Mito: *“Se nós estamos atrasados nos prazos, podemos adicionar mais programadores e tirar o atraso.”*
- ✓ Realidade: O desenvolvimento de software não é um processo mecânico igual à manufatura; acrescentar pessoas em um projeto torna-o ainda mais atrasado; pessoas podem ser acrescentadas, mas somente de uma forma planejada.

MITOS DO SOFTWARE

- **Cliente:**

- ✓ Mito: *“Uma declaração geral dos objetivos é suficiente para se começar a escrever programas – podemos preencher os detalhes mais tarde.”*
- ✓ Realidade: Uma definição inicial ruim é a principal causa de fracassos dos esforços de desenvolvimento de software; é fundamental uma descrição formal e detalhada do domínio da informação, função, desempenho, interfaces, restrições de projeto e critérios de validação.

MITOS DO SOFTWARE

- **Cliente:**

- ✓ Mito: *“Os requisitos de projeto modificam-se continuamente, mas as mudanças podem ser facilmente acomodadas, porque o software é flexível.”*
- ✓ Realidade: Uma mudança, quando solicitada tardiamente num projeto, pode ser maior do que a ordem de magnitude mais dispendiosa da mesma mudança solicitada nas fases iniciais.

MITOS DO SOFTWARE

- **Profissional:**

- ✓ Mito: *“Assim que escrevermos o programa e o colocarmos em funcionamento nosso trabalho estará completo.”*
- ✓ Realidade: Os dados da indústria indicam que entre 50 e 70% de todo esforço gasto num programa serão despendidos depois que ele for entregue pela primeira vez ao cliente.

MITOS DO SOFTWARE

- **Profissional:**

- ✓ Mito: *“Enquanto não tiver o programa ‘funcionando’, eu não terei realmente nenhuma maneira de avaliar sua qualidade.”*
- ✓ Realidade: Um programa funcionando é somente uma parte de uma Configuração de Software que inclui todos os itens de informação produzidos durante a construção e manutenção do software.

PROBLEMAS GERAIS DOS PROJETOS DE SOFTWARE

- Na maioria das vezes, os projetos de software:
 - ✓ Extrapolam o orçamento;
 - ✓ Não respeitam o cronograma;
 - ✓ Não atendem aos requisitos;
 - ✓ São cancelados;
 - ✓ Não satisfazem o usuário.

ATIVIDADES DA ENGENHARIA DE SOFTWARE

- **Definição:** “o que” será desenvolvido.
 - ✓ **Análise do Sistema:** define o papel de cada elemento num sistema baseado em computador, atribuindo em última análise, o papel que o software desempenhará;
 - ✓ **Planejamento do Projeto de Software:** assim que o escopo do software é estabelecido, os riscos são analisados, os recursos são alocados, os custos são estimados e, tarefas e programação de trabalho definidas;

ATIVIDADES DA ENGENHARIA DE SOFTWARE

- ✓ **Análise de Requisitos:** o escopo definido para o software proporciona uma direção, mas uma definição detalhada do domínio da informação e da função do software é necessária antes que o trabalho inicie.

ATIVIDADES DA ENGENHARIA DE SOFTWARE

- **Desenvolvimento** : “como” o software será desenvolvido.
 - ✓ **Projeto de Software**: traduz os requisitos do software num conjunto de representações (gráficas, tabulares ou baseadas em linguagem) que descrevem a estrutura de dados, a arquitetura do software, os procedimentos algorítmicos e as características de interface;

ATIVIDADES DA ENGENHARIA DE SOFTWARE

- ✓ **Codificação:** as representações do projeto devem ser convertidas numa linguagem artificial (a linguagem pode ser uma linguagem de programação convencional ou uma linguagem não procedimental) que resulte em instruções que possam ser executadas pelo computador;
- ✓ **Testes:** logo que o software é implementado numa forma executável por máquina, ele deve ser testado para que se possa descobrir defeitos de função, lógica e implementação.

ATIVIDADES DA ENGENHARIA DE SOFTWARE

- **Manutenção:** concentra-se nas “mudanças” que ocorrerão depois que o software for liberado para uso operacional.
 - ✓ **Correção:** mesmo com as melhores atividades de garantia de qualidade de software, é provável que o cliente descubra defeitos no software. A manutenção corretiva muda o software para corrigir defeitos;

ATIVIDADES DA ENGENHARIA DE SOFTWARE

- ✓ **Adaptação:** com o passar do tempo, o ambiente original (por exemplo a CPU, o sistema operacional e periféricos) para o qual o software foi desenvolvido provavelmente mudará. A manutenção adaptativa muda o software para acomodar mudanças em seu ambiente;
- ✓ **Melhoramento Funcional:** a medida que o software é usado, o cliente/usuário reconhecerá funções adicionais que oferecerão benefícios. A manutenção perfectiva estende o software para além de suas exigências funcionais originais.

ATIVIDADES DA ENGENHARIA DE SOFTWARE

- **Proteção:** as fases e etapas descritas anteriormente são complementadas por uma série de atividades de proteção.
 - ✓ **Revisões:** efetuadas para garantir que a qualidade seja mantida à medida que cada etapa é concluída;

ATIVIDADES DA ENGENHARIA DE SOFTWARE

- ✓ **Documentação:** é desenvolvida e controlada para garantir que informações completas sobre o software estejam disponíveis para uso posterior;
- ✓ **Controle das Mudanças:** é instituído de forma que as mudanças possam ser aprovadas e acompanhadas.



ATIVIDADE PRÁTICA

REFERÊNCIAS

- PRESSMAN, R. W, MAXIM B. R. *Software Engineering - A Practitioner's Approach*. 8th ed. New York: McGraw-Hill, 2015.