



Engenharia de Software - Fundamentos

Introdução a Engenharia de Software

Prof. Guilherme Lacerda

guilhermeslacerda@gmail.com

Roteiro

- Por que precisamos planejar?
- Problemas e desafios
- Complexidade em Desenvolvimento de Software
- Conceitos de Engenharia

"A questão não é **SE** a qualidade técnica é importante, mas **COMO** a qualidade técnica será alcançada"

Jim Highsmith

Agile Project Management: creating innovative products, Addison-Wesley Professional (2004)

Por que precisamos planejar?

Revisando Conceitos

Processo

Projeto

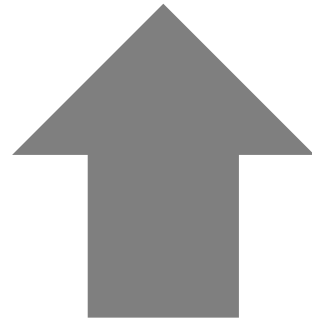
**Produto e
Serviço**

Operação



Engenharia de Software

Processo → Produto



Qualidade



**...e quando o software ficar
pronto...**



**É fundamental entender a natureza
do software**



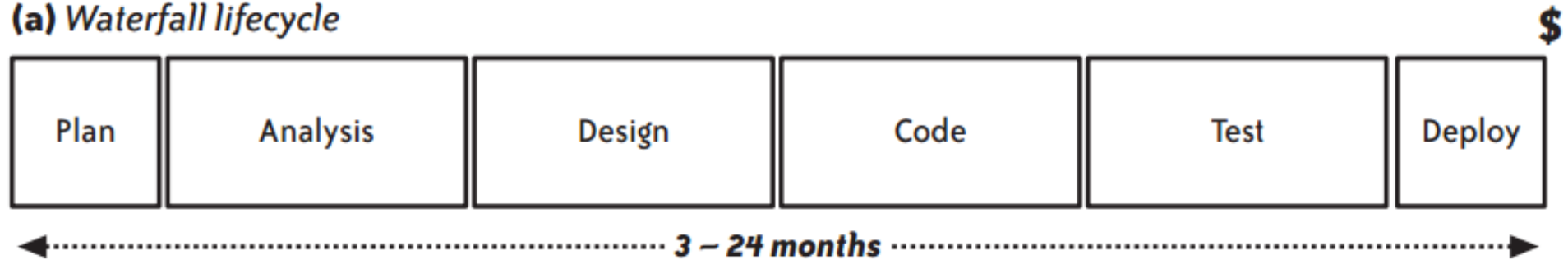
Leis de Lehman

Lei	Descrição
Mudança contínua	Um programa usado em um ambiente do mundo real deve necessariamente mudar, ou se torna progressivamente menos útil nesse ambiente.
Aumento da complexidade	Como um programa em evolução muda, sua estrutura tende a tornar-se mais complexa. Recursos extras devem ser dedicados a preservar e simplificar a estrutura.
Evolução de programa de grande porte	A evolução de programa é um processo de autorregulação. Atributos de sistema como tamanho, tempo entre <i>releases</i> e número de erros relatados são aproximadamente invariáveis para cada <i>release</i> do sistema.
Estabilidade organizacional	Ao longo da vida de um programa, sua taxa de desenvolvimento é aproximadamente constante e independente dos recursos destinados ao desenvolvimento do sistema.
Conservação da familiaridade	Durante a vigência de um sistema, a mudança incremental em cada <i>release</i> é aproximadamente constante.
Crescimento contínuo	A funcionalidade oferecida pelos sistemas tem de aumentar continuamente para manter a satisfação do usuário.
Declínio de qualidade	A qualidade dos sistemas cairá, a menos que eles sejam modificados para refletir mudanças em seu ambiente operacional.
Sistema de <i>feedback</i>	Os processos de evolução incorporam sistemas de <i>feedback</i> multiagentes, <i>multiloop</i> , e você deve tratá-los como sistemas de <i>feedback</i> para alcançar significativa melhoria do produto.

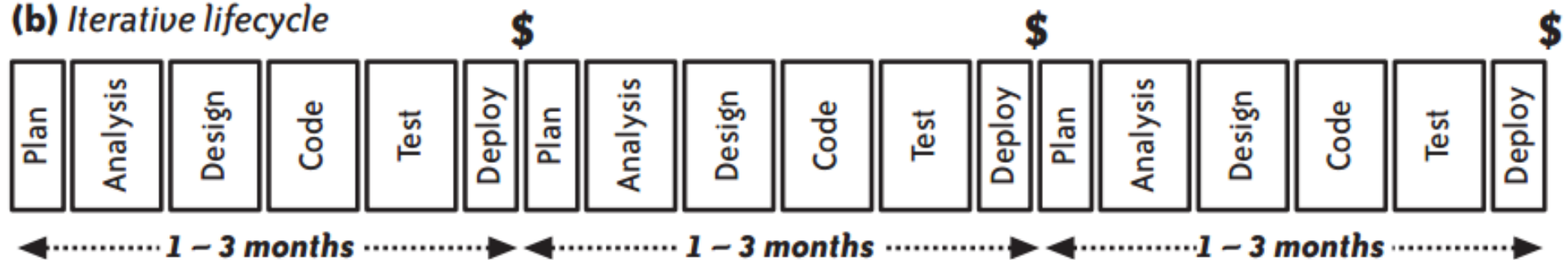


O desenvolvimento de software...

(a) *Waterfall lifecycle*



(b) *Iterative lifecycle*



\$ = Potential release



Falta de Comunicação



Como o cliente explicou



Como o consultor de negócio descreveu



Como o líder de projeto entendeu



Como o analista modelou



Como o desenvolvedor programou



O que foi recebido pelos "beta testers"



Falta de Comunicação



Resultado dos testes de "stress"



Como o software foi instalado pela equipe de operação



Como foi feita a Gerência de Riscos



Como o projeto foi documentado



Como o software foi suportado



Quando o software foi entregue



Falta de Comunicação



Resultado da utilização do software



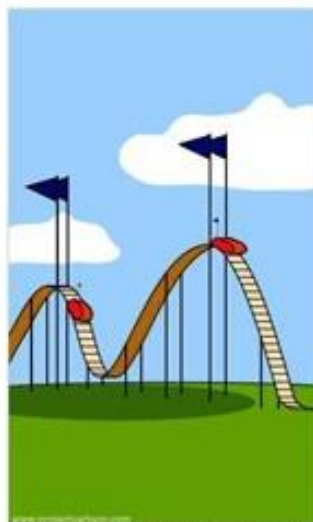
Como os "patches" foram aplicados



Como foi a campanha de marketing



O que o cliente realmente queria



Como o cliente foi cobrado



Versão Open-source do Software

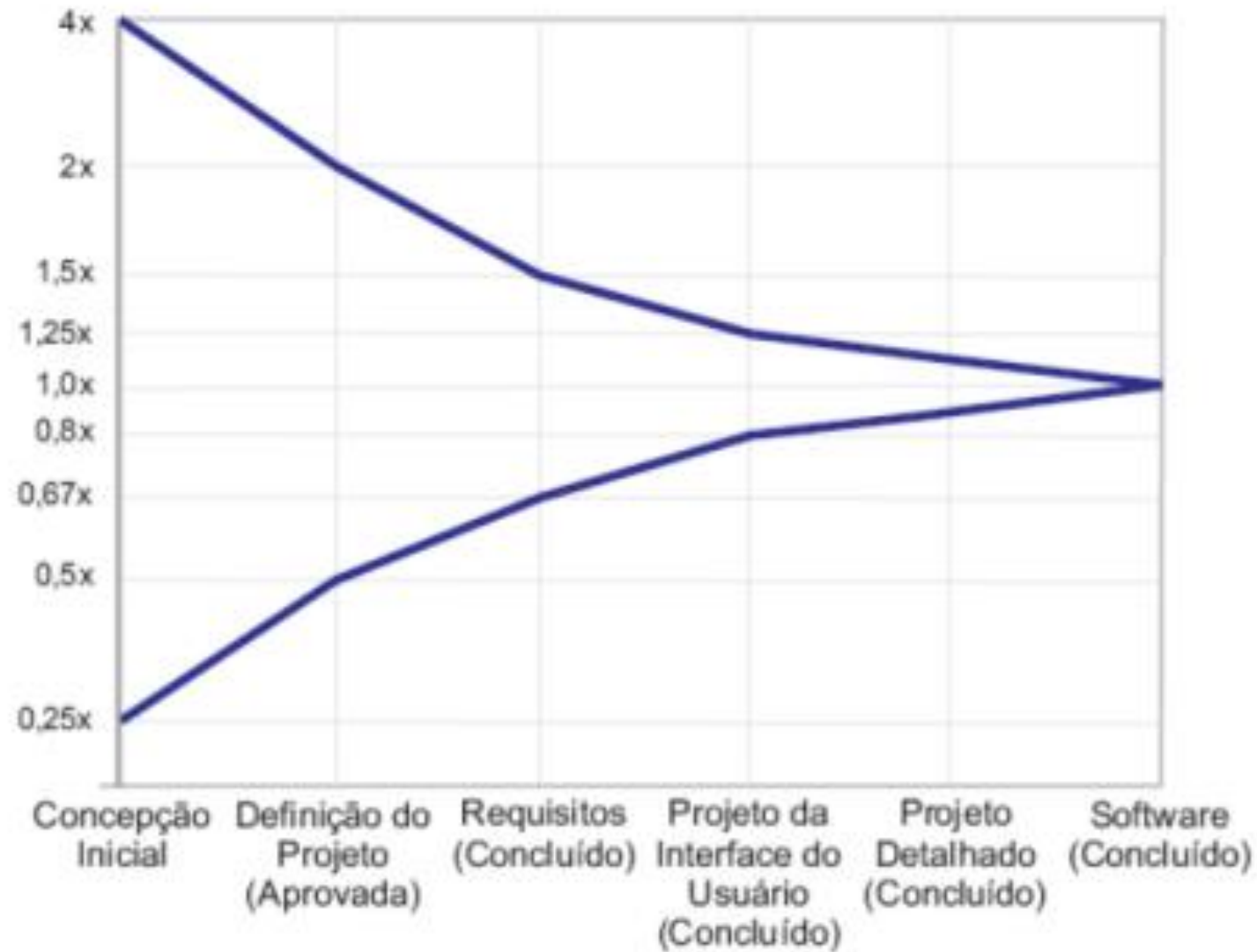


“Uma **gestão ruim** pode aumentar os custos de desenvolvimento de software mais do que qualquer outro fator”

Barry Boehm

Software Engineering Economics, Prentice Hall (1981)

Cone da Incerteza



Visões

Table 5.1. Software Maintenance Categories		
	Correction	Enhancement
Proactive	Preventive	Perfective
Reactive	Corrective	Adaptive

SWEBoK v3.0, IEEE Computer Society (2014)

Desenvolvimento + Manutenção
=
Evolução!



Problemas e Desafios

Falhas em Projetos: Therac 25



- Therac-25
 - Aparelho de Radioterapia
 - de 1985 a 1987 se envolveu em 6 acidentes, causando mortes por radiação
 - Software adaptado da versão anterior
 - Falha por falta de testes integrados

<https://pt.wikipedia.org/wiki/Therac-25>



Falhas em Projetos: Denver International Airport



- Denver International Airport
 - Custo US\$: 4,9 Bi – 100 mil passageiros/dia, 1.200 voos
 - Erro no sistema automático de bagagens interligando 3 terminais
 - Atrasos de mais de 16 meses
 - US\$ 560 mi acima do orçamento



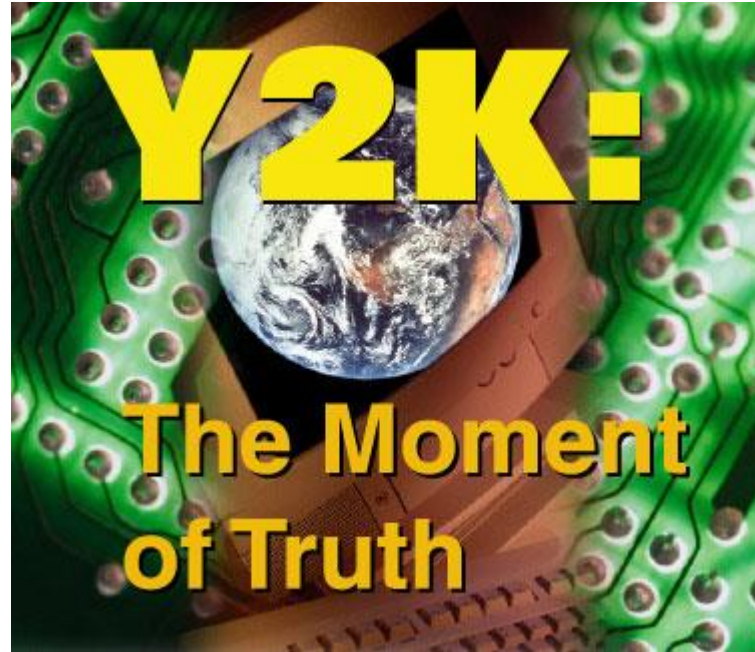
Falhas em Projetos: Ariane 5

- Ariane 5
 - Projeto Espacial da Agência Europeia
 - 10 anos de projeto – US\$ 8 Bi
 - Garantia a supremacia europeia no espaço
 - Explosão 40 seg. após decolagem
 - Carga de transporte avaliada em US\$ 500 mi
 - Ocorreu um overflow, desligando os servidores
 - Conversão de ponto flutuante para inteiro (casting)

<https://www.bugsnap.com/blog/bug-day-ariane-5-disaster>



Falhas em Projetos: Bug do Milênio



- Bug do Milênio (Y2K)

https://pt.wikipedia.org/wiki/Problema_do_ano_2000



BUG 2000

Bagé se prepara para o bug do milênio



PANE NOS COMPUTADORES: no Brasil, serão investidos R\$ 1,5 bilhão para prevenir os prejuízos do bug. Imagine a seguinte situação: é 31 de dezembro de 1999, sexta-feira, e você acaba de encerrar o expediente. Desliga o computador e vai para casa, já pensando em festejar a tão esperada virada de ano, o réveillon do ano 2000. Segunda-feira, recuperado dos abusos éticos e gastronômicos, você retorna ao trabalho e tenta

ligar o computador. Até consegue ligá-lo, mas quem garante que a máquina vai funcionar direito. É o bug do milênio.

Para economizar espa-

ço na memória dos computadores, os pioneiros em informática decidiram guardar as datas com apenas dois dígitos reservados para o ano. Por exemplo, sete de agosto de 1999, no computador, fica 07/08/99.

Por isso, ao ler 00 (de 2000), os computadores podem entender 1900. Essa simples confusão no calendário das "inteligências artificiais" pode causar um estrago enorme e prejuízos irreparáveis, ocasionando um descontrole total nos sistemas de telecomunicações, tráfego aéreo, transmissão de energia, operações financeiras e até mesmo acionar bombas nucleares.

Em todo o mundo, o custo dos reparos para evitar os transtornos do bug estão sendo calculados entre 600 bilhões a um trilhão de dólares. Os 600 bilhões são

só para solucionar os problemas nos programas. O gasto podem chegar até um trilhão contando com as eventuais indenizações judiciais. Só os Estados Unidos estão gastando cerca de 80 bilhões de dólares para prevenir a pane do réveillon.

No Brasil, o governo federal destinou R\$ 1,5 bilhão para reprogramar os computadores tupiniquins. No entanto, o prazo final para adaptação foi prorrogado até o final de setembro, uma vez que 15% dos sistemas ainda não foram testados e certificados.

Em Bagé, mesmo com as dificuldades de acesso à informações, o MINUANO efetuou um levantamento de como as instituições públicas e privadas estão se preparando para o bug do milênio.

Santo de casa milagreiro

Quem disse que santo de casa não faz milagres. Pois um estudante bageense desenvolveu um programa que previne os infortúnios do bug do milênio. Guilherme Silva de Lacerda, 20 anos, técnico em processamento de dados, aluno do sexto semestre de informática e funcionário do setor de Planejamento da Urcamp, além de auxiliar do núcleo de pesquisa e extensão do CCEI, desde setembro de 1998, se dedica a estudar o fenômeno do bug.

No início, Guilherme era movido apenas pela curiosidade. Lendo revistas especializadas, livros e artigos sobre a previsível pane dos computadores na virada do ano, Guilherme verificou que o bug não era tão assustador assim. "Claro que o bug é um problema, mas não é tudo o que falam", afirma. Em 1996, Guilherme e o colega Fabrizio Mello já haviam desenvolvido um programa para a Urcamp - de avaliação institucional - que é imune aos efeitos do bug.

A partir do ano passado, em conversas com professores e colegas, Guilherme começou a procurar so-

luções. "A melhor saída seria reescrever todos os programas, mas é muito trabalhoso, então eu optei por armazenar as datas utilizando outros calendários", resume Guilherme.

Segundo o estudante, o ponto-chave do bug está na quantidade de dígitos que se pode usar na hora de escrever a data. O sistema desenvolvido por Guilherme converte a data com dois algarismos no ano para outro calendário, fazendo com que o computador entenda o 00 como referente ao ano 2000 e não 1900. "O sistema consiste em fórmulas matemáticas que convertem as datas. Por exemplo, eu posso fazer com que a data dd/mm/99 seja interpretado pelo computador com quatro dígitos, ou até mesmo transformar em outro sistema de contagem, com o dia, o ano e o século, assim: dia yy, ano xx, do século 21", explica.

Com base nos seus estudos, Guilherme desenvolveu um programa demonstrativo, o Y2K, que está à disposição na internet (<http://attila.urcamp.tche.br/~guilherm>). No entanto, ele

afirma que não vende o programa, apenas ensina como proceder. "Estou sobrecarregado de serviço, por isso eu indico todos os passos a serem seguidos e a própria pessoa efetua a conversão", justifica.

Mas Guilherme faz uma ressalva: "Cada caso é um caso, não dá para generalizar". Segundo ele, cada empresa deve atacar seus pontos-chaves, "aquilo que não pode parar". O estudante indica que seja feito um planejamento antecipado e, depois, efetuadas várias simulações.

Para os usuários domésticos, Guilherme sugere que sejam feitos alguns testes. Conforme a sugestão do estudante, o usuário deve alterar a data do sistema, de preferência numa sexta-feira, "porque é necessário que o computador fique ocioso por, no mínimo, uns dois dias". Para alterar a data, o usuário deve acessar, no Windows, de um duplo clique na hora (exibindo no canto inferior esquerdo da área de trabalho). No MS-DOS, deve ser usado o comando date.

No local da data, o usu-



GUILHERME LACERDA: o estudante e funcionário da Urcamp desenvolveu um programa que evita os efeitos do bug

ário colocará 31/12/99. Dois dias depois, o computador deve ser ligado novamente. Se continuar funcionando, não há problemas à vista. Caso contrário, é preciso convocar o auxílio de algum

profissional especializado. "Em computadores 486 DX 2 e anteriores, há ocorrências de problemas internos na BIOS, então este teste não vai funcionar", previne Guilherme.

ONDE CONSEGUIR Auxílio

Ha varios sites na Internet onde você pode conseguir informações e até mesmo programas para evitar maiores danos nos seus arquivos. Anote e corra atrás que ainda há tempo.

- * www.s2000.gov.br
- * www.microsoft.com/y2k
- * www.microsoft.com/year2000
- * www.natl.com
- * www.symantec.com
- * www.compaq.com/year2000/year2000-qa.html
- * www.hp.com/year2000/allproducts.html
- * www.ibm.com/bm/year2000/pcsa/
- * www.itatuc.com.br/projeto2000



Falhas em Projetos: F35



- Início do projeto: 2001
- DoD
- Concorrência entre Boeing e **Lockheed Martin**
- 24 milhões de linhas de código
- Cerca de 1 trilhão de dólares

<http://m.spectrum.ieee.org/riskfactor/aerospace/military/f35-program-continues-to-struggle-with-software>



Falhas em Projetos: Airbus-A350



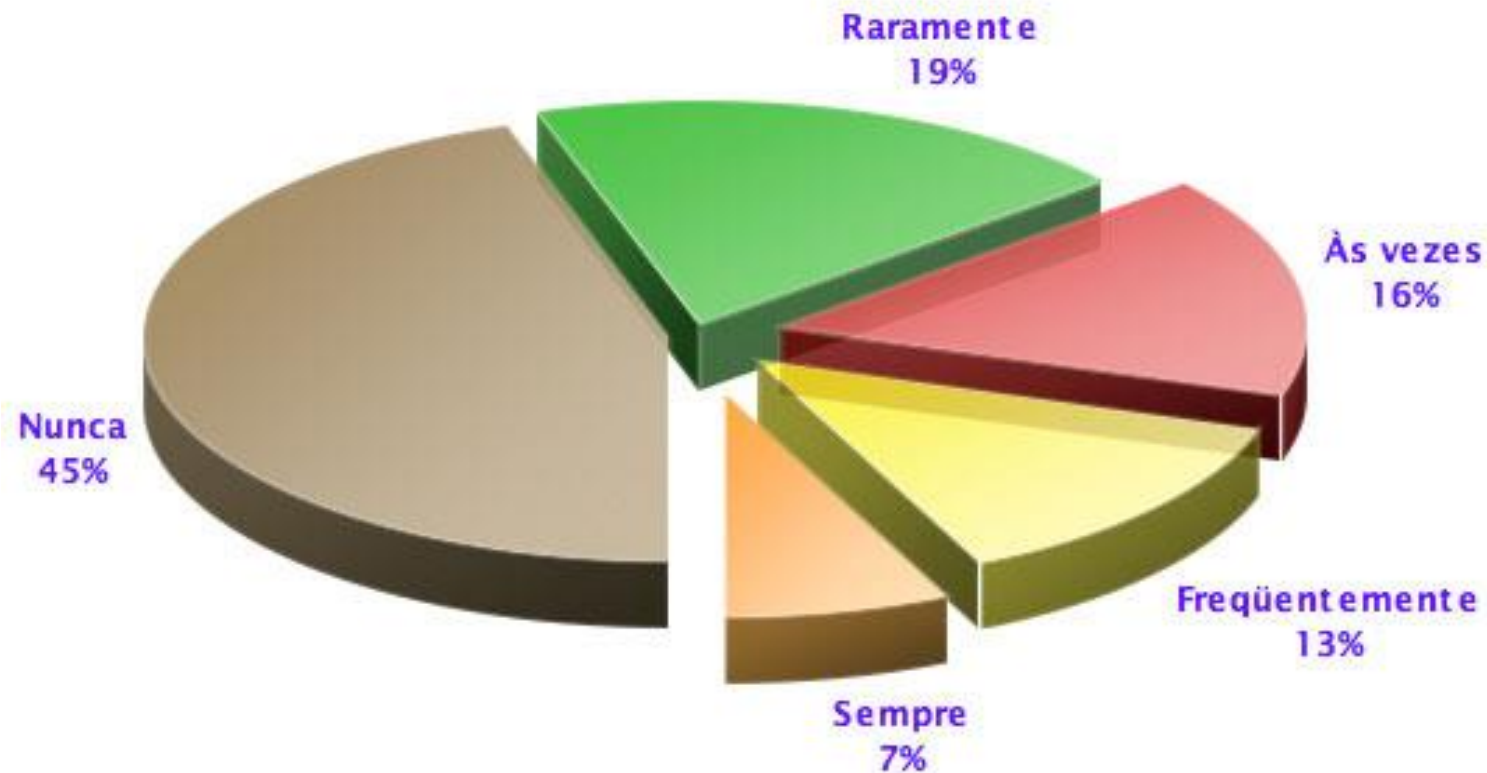
- “Reset a cada 149 horas”

<https://gizmodo-uol-com-br.cdn.ampproject.org/c/s/gizmodo.uol.com.br/reiniciar-aviao-airbus-solucao-bug/amp/>



Por que mais de 80% de projetos de software fracassam?

Frequência de Utilização de Funcionalidades
Sistemas Comerciais Típicos (EUA)



Fonte: Standish Group

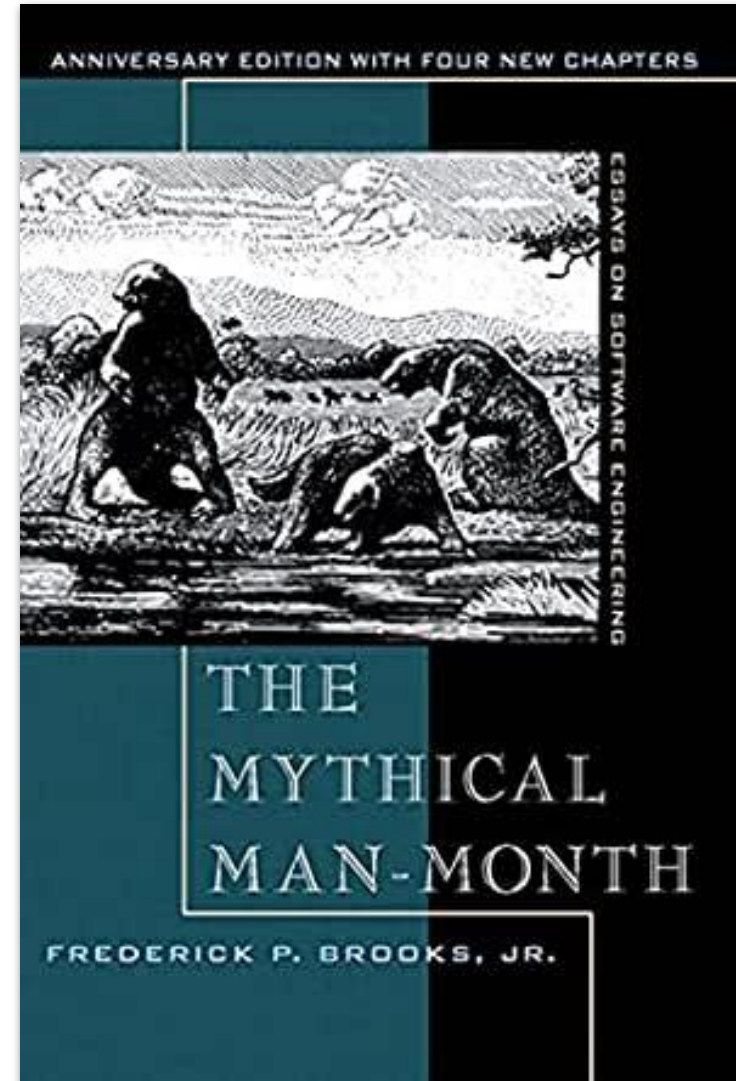
<https://www.opendoorerp.com/the-standish-group-report-83-9-of-it-projects-partially-or-completely-fail/>



Complexidade

Complexidade

Essencial
X
Acidental



Crise do Software

- Crise ou aflição crônica?
 - **"Crise:** Ponto decisivo no curso de algo"
 - **"Aflição Crônica:** algo que causa sofrimento, que dura longo tempo ou retorna freqüentemente"
- Ilusão: Desenvolver software é "soft"
 - Economicamente "soft"
 - Operacionalmente "soft"
 - Intellectualmente "soft"
- Realidade: Desenvolver software é "hard"
 - Atender a requisitos
 - Satisfazer o usuário
 - Respeitar cronograma e orçamento
 - Produzir com qualidade
 - Dificuldade em mantê-lo



Desenvolvimento de Software



- Desenvolvimento = 30% do esforço
 - Iniciado com a identificação das necessidades do produto
 - Finalizado com o produto implantado e concluído (entregue para operação/produção)
- Manutenção = 70% do esforço
 - Todas as atividades após a entrega
 - Aumento da capacidade do produto (60%)
 - Adaptação do produto a novos ambientes (20%)
 - Correção de erros (20%)



Desenvolvimento de Software

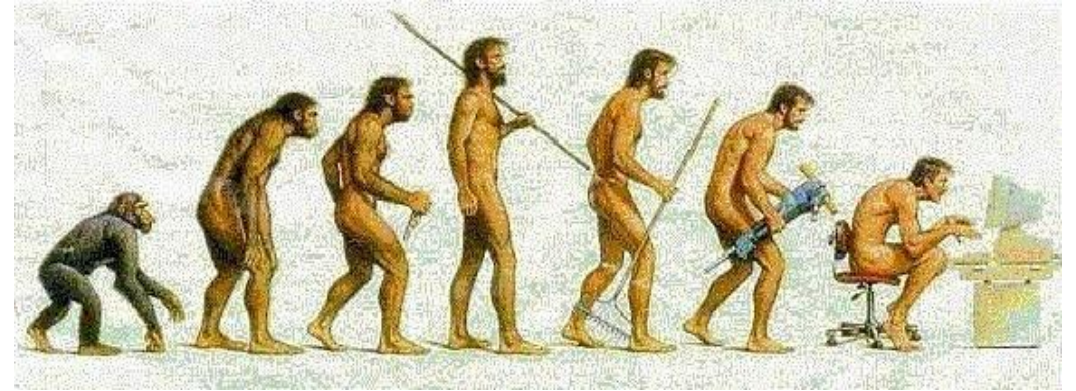
- Manutenção é cara e inevitável
 - Sistemas legados
 - Necessidades de ajustes
 - Mudança nos requisitos
- Como diminuir estes problemas?
 - Melhoria na qualidade do produto produzido
 - Melhoria do processo de produção

**Mudança cultural no
desenvolvimento de Software**



Desenvolvimento de Software

- Computadores mais baratos
- Software = Atividade intelectual
 - Mão de obra qualificada (\$\$)
 - Baseada em experiência
 - Baixos níveis de reutilização e automação
- Usuários mais exigentes
- Softwares mais complexos
 - Extinção do “Processamento de Dados”
 - Sistemas distribuídos, Web, SOA, Mobile, Embarcados
 - Processos tradicionais já não servem



Conceitos

Engenharia de Software

- Engenharia = Uso de princípios científicos para uma atividade de projeto e construção
- Engenharia de Software (várias definições)
 - “Aplicação prática do conhecimento científico para projeto, construção de programas computacionais e a documentação necessária à sua operação e manutenção”. (Barry Boehm, 1976)
 - “O estabelecimento e uso de sólidos princípios de engenharia para que se possa obter economicamente um software que seja confiável e que funcione eficientemente em máquinas reais”. (Fritz Bauer, 1969)



Engenharia de Software

- O que é engenharia de software?
 - Engenharia dedicada a todos os aspectos de produção de software
 - Da **especificação** até a **manutenção** do sistema que já está em uso
- ... da engenharia
 - Teorias e métodos para resolver problemas tendo em mente restrições organizacionais e financeiras
- para todos os aspectos de produção de software
 - não somente processos técnicos de desenvolvimento, mas:
 - gerenciamento de projetos
 - desenvolvimento de ferramentas e métodos para ajudar na produção de software
 - Relacionamento com usuários/clientes

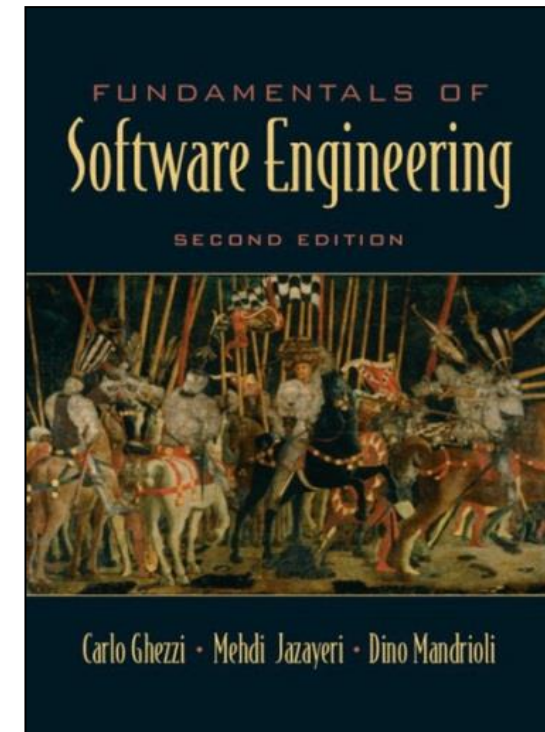
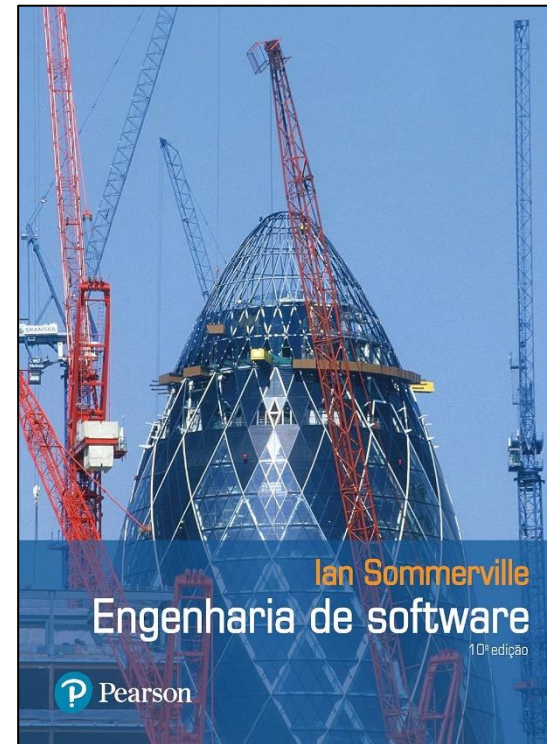
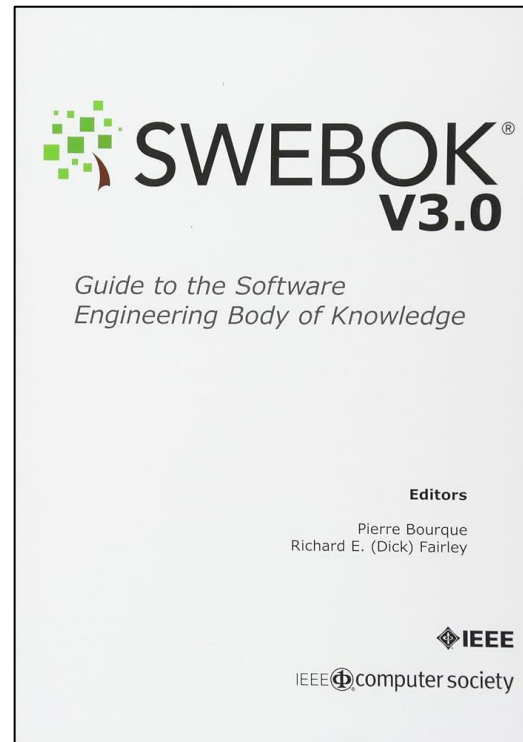
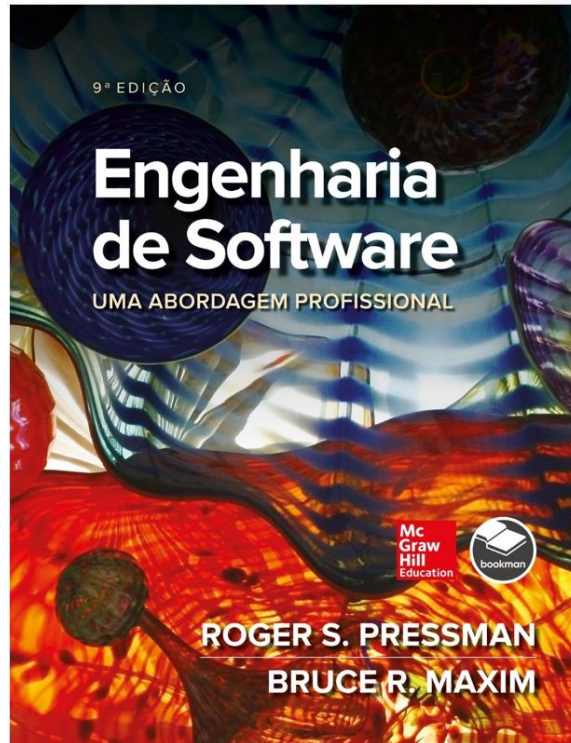


Engenharia de Software

- Qual a diferença entre engenharia de software e ciência da computação?
 - CC está relacionada a teoria e fundamentos
 - ES está relacionada com a prática de desenvolvimento e entrega de software útil
- Qual a diferença entre engenharia de software e engenharia de sistemas?
 - **Engenharia de sistemas** está relacionada a todos os aspectos de desenvolvimento de sistemas baseados em computadores, incluindo
 - hardware,
 - software e
 - engenharia de processo.
 - **Engenharia de software** é parte deste processo



Engenharia de Software



Engenharia de Software

Engenharia
de Software =

Desenvolver

Usar

Reusar

Integrar

n

X

n

Linguagens

Ferramentas

Técnicas

Métodos

Metodologias

Processos

Modelos

Artefatos

Conceitos

Princípios

Equipes

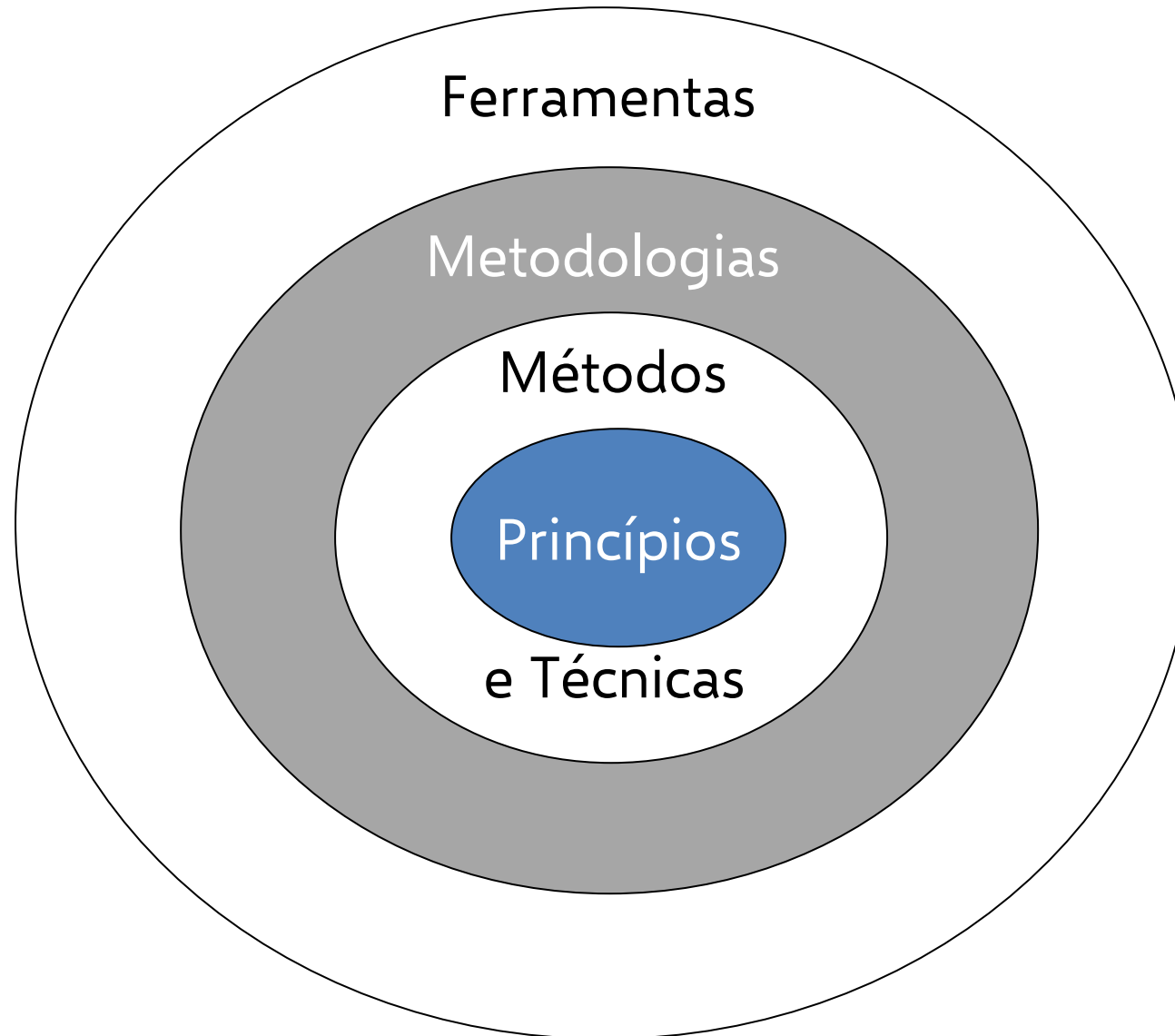


Núcleo da Engenharia de Software

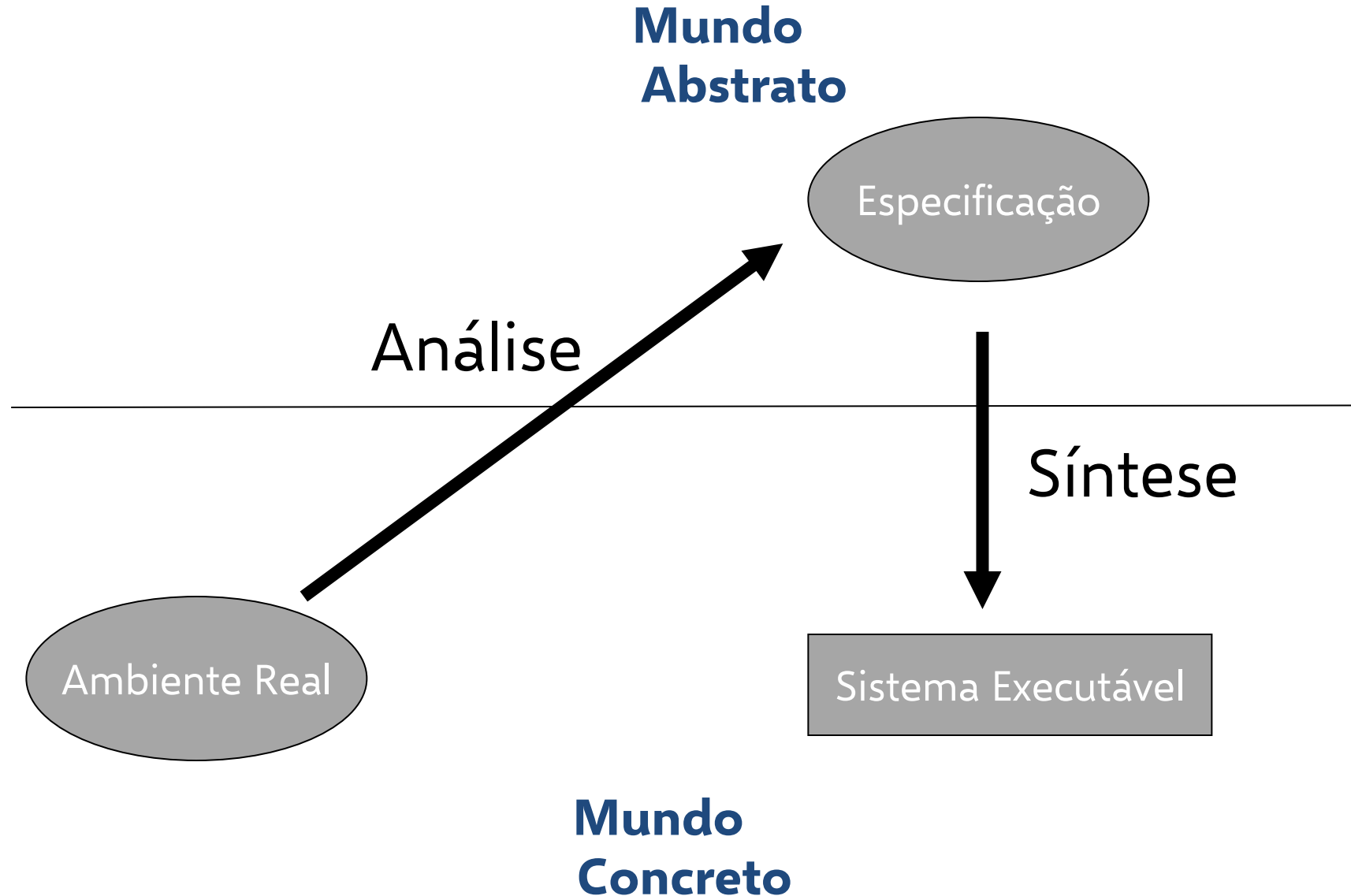
- Princípios
 - Declarações gerais e abstratas que descrevem propriedades
- Métodos
 - Linhas gerais que regulam previamente algumas ações
- Técnicas
 - Maneiras ou habilidades especiais de executar algo
- Metodologias
 - Conjunto de Métodos e/ou Técnicas para a solução de problemas
- Ferramentas
 - Mecanismos de apoio à aplicação de métodos, técnicas e metodologias



Núcleo da Engenharia de Software



Atividade de Desenvolvimento de Software



Atividade de Desenvolvimento de Software

