



INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E
TECNOLOGIA DO RIO GRANDE DO SUL
CAMPUS Canoas

ENGENHARIA DE SOFTWARE I

Como Construir Algo?

- Método Empírico;
- Método Científico;

Método Empírico

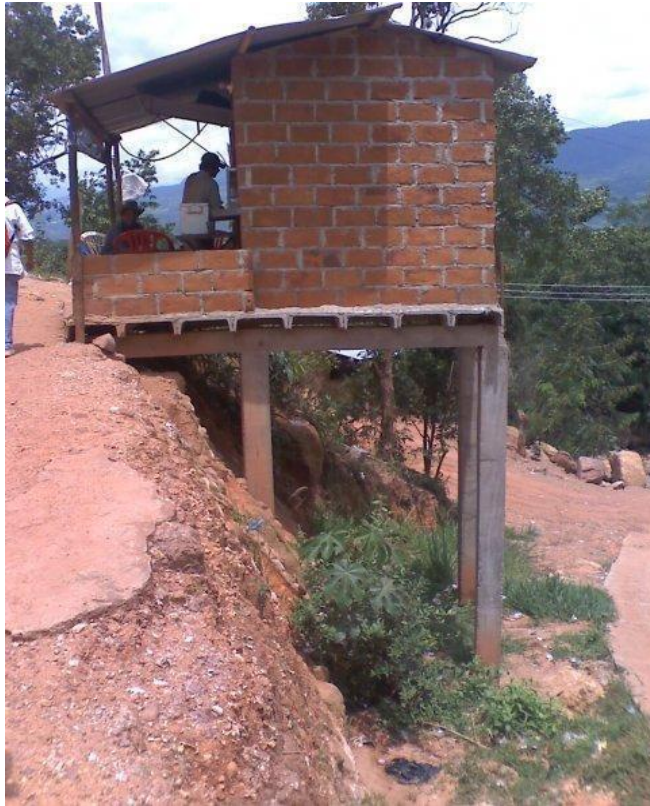


O produto gerado tem o problema **de não estar padronizado e dificilmente poderá ser medido e ser aferido o grau de qualidade;**

Você faz algo que utiliza como base **seu** conhecimento e corrige erros e faz aprimoramentos através de tentativa e erros;

Como exemplo temos casas construídas sem planejamento, programas de software que dão muitas manutenções.

Observe a qualidade da construção abaixo, como também a escada que “termina” em uma parede.



Método Empírico

No método empírico, problemas aparecem quando precisamos dar uma manutenção no produto. Ou mesmo adicionar novas funcionalidades



Observe a qualidade das construções aqui apresentadas. Como acessar as portas para entra nos ambientes da qual fazem parte.



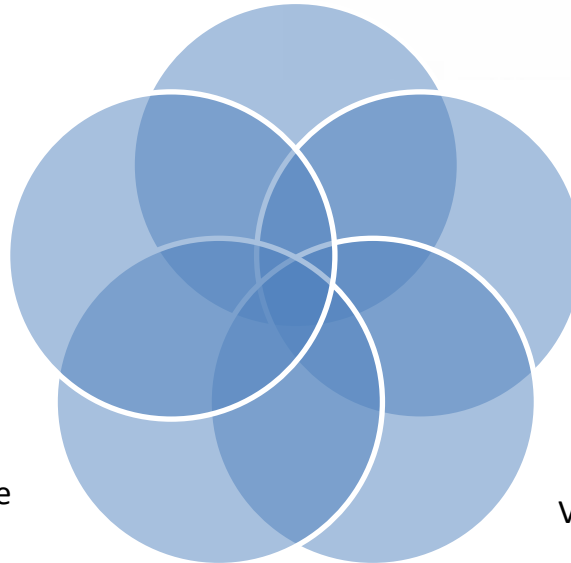
Método Científico



Faz um planejamento e elabora um projeto do que será construído.

Constrói o produto e verifica através de experimentos/estudos de caso se ele está de acordo com os critérios de qualidade.

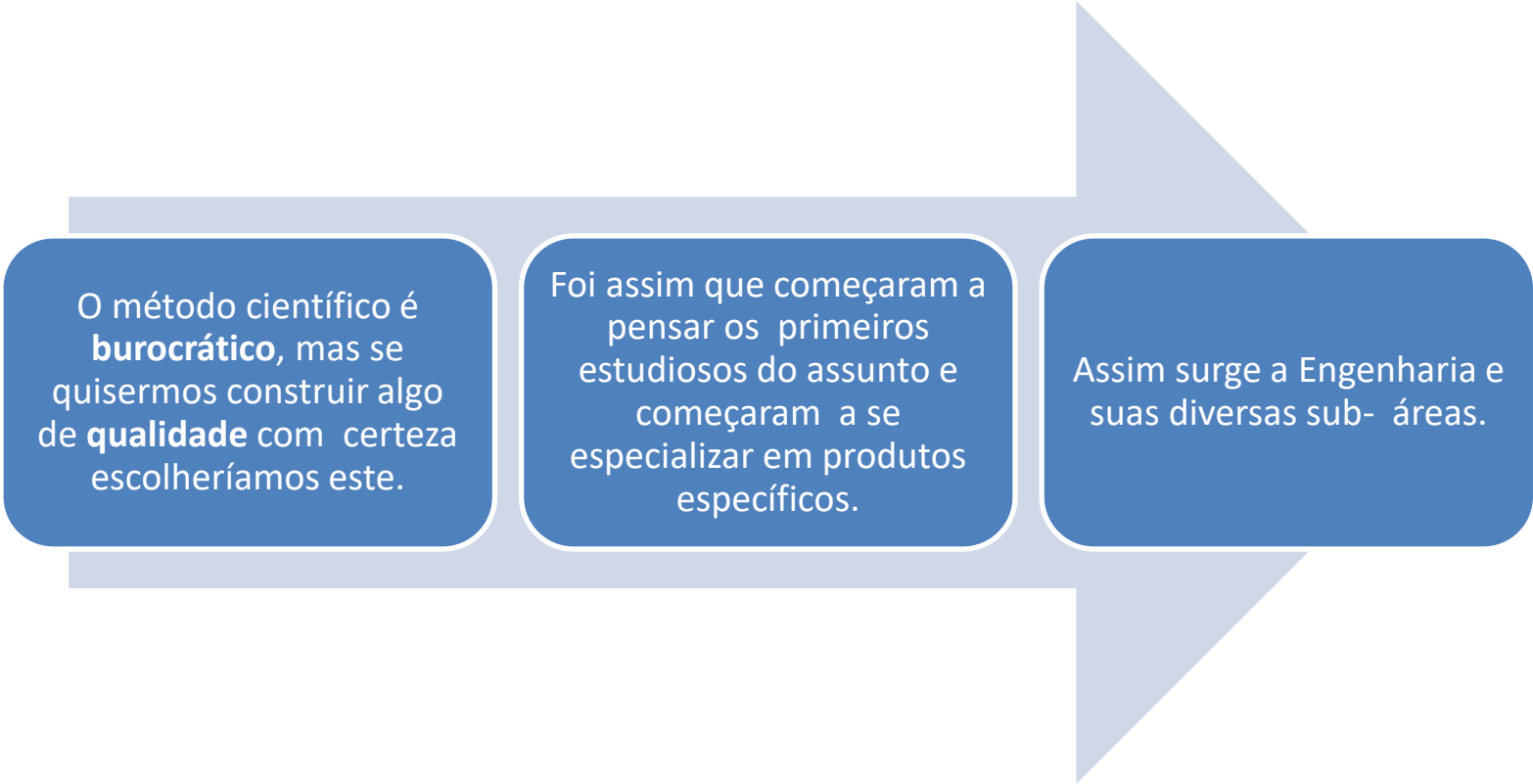
Estuda trabalhos relacionados e verifica o que poderá ser aproveitado.



Estuda trabalhos que poderão auxiliar na construção do novo produto;

Verifica padronizações e leis existentes no âmbito do projeto.

Qual o Melhor Método para Construir um Produto?



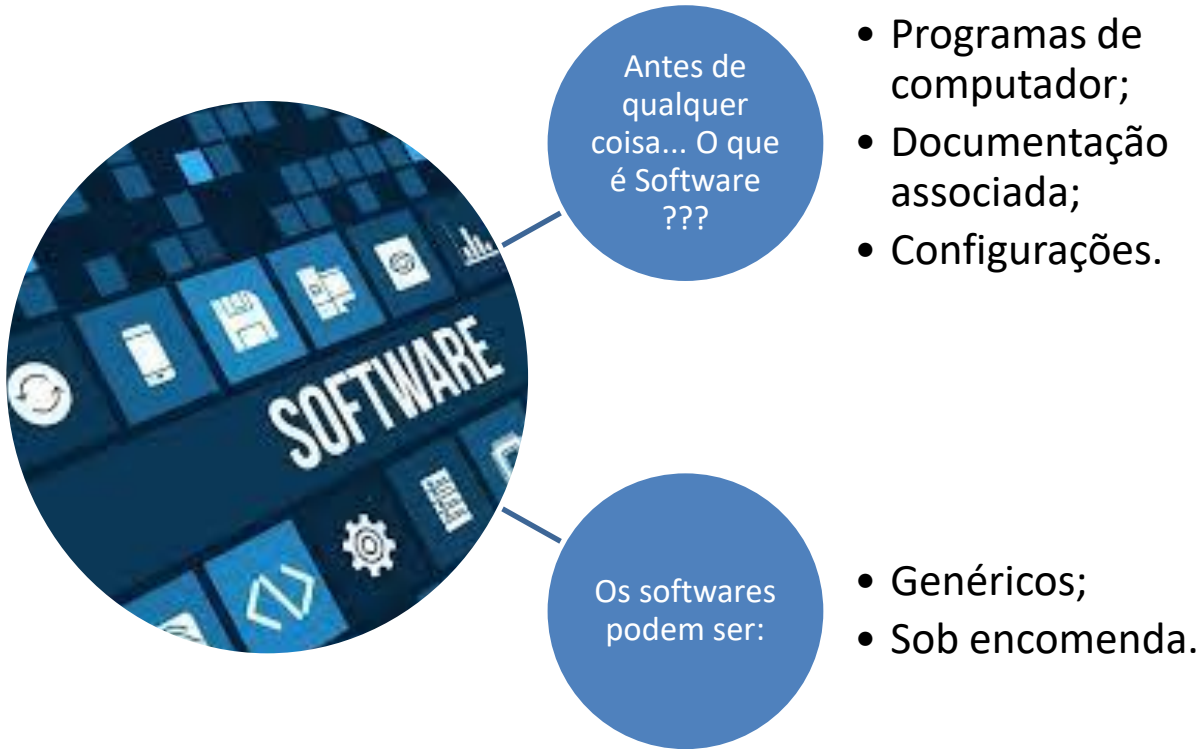
O método científico é **burocrático**, mas se quisermos construir algo de **qualidade** com certeza escolheríamos este.

Foi assim que começaram a pensar os primeiros estudiosos do assunto e começaram a se especializar em produtos específicos.

Assim surge a Engenharia e suas diversas sub- áreas.

Introdução

2



Introdução

3

E a
engenharia
de
Software ?

- Disciplina de engenharia relacionada com os aspectos da produção de Software.
- Possui uma abordagem sistemática e organizada.
- O principal objetivo é produzir um produto de maneira eficaz e de alta qualidade.

E o Software?

- Você conhece algum exemplo de software que seja bem sucedido ?
- Será que o Software acompanha a evolução do Hardware?

Engenharia x Software

Podemos construir um software da mesma maneira que construimos uma casa?

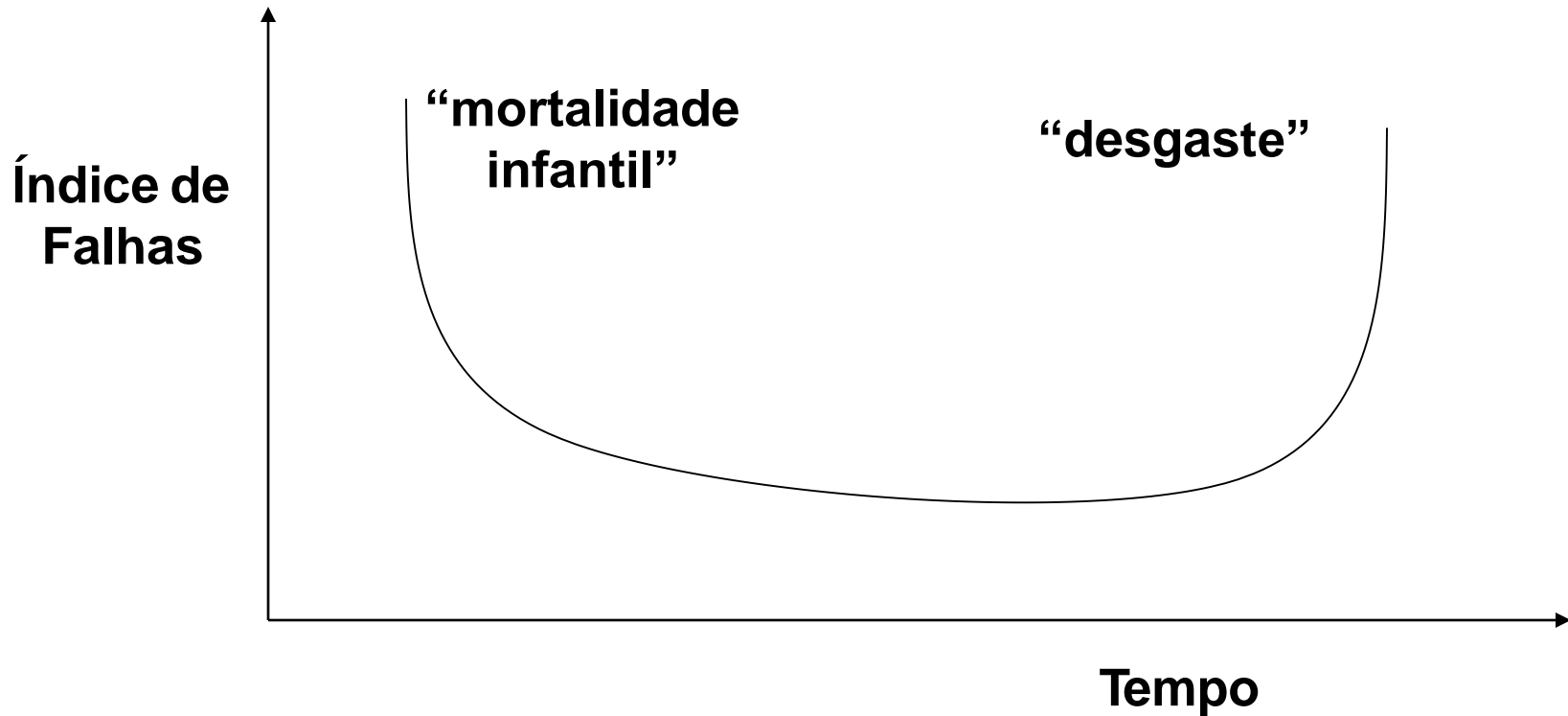
Características do Software

Não se desgasta com o tempo, mas pode se deteriorar;

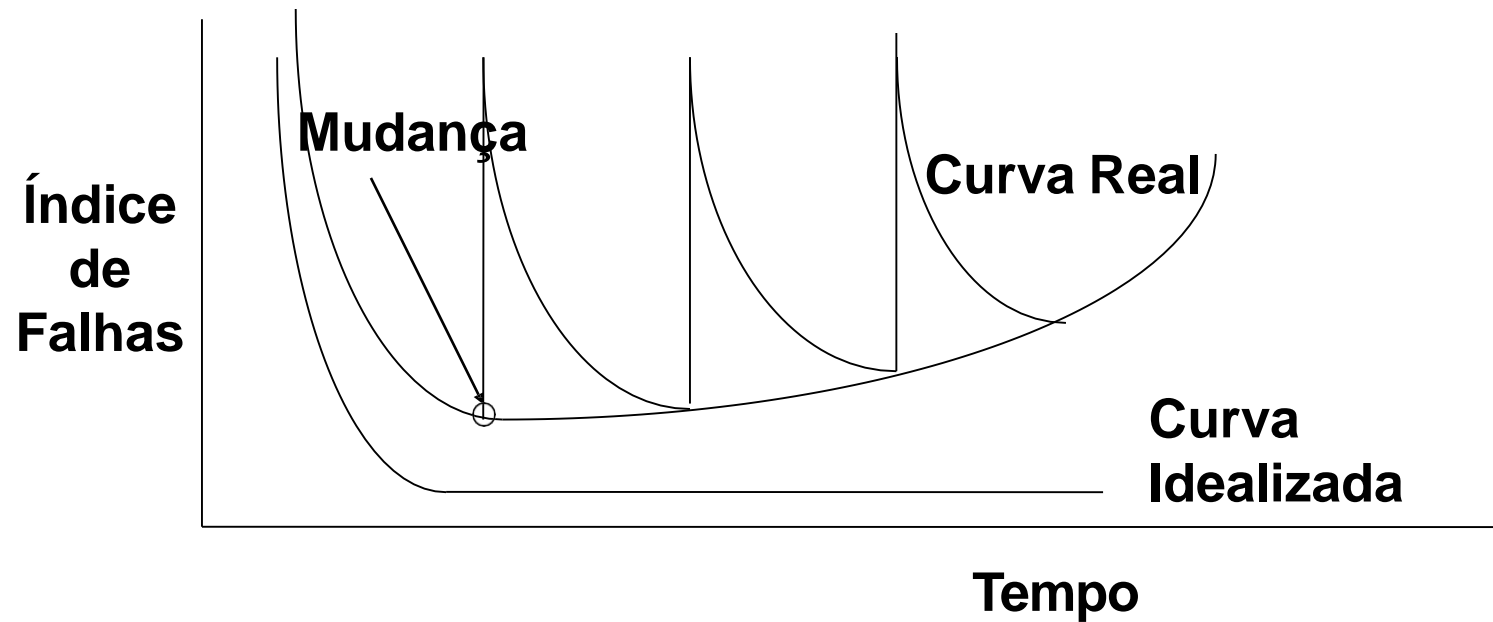
Pode ser desenvolvido ou projetado pela engenharia, não manufaturado no sentido clássico;

Principal destaque na melhoria da performance de uma Empresa.

Hardware Curva de desgaste



Software Curva de Desgaste



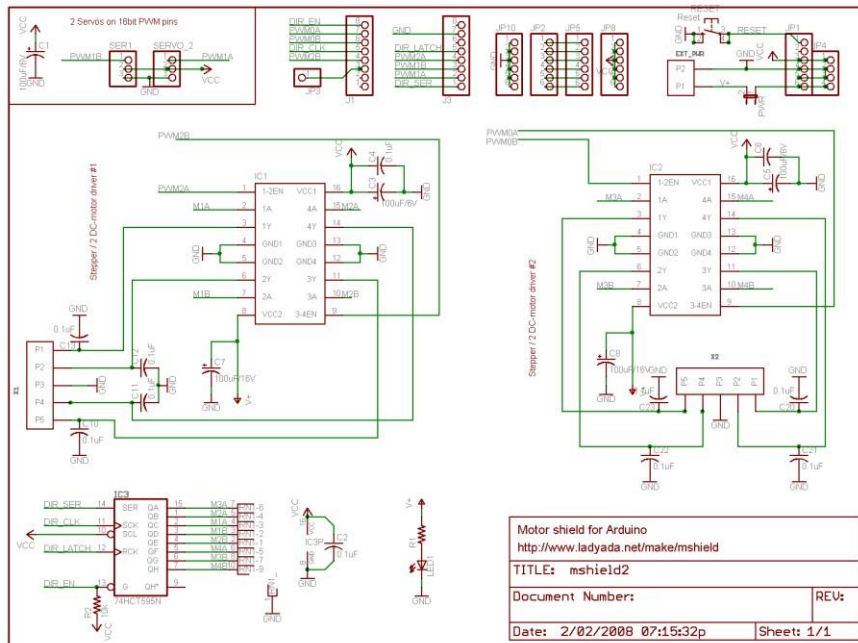
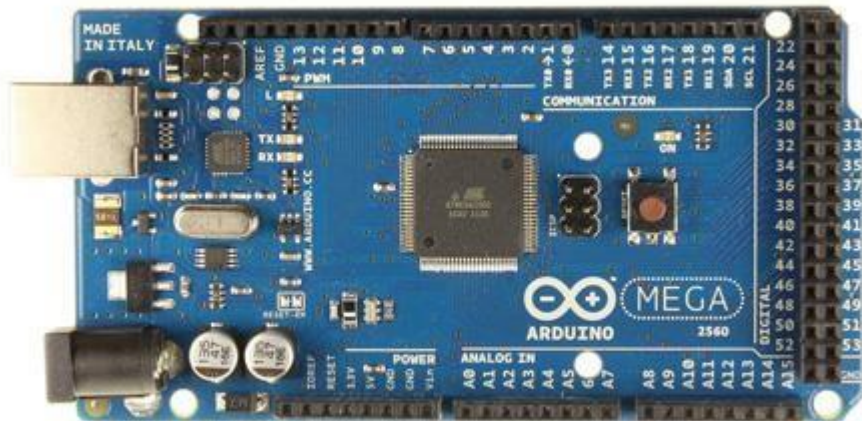
Características - Software

A maioria é feita sob medida em vez de ser montada a partir de componentes existentes;



Grande parte dos softwares produzidos no passado não possuem documentação e nem planejamento.

Como é o Hardware?



Arduino Pin Mapping

www.arduino.cc

digital pin 0 (RX)
digital pin 1 (TX)
digital pin 2
digital pin 3
digital pin 4

digital pin 5
digital pin 6
digital pin 7
digital pin 8

(RESET) PC6	1	28	PC5 (ADC5/SCL)
(RXD) PD0	2	27	PC4 (ADC4/SDA)
(TXD) PD1	3	26	PC3 (ADC3)
(INT0) PD2	4	25	PC2 (ADC2)
(INT1) PD3	5	24	PC1 (ADC1)
(XCK/T0) PD4	6	23	PC0 (ADC0)
VCC	7	22	GND
GND	8	21	AREF
(XTAL1/TOSC1) PB6	9	20	AVCC
(XTAL2/TOSC2) PB7	10	19	PB5 (SCK)
(T1) PD5	11	18	PB4 (MISO)
(AIN0) PD6	12	17	PB3 (MOSI/OC2)
(AIN1) PD7	13	16	PB2 (SS/OC1B)
(ICP1) PB0	14	15	PB1 (OC1A)

analog input 5
analog input 4
analog input 3
analog input 2
analog input 1
analog input 0

digital pin 13 (LED)
digital pin 12
digital pin 11 (PWM)
digital pin 10 (PWM)
digital pin 9 (PWM)

ATmega8

História

4

Os primeiros softwares surgiram na década de 50.

O foco dos pesquisadores era o hardware.

Software era algo inviável.



História

5

Na década de 60 surgiram os computadores baseados em CI (circuitos integrados).

Seu poder fez surgir o interesse no desenvolvimento de aplicações, até então consideradas não realizáveis.

Os pesquisadores começaram a mudar o foco para os softwares.

As aplicações se tornaram muito **complexas**.

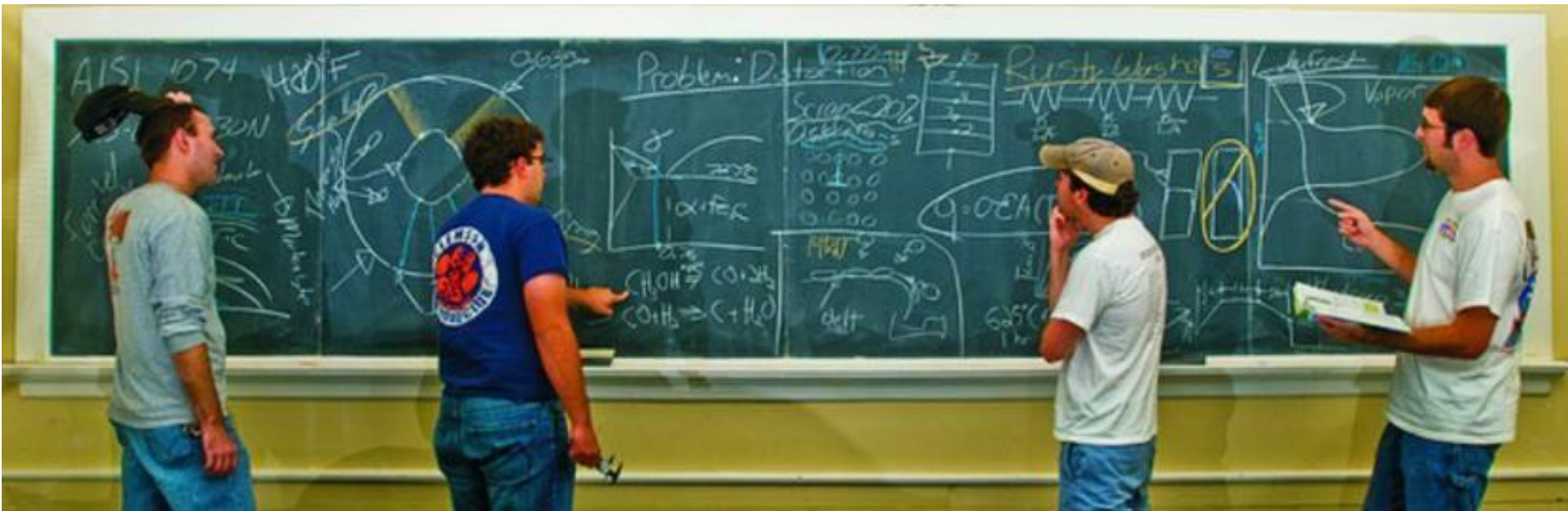
Desenvolvimento de software informal.

Resultava em sistemas caros, difíceis de manter e de desempenho insatisfatório

O Software estava em crise.

História

6



O que fazer para reverter a situação?

História

7

A **Engenharia de Software (ES)** surgiu para tentar solucionar o problema da “crise do software”.

Este conceito foi inicialmente proposto em 1968.

Foram criadas novas técnicas e métodos para controlar a complexidade dos grandes sistemas e organizar o processo de desenvolvimento de Software.



Quais os desafios da ES?

8

- ❑ Diminuir custos;
- ❑ Melhorar qualidade;
- ❑ Atender às expectativas do cliente.



Fundamentos da ES

9

A Engenharia de Software utiliza conceitos de várias áreas, como:

- **Ciências da computação:** abrange arquitetura de computadores, lógica de programação, estrutura de dados, algoritmos, etc.
- **Administração:** o engenheiro de software atua como gestor de um projeto, administrando prazos, equipe, custos, resultados etc.
- **Comunicação:** habilidade para saber se expressar com clientes ou usuários;

Camadas da ES

10

- A base da Engenharia de Software é o foco na qualidade.



Camadas da ES

11

Processo: define uma metodologia que deve ser utilizada no desenvolvimento do software. O processo é a base para o controle do gerenciamento do projeto.

Métodos: fornecem informações técnicas para o desenvolvimento do projeto. Eles podem envolver diversas tarefas como: comunicação, análise de requisitos, etc.

Ferramentas: responsáveis por fornecer suporte automatizado para o processo e os métodos.

Identificando a qualidade

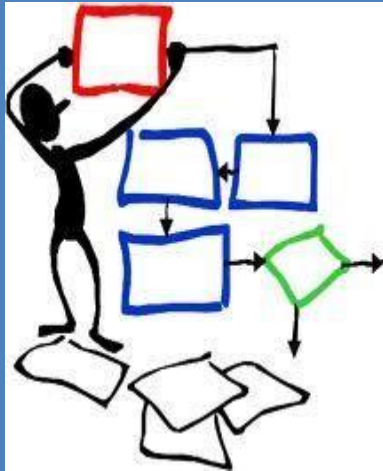
12

Como identificar a qualidade de um software?

Existem alguns atributos que indicam qualidade em um software:

- **Facilidade de manutenção:** o software deve evoluir para atender às necessidades de mudança do cliente.
- **Confiança:** o software deve ser confiável.
- **Eficiência:** o sistema deve realizar suas atividades utilizando a menor quantidade de recursos possíveis.
- **Usabilidade:** o sistema deve ser compreensível, usável e compatível com outros sistemas.

Processo de Software



Abrange um conjunto de três elementos fundamentais: **Métodos, Ferramentas e Procedimentos** para projetar, construir e manter grandes sistemas de software de forma profissional

Processo de Software

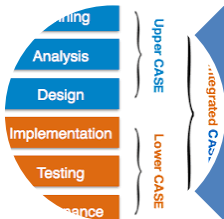
13



Segundo SOMERVILLE, Processo de Software é um conjunto de atividades que leva à produção de um produto de Software.



Processos complexos.



Podemos contar com ferramentas de apoio com o objetivo de automatizar esses processos, conhecidas como ferramentas CASE (*Computer-Aided Software Engineering*).



Por se tratar de um objeto subjetivo e da imensa diversidade de processos de software, as tentativas de automatização têm tido sucesso limitado.

Processo de Software

14

Existem diversos processos de software.

Não existe um processo ideal.

Existem atividades comuns aos processos de software, são elas:

- **Especificação de software:** Definição das funcionalidades e das restrições sobre suas operações.
- **Projeto e implementação de software:** produção do software que atenda às especificações solicitadas.
- **Validação de software:** O software deve ser validado para garantir que ele faça o que o cliente deseja.
- **Evolução de software:** o software deve evoluir para atender às necessidades mutáveis do cliente.

Modelos de Processo de Software

15

É uma
**representação
abstrata de um
processo de
software.**



Representa um
processo sob uma
determinada
perspectiva.



Os principais
modelos de
processos são:

- Modelo cascata
- Desenvolvimento evolucionário
- Engenharia de software baseada em componentes
- Iterativo:
 - Iterativo-incremental
 - Espiral

Modelos de Processo de Software

16

Por quê utilizar um modelo de processo de software?

- Padroniza o desenvolvimento de software.
- Padronização dos artefatos de software.
- Melhora a comunicação da equipe.
- Consequentemente, agrega qualidade ao software.

Modelo Cascata

17

Foi o primeiro modelo de processo de software proposto.

É um modelo sequencial.

Devido ao encadeamento de suas fases, o modelo ficou conhecido como modelo cascata.

Uma nova fase só pode ser iniciada com a finalização da anterior.

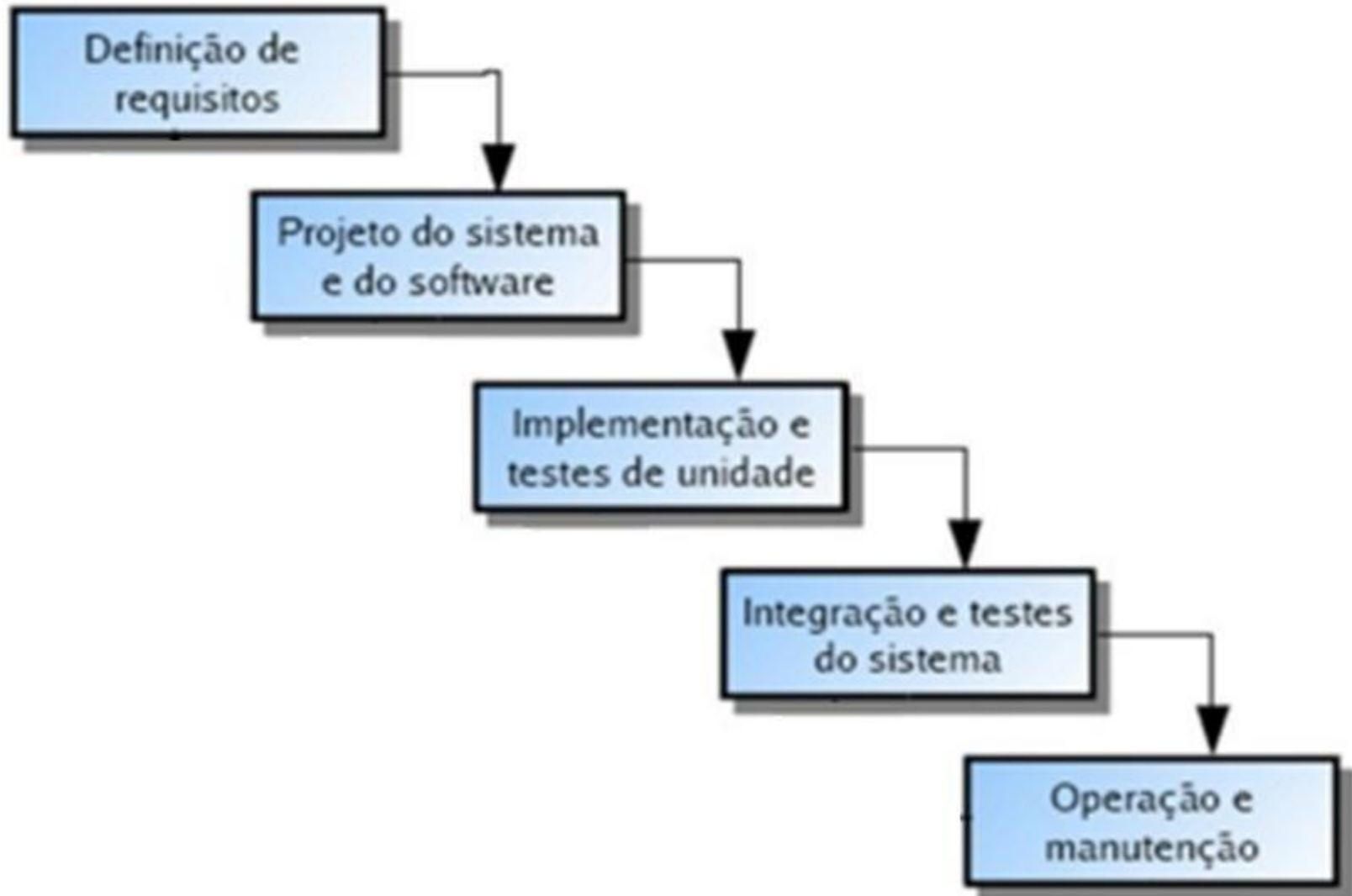
Possui 5 fases definidas:

- Requisitos, projeto, implementação, integração e manutenção.

Ao final de cada fase, um ou mais documentos são aprovados.

Modelo Cascata

18



Modelo Cascata

19

- Documentação rígida em cada fase do processo.
- Aderência a outros modelos de processo
- Pode ser combinado com outros modelos.

Vantagens

- Projetos reais raramente seguem um fluxo sequencial.
- É difícil para o cliente definir todos os requisitos no início do processo.
- Difícil adequação à mudanças de requisitos.
- Uma versão executável somente ficará pronta no final do processo.

Desvantagens

Desenvolvimento evolucionário

20



Desenvolvimento evolucionário

21

Existem 2 tipos de desenvolvimento evolucionário:

Desenvolvimento exploratório:

- Trabalha diretamente com o cliente a fim de explorar seus requisitos e entregar um sistema final.
- Possui início com os requisitos compreendidos.
- O sistema evolui com o acréscimo de características, à medida que são propostas pelo cliente.

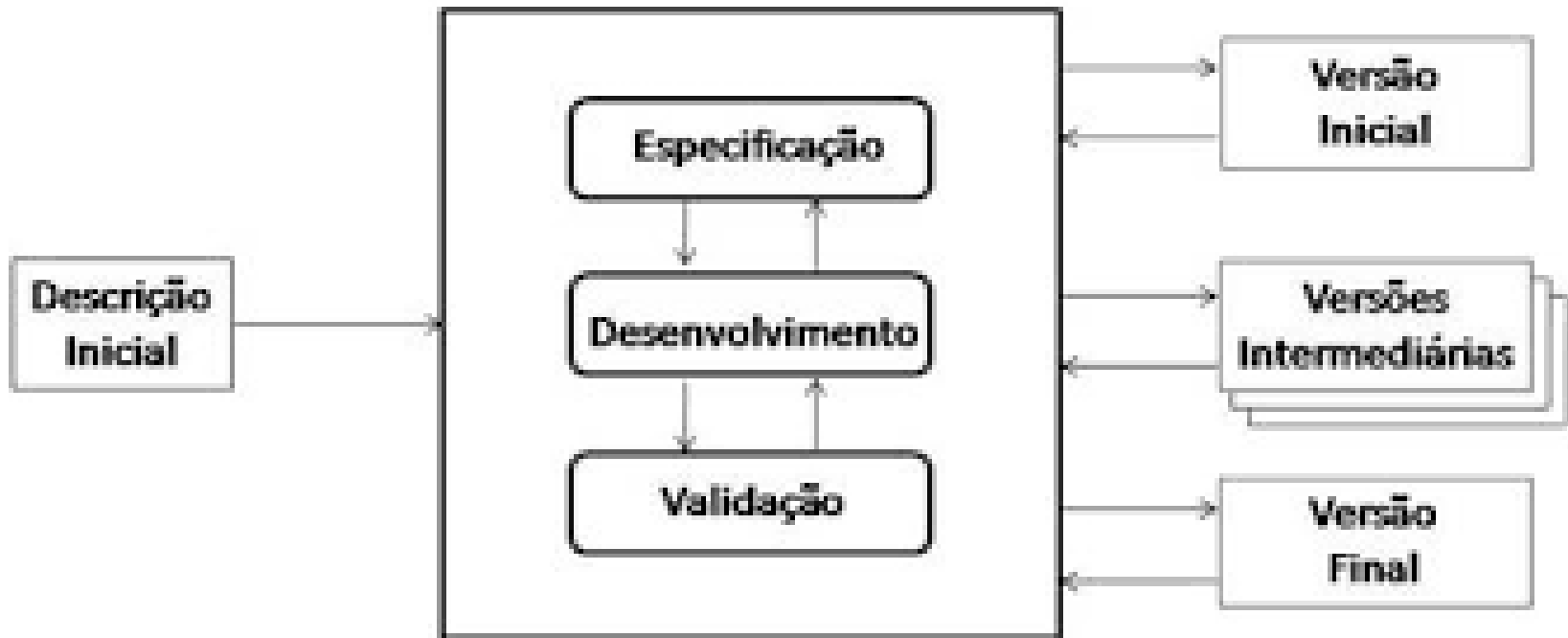
Prototipação *throwaway*:

- O objetivo deste tipo de desenvolvimento é utilizar um protótipo (experimentos) para compreender melhor requisitos mal entendidos.

Desenvolvimento evolucionário

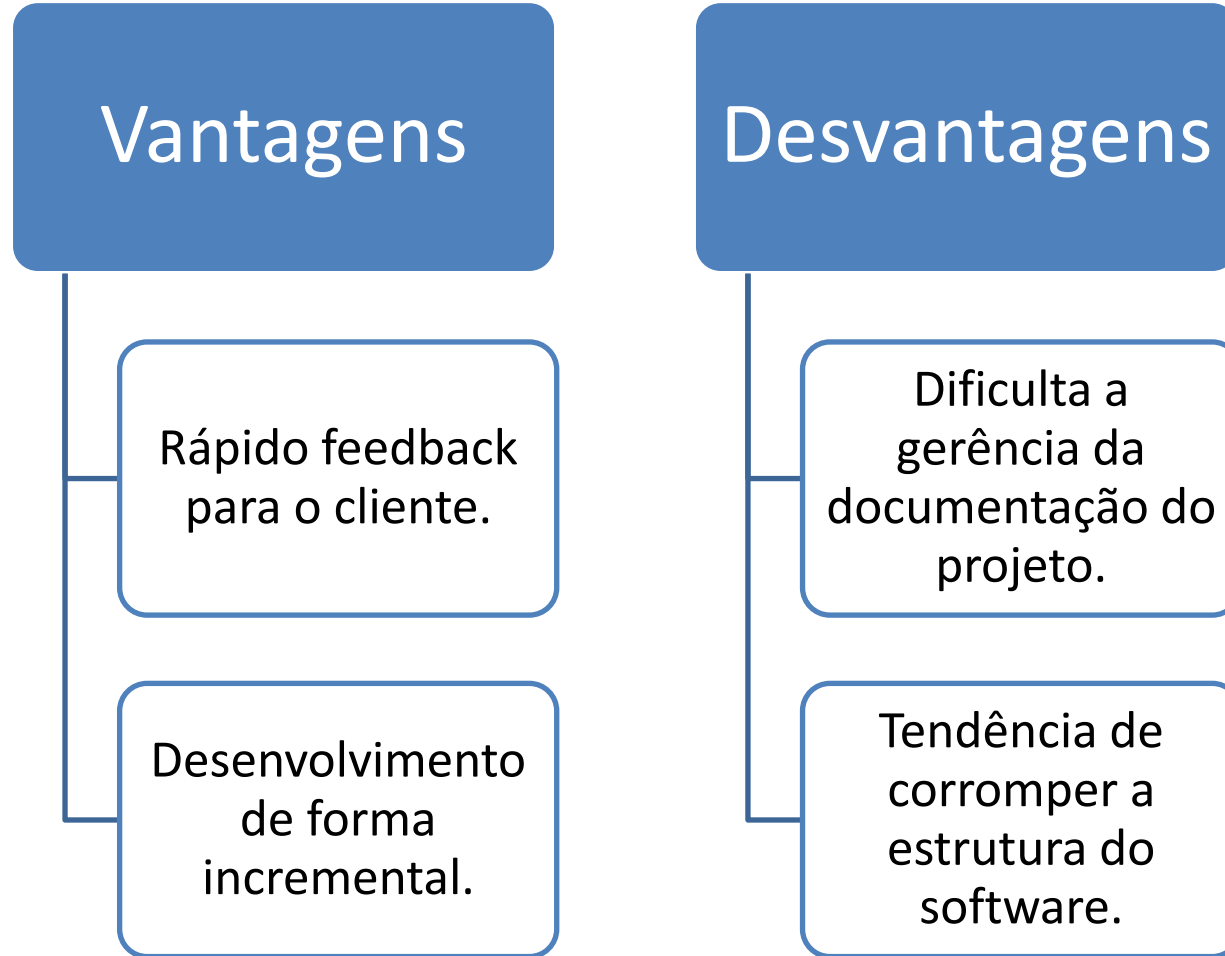
22

Atividades Simultâneas



Desenvolvimento evolucionário

23



ES baseada em componentes

24

Apoiada pela orientação a objetos:

- Base na reusabilidade de códigos ou componentes adequadamente projetadas e implementadas.
- São reutilizáveis em diferentes aplicações e arquiteturas.

Compõe a aplicação de software a partir de componentes previamente preparados.

Resumindo, é o processo de **definição, implementação, integração ou composição** de componentes independentes.

ES baseada em componentes

25

O que são componentes?

- É uma entidade executável e independente
- Possui serviços que estão disponíveis por meio de uma interface, por onde suas interações acontecem.

Principais características de um componente:

- Padronizado
- Independente
- Passível de composição
- Implantável
- Documentado

ES baseada em componentes

26

Interfaces do componente:

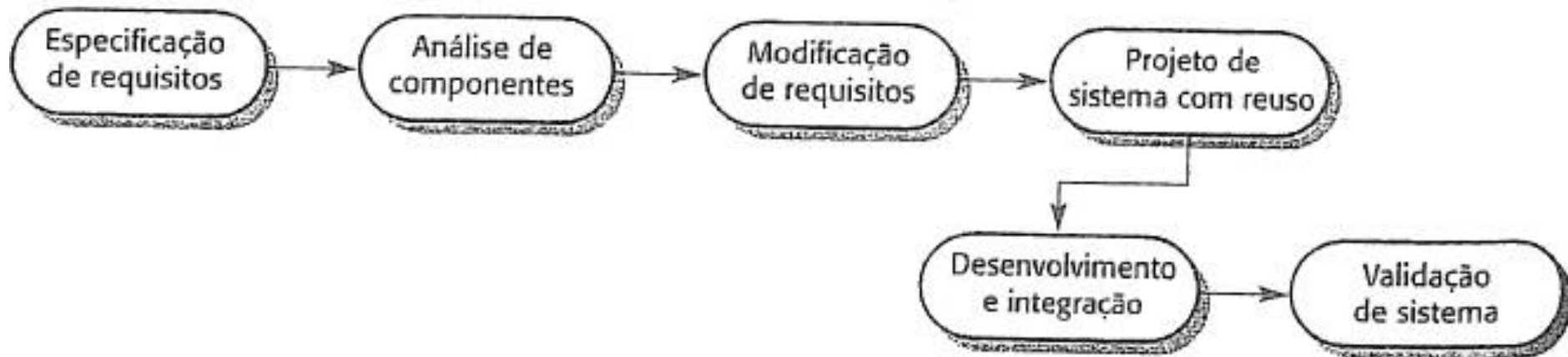
- **Requires:** serviços que devem ser fornecidos ao componente.
- **Provides:** serviço fornecido pelo componente.



ES baseada em componentes

27

Processo de desenvolvimento de software baseado em componentes



Modelos Iterativos

28

Esse tipo de modelo repete partes do processo à medida que os requisitos do sistema evoluem.

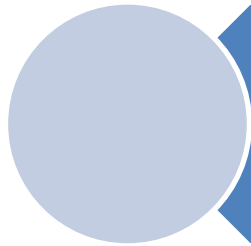
Mistura conceitos dos modelos vistos anteriormente.

Os principais modelos iterativos são:

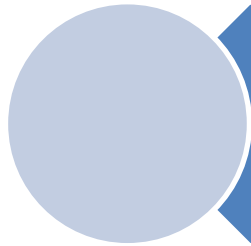
- Espiral
- Iterativo-incremental

Modelo Espiral

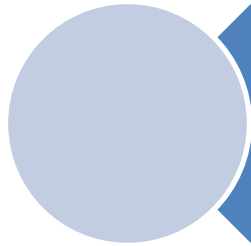
29



Em vez de representar o processo de software como uma sequência de atividades, o processo é representado como uma espiral.



Cada volta da espiral apresenta uma fase do processo de software.



Ex: a volta mais interna pode estar relacionado à viabilidade do sistema; a volta seguinte pode estar relacionada à definição de requisitos e assim por diante.

Modelo Espiral

30

Cada volta da espiral é dividida em 4 setores:

- **Definição dos objetivos:** definição dos objetivos específicos para essa fase do processo.
- **Avaliação e redução de riscos:** identificar os riscos, analisar detalhadamente cada um e tomar providências para reduzir sua probabilidade.
- **Desenvolvimento e validação:** aplicar as fases do desenvolvimento.
- **Planejamento:** revisão do projeto e planejamento da próxima volta da espiral, se for o caso.

Modelo Espiral

31

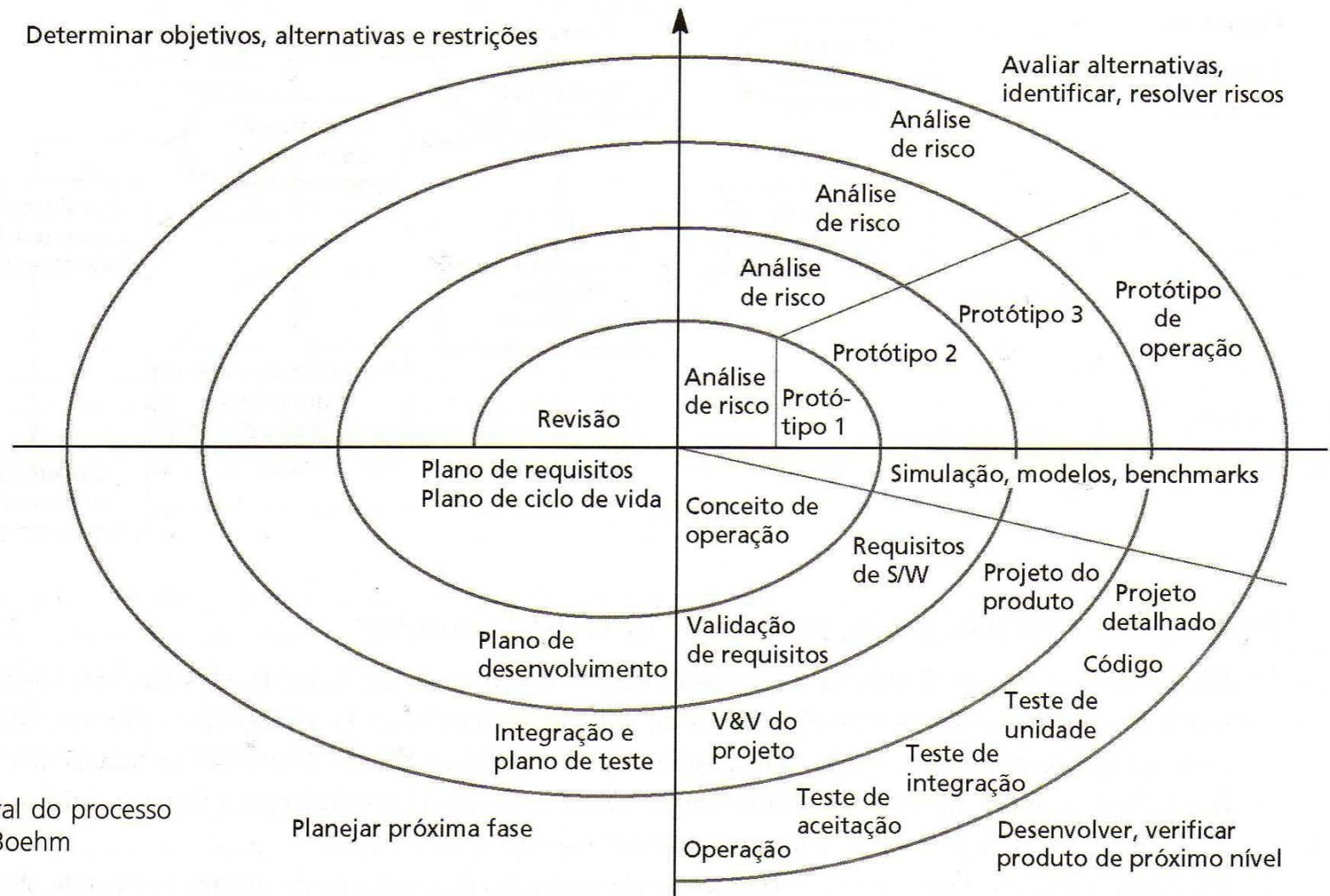


Figura 4.5

Modelo em espiral do processo de software de Boehm (©IEEE, 1988).

Iterativo-Incremental

32

Foi sugerida com o objetivo de reduzir o “retrabalho” no processo de desenvolvimento.



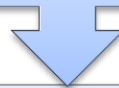
O cliente não precisa detalhar TODOS os requisitos na fase inicial.



Priorização das funcionalidades mais importantes.



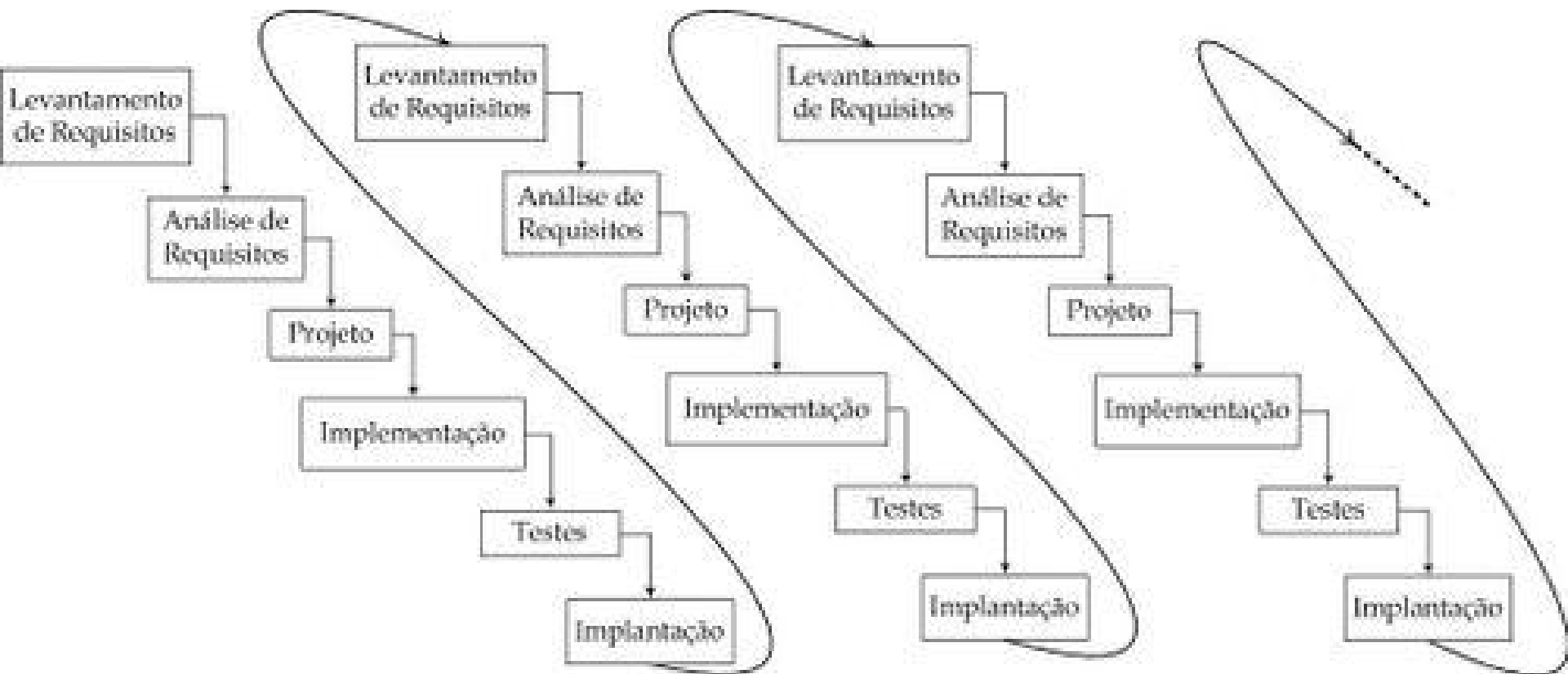
O conjunto de funcionalidades que serão desenvolvidos são conhecidos como incremento.



Uma vez que o incremento é finalizado, o produto pode ser colocado em operação.

Iterativo-Incremental

33



Iterativo-Incremental

34

Vantagens

- O cliente não precisa esperar até que todo sistema seja entregue.
- Risco menor de fracasso do sistema.
- Priorização de funcionalidades mais importantes.

Desvantagens

- Os incrementos devem ser relativamente pequenos.
- Dificuldade no mapeamento de funcionalidades com dependência.

Exercícios

35

1) Considere o seguinte problema encontrado em projetos de desenvolvimento de software: Projetos reais raramente seguem um fluxo sequencial. Apesar de um modelo linear poder acomodar a iteração, ele o faz indiretamente. Como resultado, as modificações podem causar confusão à medida que a equipe de projeto prossegue. Esse é um dos problemas que são algumas vezes encontrados quando é aplicado o modelo de desenvolvimento.

- ☐ em cascata.
- ☐ ágil.
- ☐ espiral.
- ☐ incremental.
- ☐ unificado.

Exercícios

36

2) O processo de desenvolvimento de software conhecido como modelo em espiral (Modelo espiral de Boehm), divide cada volta da espiral em quatro setores

- Correto
- Errado

Exercícios

37

- 3) O modelo em cascata inclui 5 estágios considerados fundamentais para o desenvolvimento de um software: a análise e definição de requisitos, o projeto de sistema e software, a implementação e o teste de unidade, a integração e o teste de sistema e a operação e manutenção. Apesar disso, o modelo em cascata tem como desvantagem a
- a) documentação produzida em cada estágio.
 - b) aderência a outros modelos de processo de engenharia.
 - c) dificuldade de reação a mudanças de requisitos do usuário.
 - d) falta de estruturação para desenvolvimento de software.

Exercícios

38

4) A cerca da Engenharia de Software, assinale a opção correta:

- a) A engenharia de software, que tem como alicerce os métodos, é formada de várias camadas, tais como ferramentas, métodos e processo.
- b) Apesar de importante no contexto de produtos, a qualidade não é tratada na engenharia de software e, sim, no gerenciamento de projetos.
- c) Análise e projeto são duas ações da engenharia de software. A definição de arquitetura e componentes são tarefas da análise, enquanto levantamento e especificação de requisitos são tarefas do projeto.
- d) A engenharia de software visa obter softwares econômicos que sejam confiáveis e que trabalhem eficientemente em máquinas reais.
- e) A engenharia de software aplica abordagem sistemática, disciplinada e não quantificável para desenvolvimento e manutenção de software.

Exercícios

39

Com relação aos Conceitos de Desenvolvimento de Sistemas, julgue os itens a seguir (Certo ou Errado) e justifique a sua escolha:

- 5 Os princípios de engenharia de software definem a necessidade de formalidades para reduzir inconsistências e a decomposição para lidar com a complexidade.
- 6 Para garantir o desenvolvimento de qualidade, é suficiente que a equipe tenha as ferramentas mais atuais de engenharia de

Com relação aos princípios de engenharia de software, julgue o item a seguir (Certo ou Errado) e justifique a sua escolha:

- 7 Um modelo de processo de software descreve os processos que são realizados para atingir o seu desenvolvimento. A notação para as tarefas, os artefatos, os atores e as decisões varia conforme o modelo de processo utilizado.

Exercícios

36

- 8) Pesquise sobre a curva de desgaste do software e do hardware e explique os gráficos dos slides 14 e 15. Coloque a referência bibliográfica relacionada a essa pesquisa.