LABES-UFPA
Laboratório de Engenharia de Software da UFPA
www.labes.ufpa.br

ENGENHARIA DE SOFTWARE

2 - DEFININDO A ENGENHARIA DE SOFTWARE

Prof. Cleidson de Souza
Profa. Carla Alessandra Lima Reis
Prof. Rodrigo Quites Reis

LABES-UFPA
Laboratório de Engenharia de Software da UFPA
www.labes.ufpa.br

Introdução e Motivação

"*The programs that run on a computer and perform certain functions*"

(Merriam-Webster)

"*Our civilization runs on software*"

Bjarne Stroustrup

"*Software is eating the world*"

Marc Andreessen, um dos autores do Mosaic

"Software Is Eating the World, but AI is Going to eat Software"

Jensen Huang, CEO da NVidia

Para construir uma casa de cachorro...

- Uma pessoa consegue construir
- Precisa de pouco projeto
- O processo é simples
- As ferramentas são simples e fáceis de encontrar

Fonte: Grady Booch. Software Architecture and the UML.

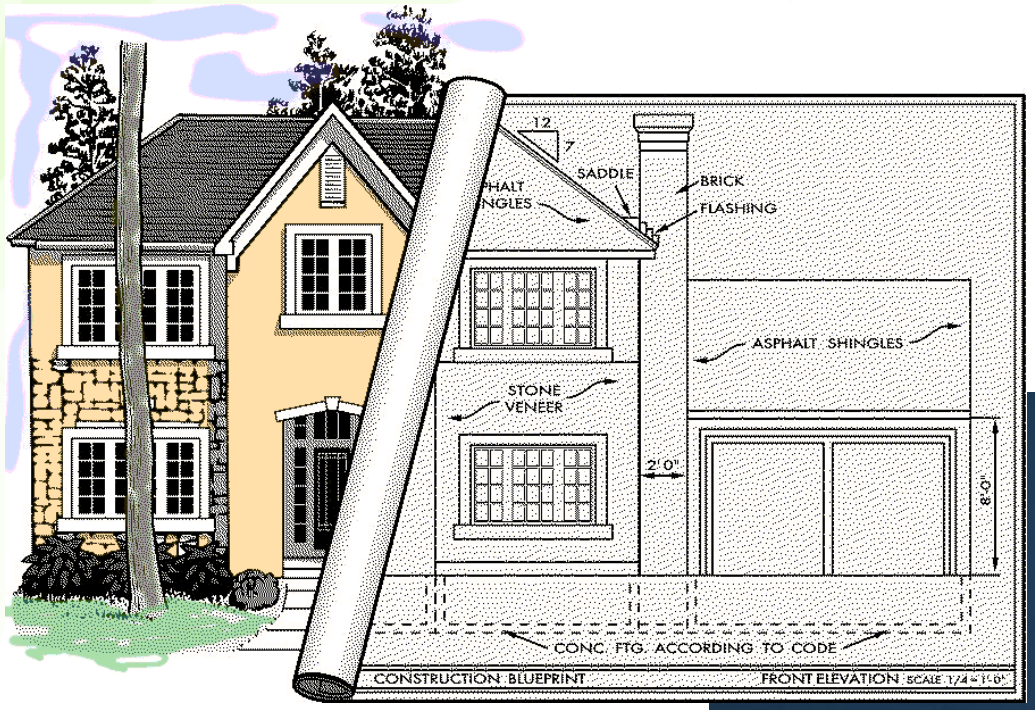
Para construir uma casa...

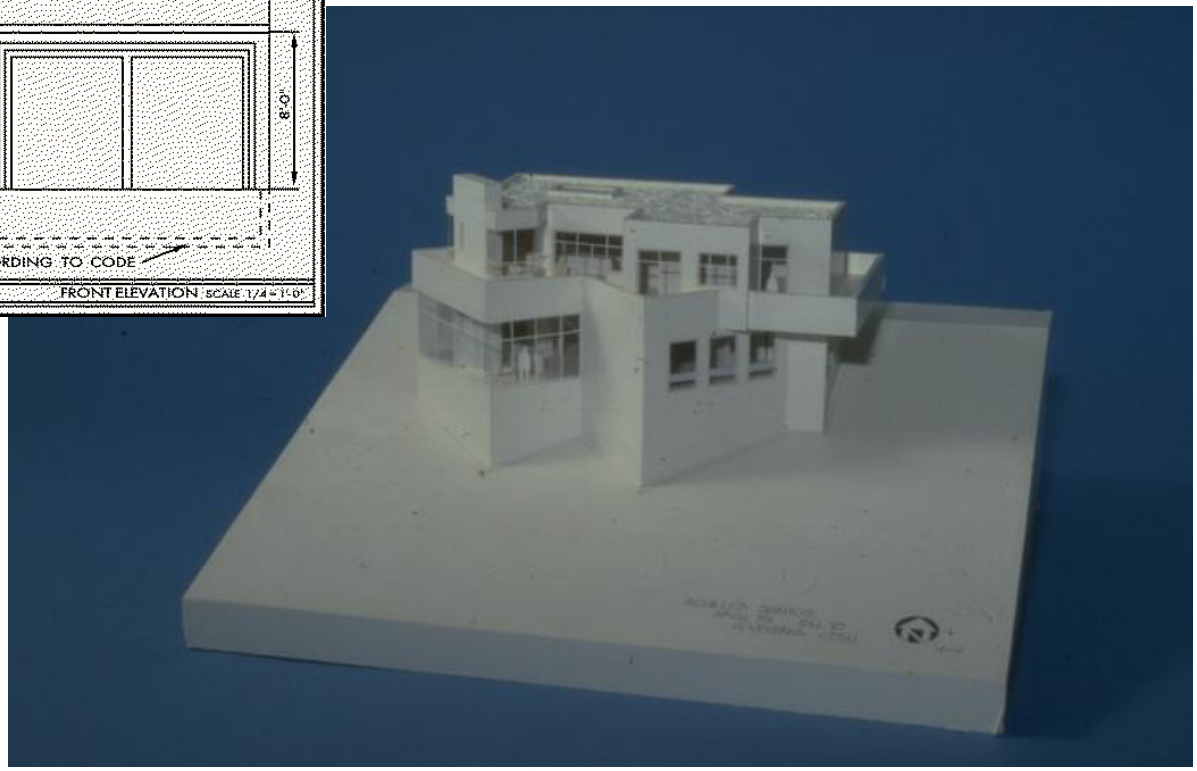
- É necessária uma equipe
- Requer um projeto
- Requer um processo bem definido
- Requer ferramentas especializadas

Fonte: Grady Booch. Software Architecture and the UML.

Projetando uma casa

LABES-UFPA
Laboratório de Engenharia de Software da UFPA
www.labes.ufpa.br





Fonte: Grady Booch. Software Architecture and the UML.

E para construer um prédio?

Fonte: Grady Booch. Software Architecture and the UML.

Por que precisamos da Engenharia de Software?

LABES-UFPA
Laboratório de Engenharia de Software da UFPA
www.labes.ufpa.br

Mas eu já sei programar!! ☺️

Mas o quê programar?
Como programar?
O que eu programei está certo?

Engenharia vs. Programação

LABES-UFPA
Laboratório de Engenharia de Software da UFPA
www.labes.ufpa.br

- Programação
  - Projeto pequeno
  - Você
  - Construir o que você mesmo quer
  - Um produto
  - Poucas modificações, feitas em sequência
  - Tempo de vida curto
  - Barato
  - Pouco impacto
- Engenharia
  - Projeto Grande
  - Times
  - Construir o que os clientes querem
  - Família de produtos
  - Várias modificações, feitas em paralelo
  - Tempo de vida longo
  - Caro
  - Grande Impacto

Por que Engenharia de Software é a disciplina mais importante do curso, até o momento?

LABES-UFPA
Laboratório de Engenharia de Software da UFPA
www.labes.ufpa.br

Antes desta disciplina

Depois desta disciplina

Introdução

LABES-UFPA
Laboratório de Engenharia de Software da UFPA
www.labes.ufpa.br

- Características do Software
  - O software é desenvolvido, ou passa por um processo de engenharia, não é manufaturado no sentido clássico;
  - Software não se desgasta, mas se "deteriora";
  - Apesar da indústria de software cada vez mais se utilizar de componentes "pré-fabricados", a maior parte do software continua a ser construído sob encomenda;

O que é software?

LABES-UFPA
Laboratório de Engenharia de Software da UFPA
www.labes.ufpa.br

- Software consiste em:
  - (1) instruções (programas de computador) que, quando executadas, fornecem características, funções e desempenho desejados;
  - (2) estruturas de dados que possibilitam aos programas manipular informações adequadamente; e
  - (3) informação descritiva, tanto na forma impressa quanto na virtual, descrevendo a operação e o uso dos programas.

Roger, PRESSMAN,, MAXIM, Bruce. Engenharia de Software, 8th Edition. AMGH, 01/2016. VitalBook file.

LABES-UFPA
Laboratório de Engenharia de Software da UFPA
www.labes.ufpa.br

Construir Software é difícil

Hardware

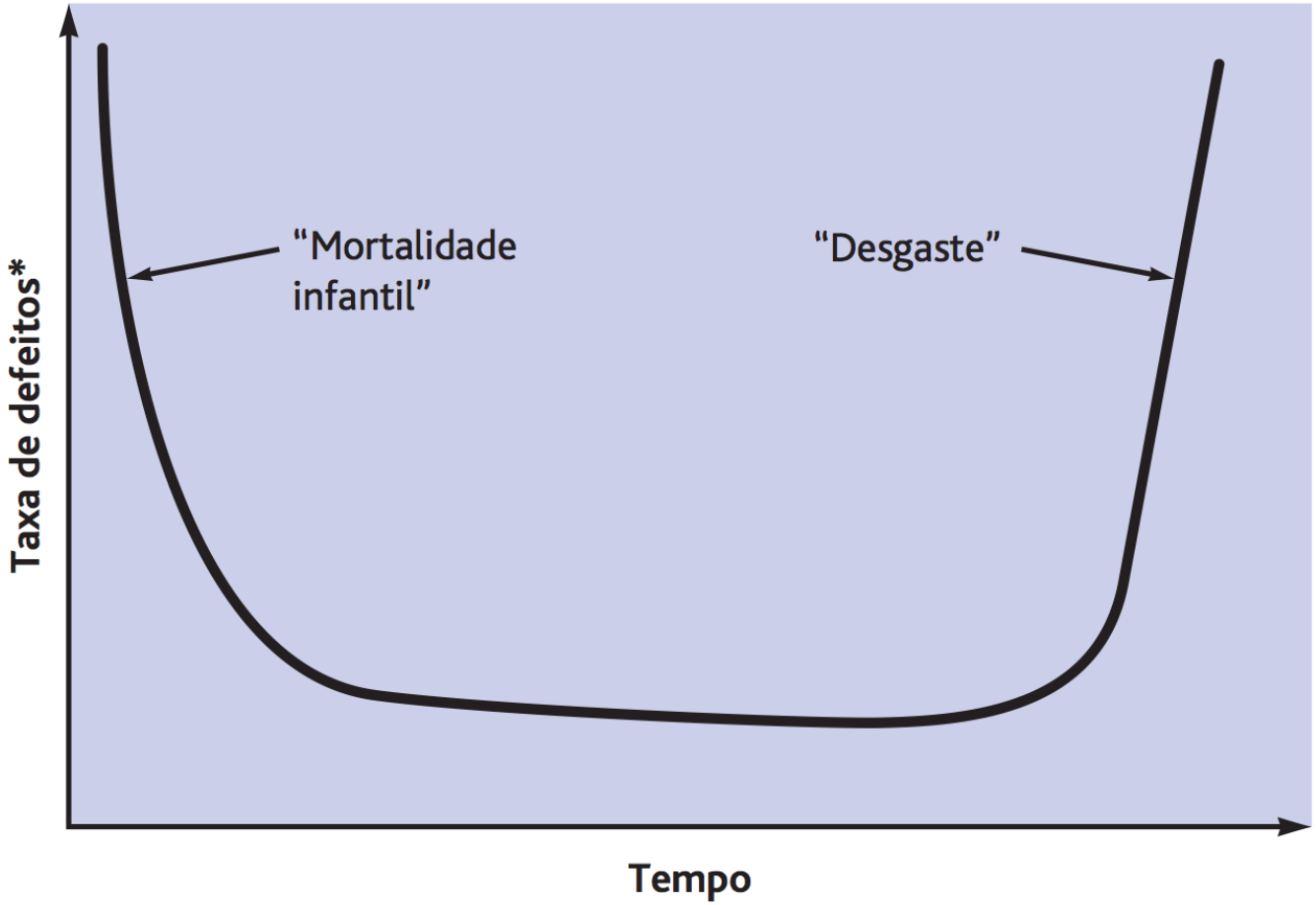


FIGURA 1.1 Curva de defeitos para hardware.

Pressman

Software

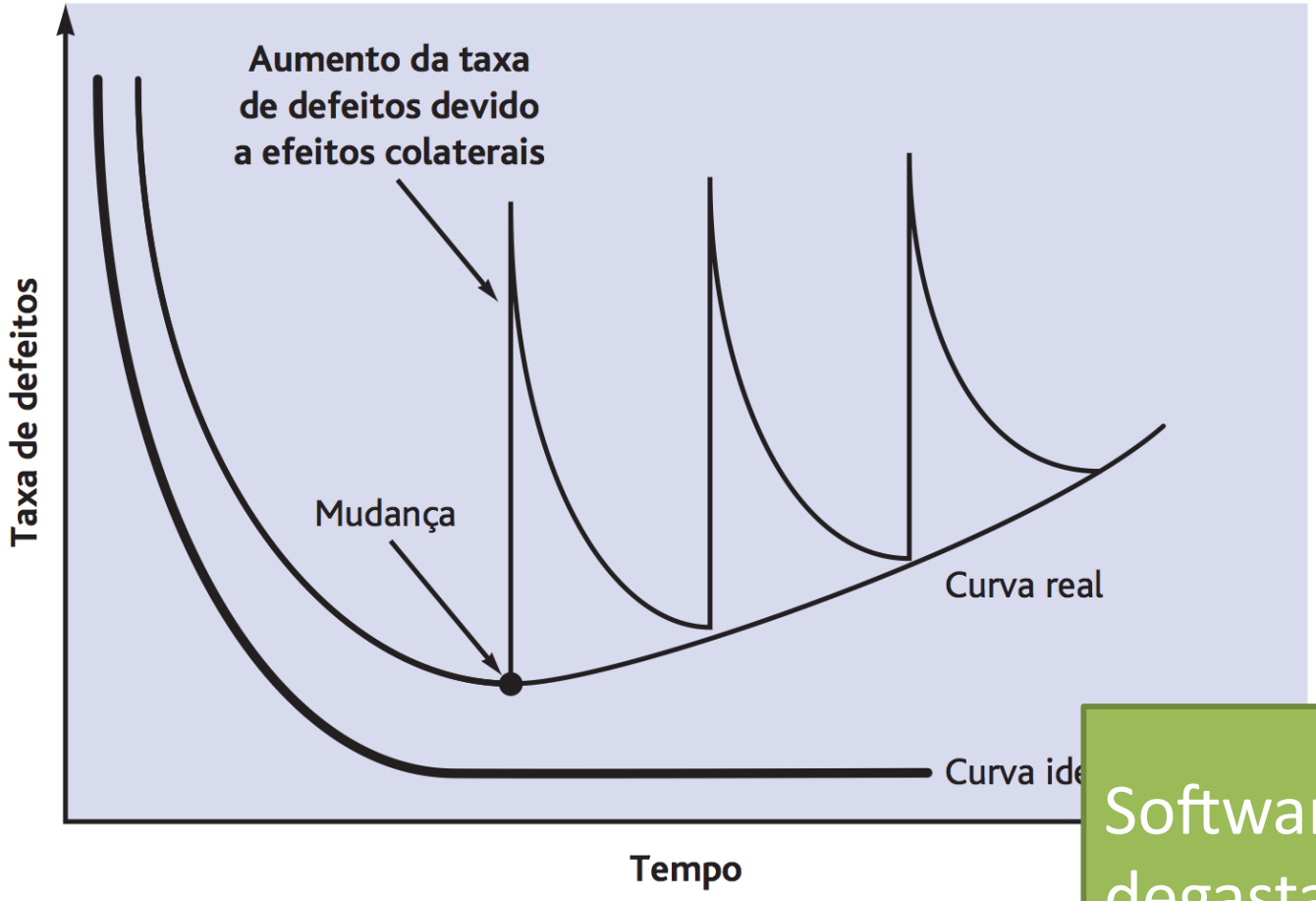
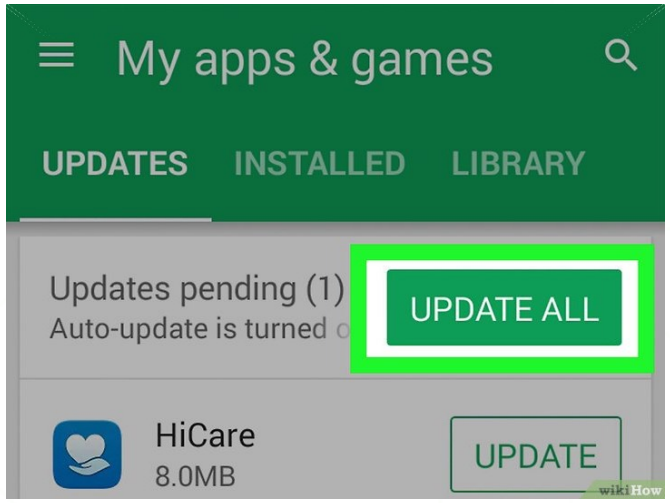


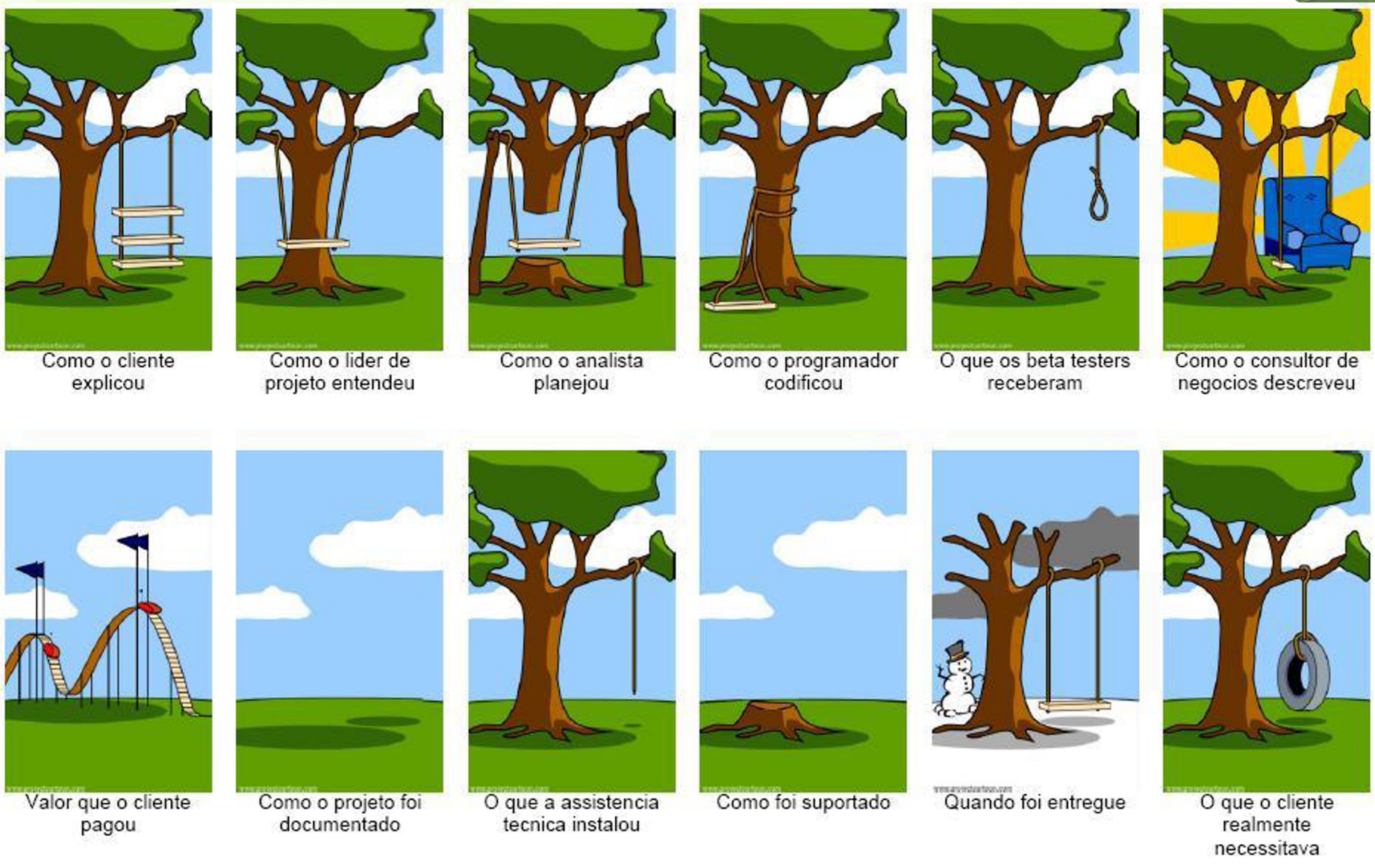
FIGURA 1.2 Curva de defeitos para software.

Software não se degasta, mas se "deteriora".

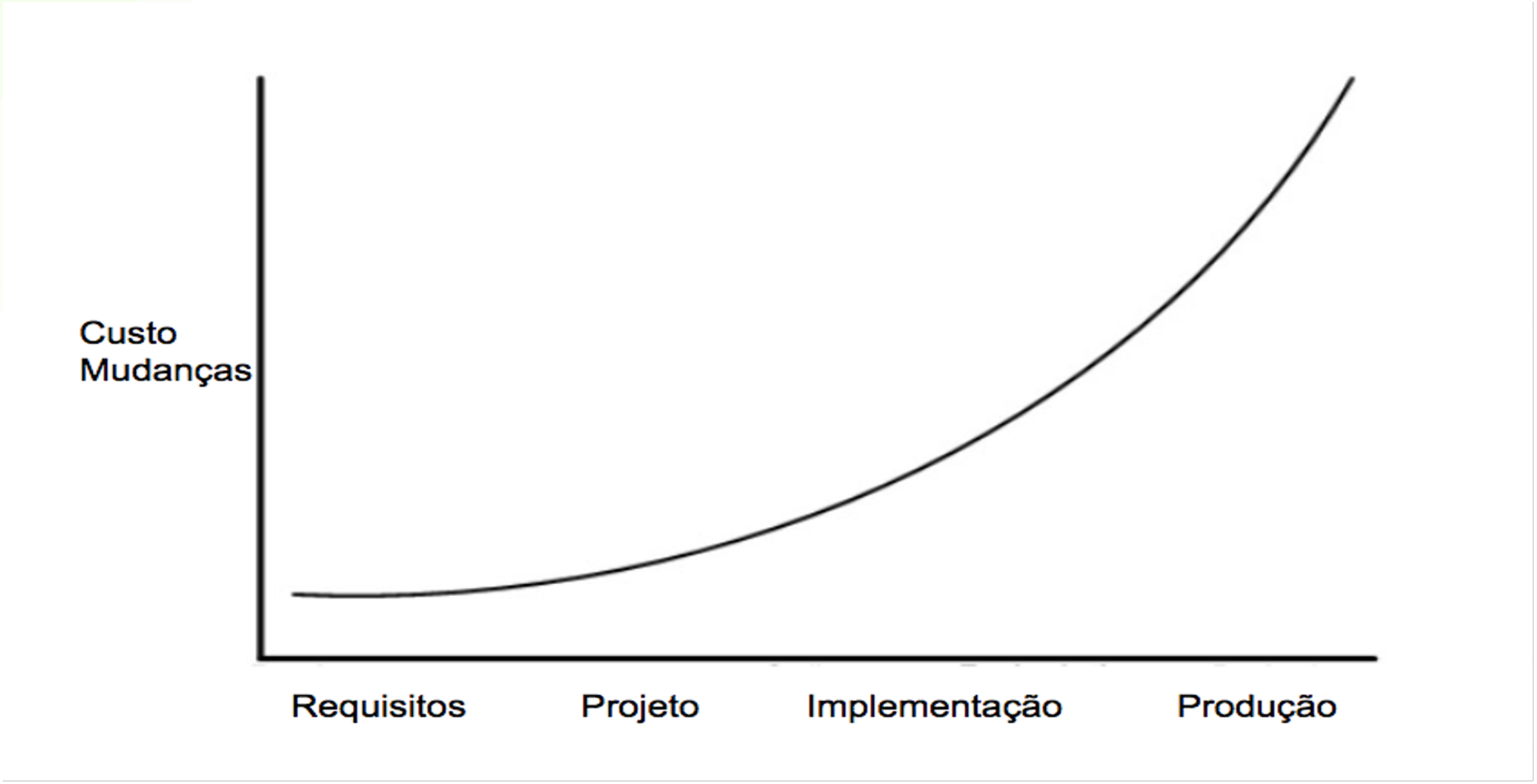
Hardware vs. Software

- Software não se desgasta… Mas se "deteriora"!
  - Quando um componente de hardware se desgasta, ele é substituído por uma peça de reposição.
  - Não existem peças de reposição de software! Cada defeito de software indica um erro no projeto ou no processo pelo qual o projeto foi traduzido em código de máquina executável.
  - Isto significa que as tarefas de que envolvem solicitações de mudanças no software, implicam em uma complexidade maior do que a de manutenção de hardware.




Como o cliente explicou
Como o lider de projeto entendeu
Como o analista planejou
Como o programador codificou
O que os beta testers receberam
Como o consultor de negocios descreveu
Valor que o cliente pagou
Como o projeto foi documentado
O que a assistencia tecnica instalou
Como foi suportado
Quando foi entregue
O que o cliente realmente necessitava

Custo de correção



Custos gerados por problemas em requisitos

- Segundo Boehm e Papaccio (apud Pfleeger, 2004), o custo relativo para o conserto de um problema de requisitos em cada fase de desenvolvimento de sistema é:
  - $1 na fase de análise de requisitos
  - $5 na fase de projeto do sistema
  - $10 na fase de codificação
  - $20 na fase de teste de unidade
  - $200 após a entrega do sistema

[Conte 2007]

Erros em requisitos

- Quanto mais tarde um erro é detectado, maior o custo de corrigí-lo.
- Muitos erros são realizados durante a elicitação e definição de requisitos
- Muitos erros nos requisitos podem ser detectados cedo no ciclo de desenvolvimento.

- Erros típicos incluem fatos incorretos, omissões, inconsistências e ambiguidade.
- Erros nos requisitos constituem uma das maiores preocupações da indústria de software.

[Breitman, 2007]

Erros em requisitos

- Fred Brook's adiciona:

"a parte mais difícil da construção de um sistema de software é decidir, precisamente, o que deve ser construído… Nenhuma outra parte do trabalho aleja mais o sistema resultante se feita errada. Nenhuma outra parte é mais difícil de corrigir depois."

[Breitman, 2007]

Mitos do Software

LABES-UFPA
Laboratório de Engenharia de Software da UFPA
www.labes.ufpa.br

- Mitos do Cliente
  - O estabelecimento geral de objetivos é suficiente para iniciar a escrita de programas – podemos fornecer os detalhes posteriormente.
  - Os requisitos de projeto mudam continuamente, mas as mudanças podem ser facilmente acomodadas porque o software é flexível.

Mitos do Software (2)

LABES-UFPA
Laboratório de Engenharia de Software da UFPA
www.labes.ufpa.br

- Mitos do profissional
  - Quando escrevemos um programa e o fazemos funcionar, nosso trabalho está completo.
  - Até que eu esteja com o programa “rodando” não tenho como avaliar a sua qualidade.
  - O único produto de trabalho que pode ser entregue para um projeto de software bem-sucedido é o programa executável.
  - A Engenharia de Software vais nos fazer criar documentação volumosa e desnecessária que certamente nos atrasará.

Mitos do Software (3)

LABES-UFPA
Laboratório de Engenharia de Software da UFPA
www.labes.ufpa.br

- Mitos da Gerência
  - Já temos um livro que está cheio de padrões e procedimentos para elaborar um software. Isso não fornece ao meu pessoal tudo o que eles precisam saber?
  - Se nos atrasarmos no cronograma, podemos adicionar mais programadores e ficar em dia? (Lei de Brooks)
  - Se eu decidir terceirizar um projeto de software, vou poder relaxar e deixar que aquela firma o elabore?

LABES-UFPA
Laboratório de Engenharia de Software da UFPA
www.labes.ufpa.br

Historicamente

A **Crise de Software**...

LABES-UFPA
Laboratório de Engenharia de Software da UFPA
www.labes.ufpa.br

- Histórico:
  - Conferência da OTAN em 1968, onde um grupo de especialistas se reuniu para discutir um "problema crucial do uso de computadores, o chamado software".
  - O termo Engenharia de Software foi cunhado

Crise do Software (2)

LABES-UFPA
Laboratório de Engenharia de Software da UFPA
www.labes.ufpa.br

> *"O problema é que certas classes de sistemas estão colocando demandas sobre nós que estão além das nossas capacidades e das teorias e métodos de projeto que conhecemos no presente tempo. Em algumas aplicações não existe uma crise, como rotinas de ordenação e folhas de pagamento, por exemplo. Porém, estamos tendo dificuldades com grandes aplicações. Não podemos esperar que a produção de tais sistemas seja fácil."*

- *Valente, M. T.. Engenharia de Software Moderna: Princípios e Práticas para Desenvolvimento de Software com Produtividade, 2020.*

Crise do Software (3)

LABES-UFPA
Laboratório de Engenharia de Software da UFPA
www.labes.ufpa.br

- Projetos estourando o prazo;
- Projetos estourando o orçamento;
- Software de baixa qualidade;
- Software muitas vezes não satisfaz os requisitos;
- Projetos ingerenciáveis e código difícil de manter;

Crise do Software (em 2015)

LABES-UFPA
Laboratório de Engenharia de Software da UFPA
www.labes.ufpa.br

MODERN RESOLUTION FOR ALL PROJECTS

	2011	2012	2013	2014	2015
SUCCESSFUL	29%	27%	31%	28%	29%
CHALLENGED	49%	56%	50%	55%	52%
FAILED	22%	17%	19%	17%	19%

The Modern Resolution (OnTime, OnBudget, with a satisfactory result) of all software projects from FY2011–2015 within the new CHAOS database. Please note that for the rest of this report CHAOS Resolution will refer to the Modern Resolution definition not the Traditional Resolution definition.

- CHAOS Report *2015. The Standish Group International, Inc. Consultado em 11 de Março de 2021. https://www.standishgroup.com/sample_research_files/CHAOSReport2015-Final.pdf*

Crise do Software (em 2020)

LABES-UFPA
Laboratório de Engenharia de Software da UFPA
www.labes.ufpa.br

- Successful: 31%
- Challenged: 50%
- Failed 19%