

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E
TECNOLOGIA DO RIO GRANDE DO NORTE
CAMPUS JOÃO CÂMARA

ENGENHARIA DE SOFTWARE

Nickerson Fonseca Ferreira
nickerson.ferreira@ifrn.edu.br

Introdução

2

- Antes de qualquer coisa... O que é Software ???
 - Programas de computador
 - Documentação associada
 - Configurações
- Os softwares podem ser:
 - Genéricos
 - Sob encomenda

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.

História

4

- ❑ Os primeiros softwares surgiram na década de 50.
- ❑ O foco dos pesquisadores era o hardware.
- ❑ Software era algo inviável.



História

- 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

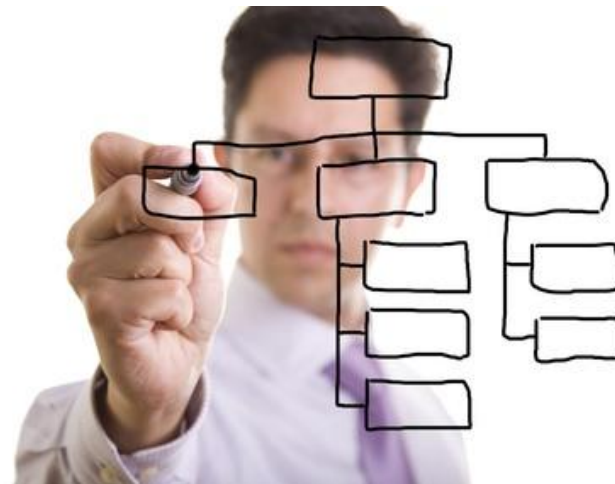


O que fazer para reverter a situação?

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

- 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

- **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 num 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

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.
- Conseqüentemente, agrega qualidade ao software.

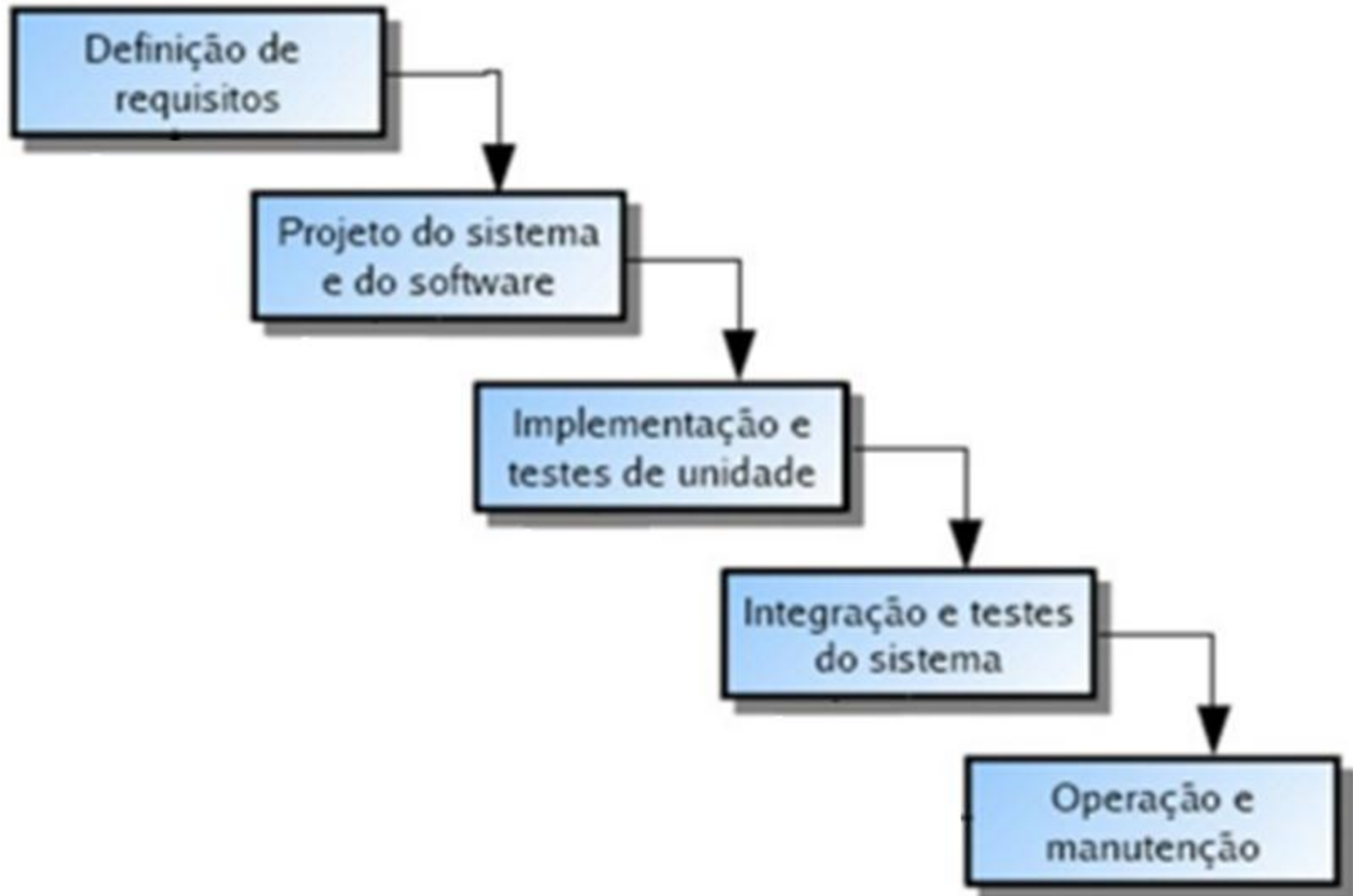
Modelo Cascata

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

□ Vantagens

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

□ Desvantagens

- 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.

Desenvolvimento evolucionário

20

- Tem como base uma implementação inicial e realizar aprimoramentos por meio de várias versões.
- Todo o trabalho é realizado concorrentemente.
- Rápido retorno (*feedback*) para o cliente.

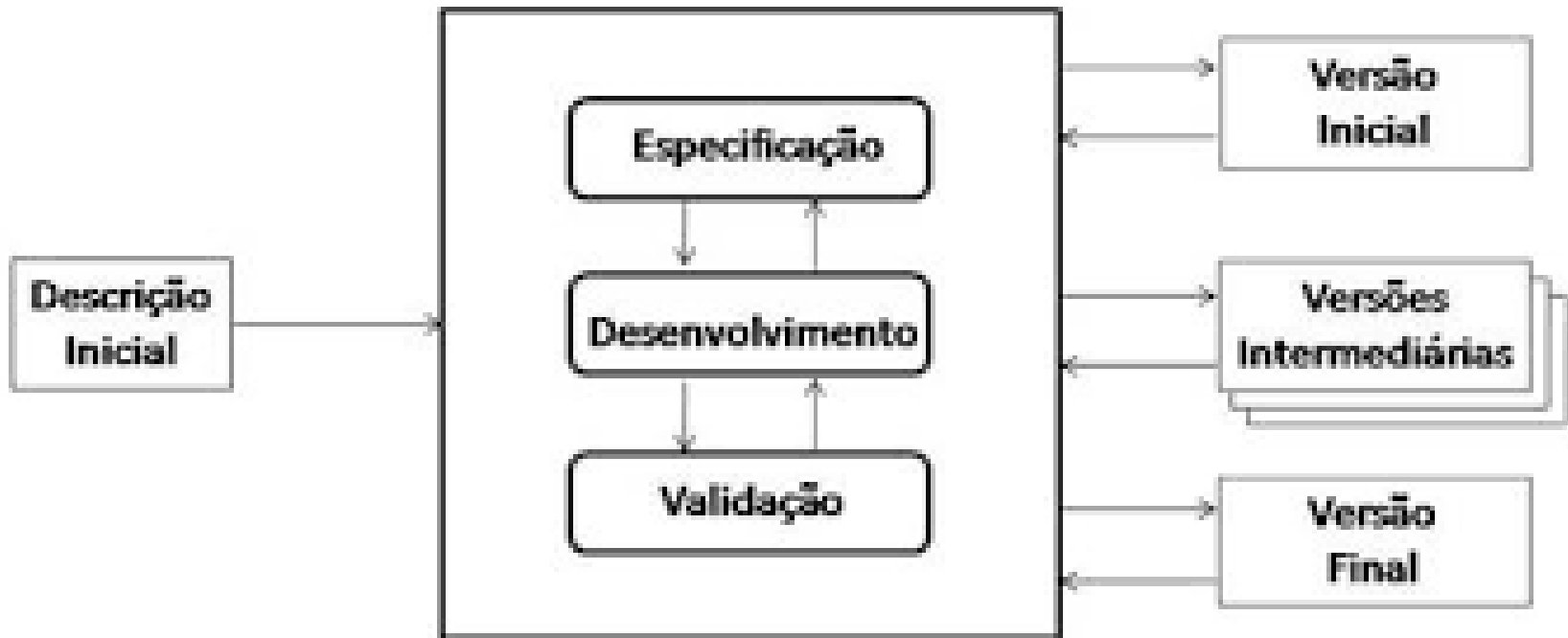
Desenvolvimento evolucionário

- 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

□ Vantagens

- Rápido feedback para o cliente.
- Desenvolvimento de forma incremental.

□ Desvantagens

- Dificulta a gerência da documentação do projeto.
- Tendência de corromper a estrutura do software.

ES baseada em componentes

- 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

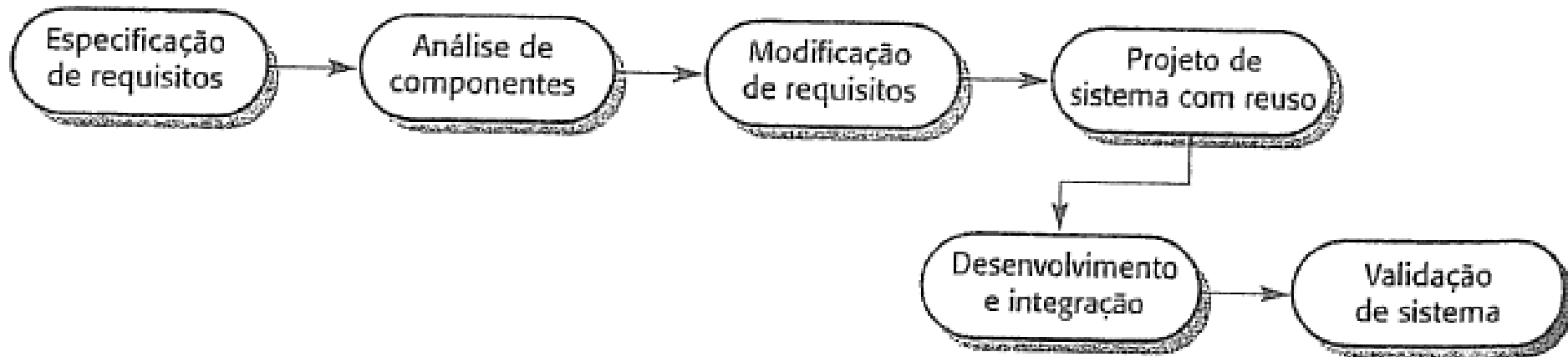
- Interfaces do componente:
 - **Requires**: serviços que devem ser fornecidos ao componente.
 - **Provides**: serviço fornecido pelo componente.



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

- 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

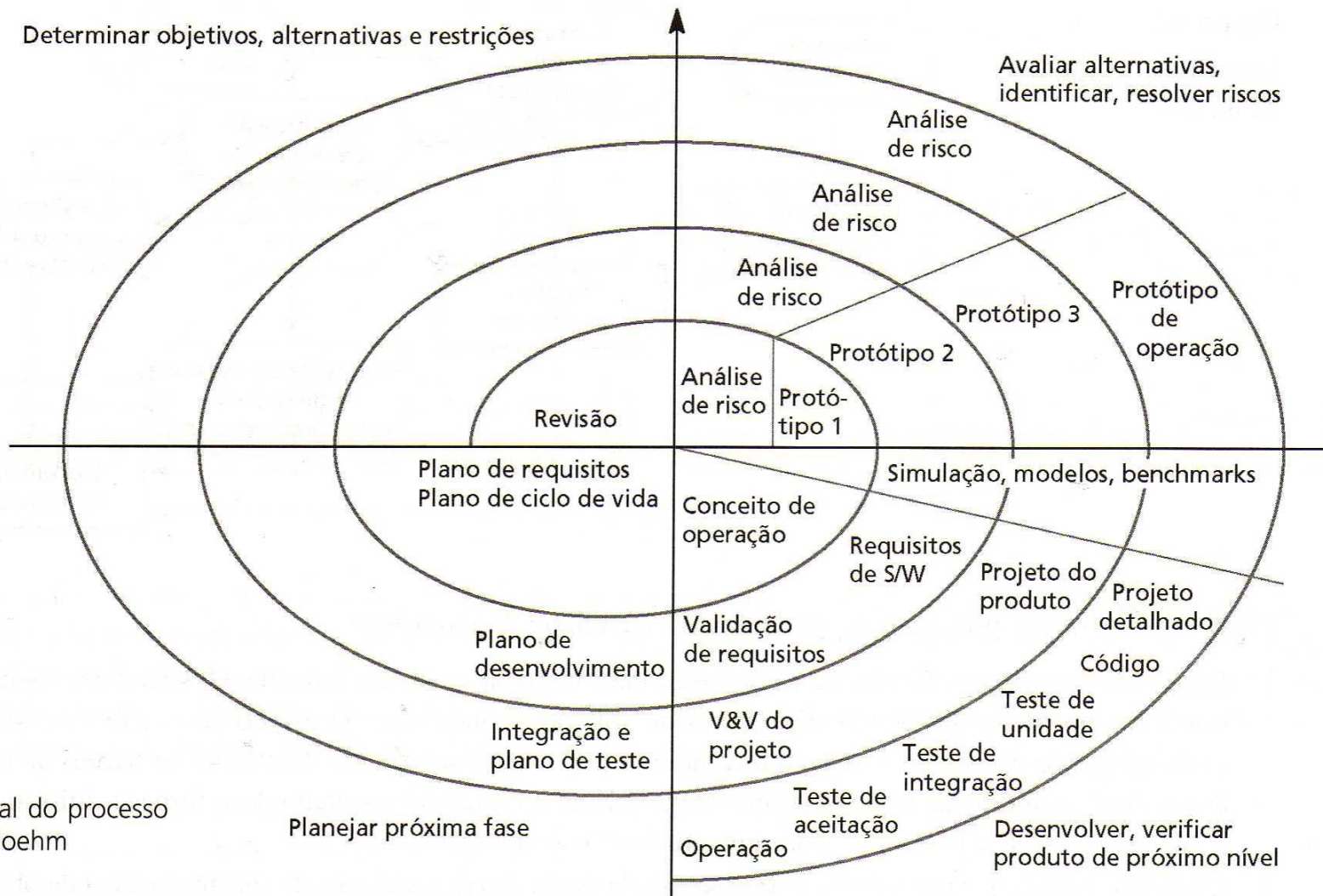


Figura 4.5

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

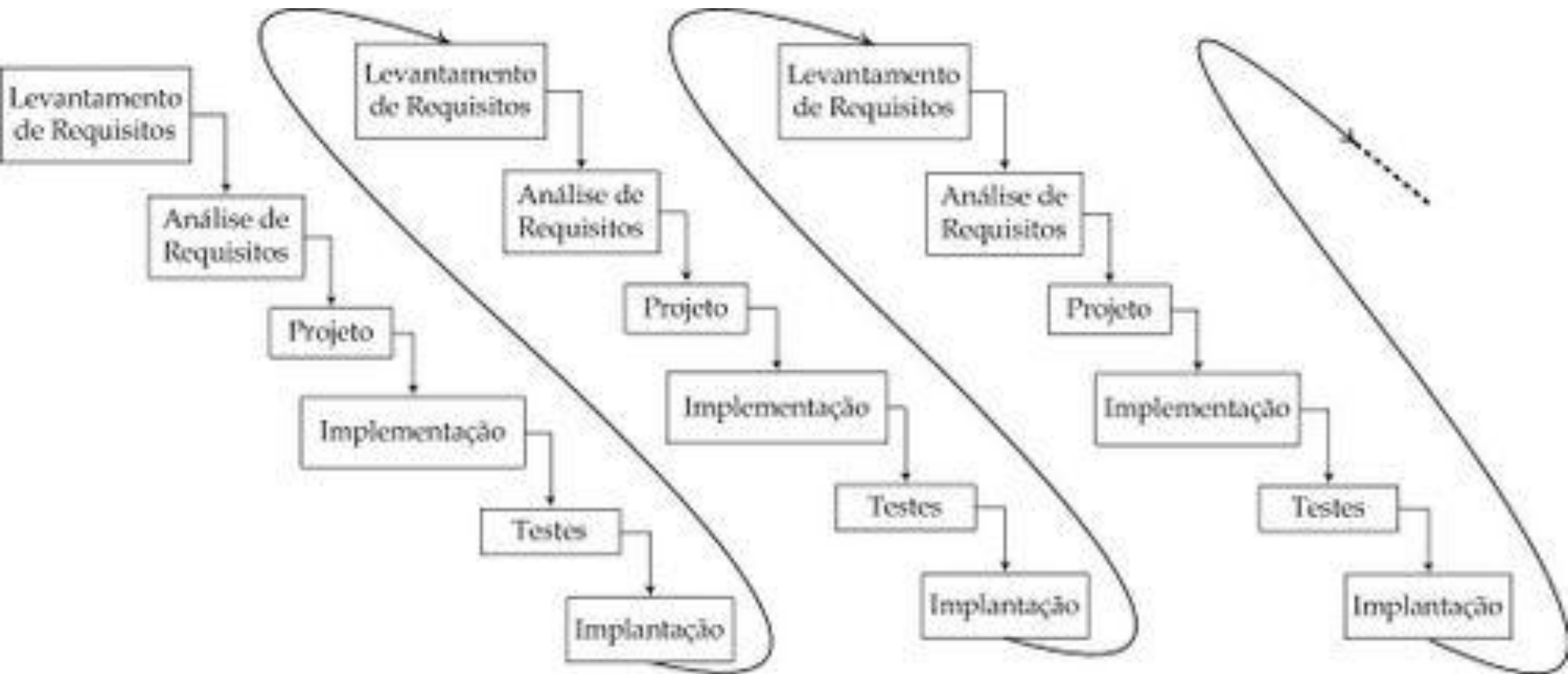
Iterativo-Incremental

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.

- a) em cascata.
- b) ágil.
- c) espiral.
- d) incremental.
- e) 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

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 (SECGE – PE – ACI – 2010 – Cespe) Acerca 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

(Banco da Amazônia – TI – 2010 – Cespe) Com relação aos conceitos de desenvolvimento de sistemas, julgue os itens a seguir.

- 5 [74] Os princípios de engenharia de software definem a necessidade de formalidades para reduzir inconsistências e a decomposição para lidar com a complexidade.
- 6 [75] Para garantir o desenvolvimento de qualidade, é suficiente que a equipe tenha as ferramentas mais atuais de engenharia de software e os melhores computadores.

(Embasa – Desenvolvimento – 2010 – Cespe) Acerca de princípios de engenharia de software, julgue os itens a seguir

- 7 [67] 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

40

□ **Gabarito**

- 1) A
- 2) C
- 3) C
- 4) D
- 5) C
- 6) E
- 7) C