PROF. BRUNO DE CASTRO H. SILVA

APS

PROCESSO DE DESENVOLVIMENTO DE SOFTWARE

SUMÁRIO

- Introdução;
- Etapas de Desenvolvimento;
- Componente Humano;
- Modelos de Ciclo de Vida;
- Organização do Modelo Iterativo Incremental;
- Prototipagem;
- Ferramentas Case;

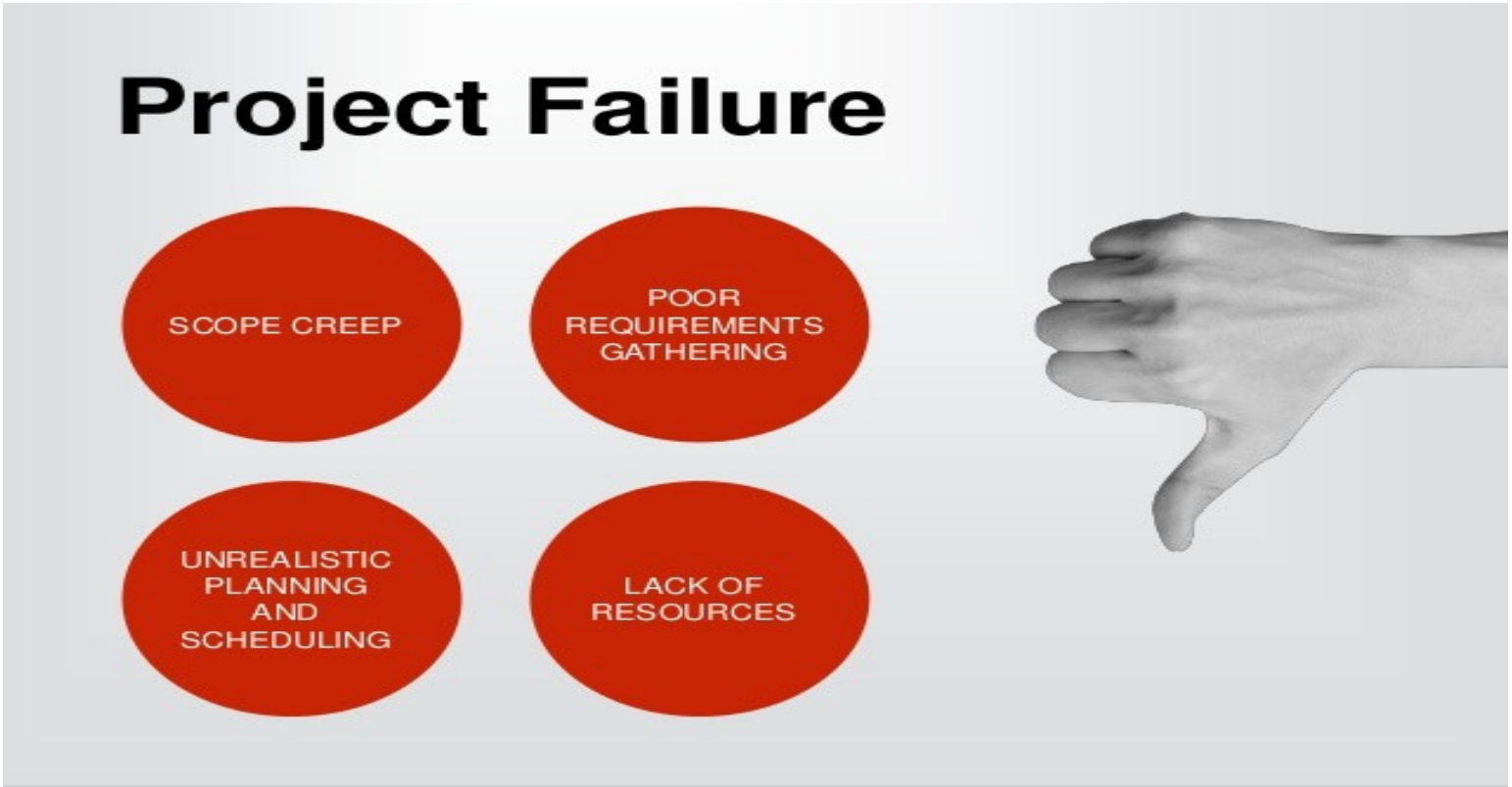
INTRODUÇÃO

INTRODUÇÃO

- O desenvolvimento de software é uma atividade complexa, sendo essa complexidade influenciada por fatores como: Software, Hardware, RH etc.

INTRODUÇÃO

- Isso se reflete no alto número de projetos de software que não chegam ao fim, ou que extrapolam recursos de tempo e de dinheiro alocados.

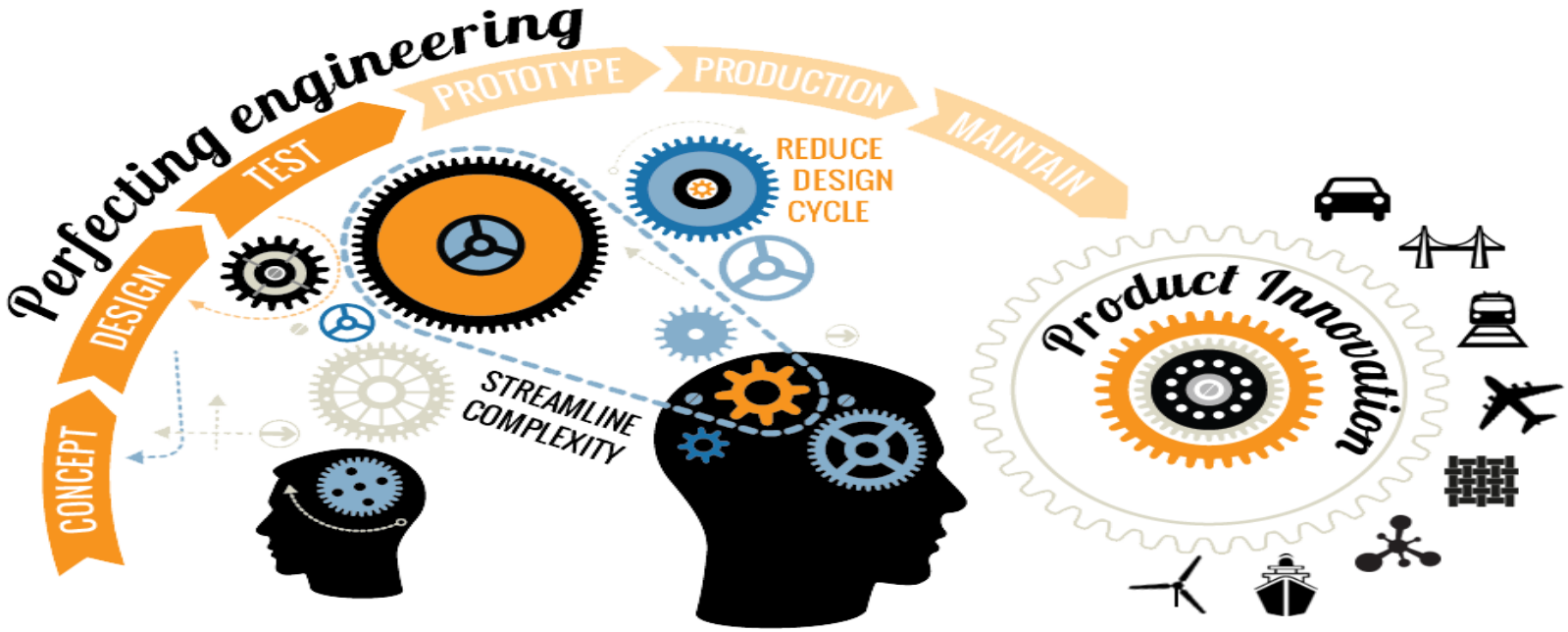


INTRODUÇÃO

- *Chaos Report*, um estudo clássico feito pelo *Standish Group* sobre projetos de desenvolvimento (*Chaos*, 1994):
  - Porcentagem de projetos que terminam dentro do prazo estimado: 10%.
  - Porcentagem de projetos que são descontinuados antes de chegarem ao fim: 25%.
  - Porcentagem de projetos acima do custo esperado: 60%.
  - Atraso médio nos projetos: um ano.

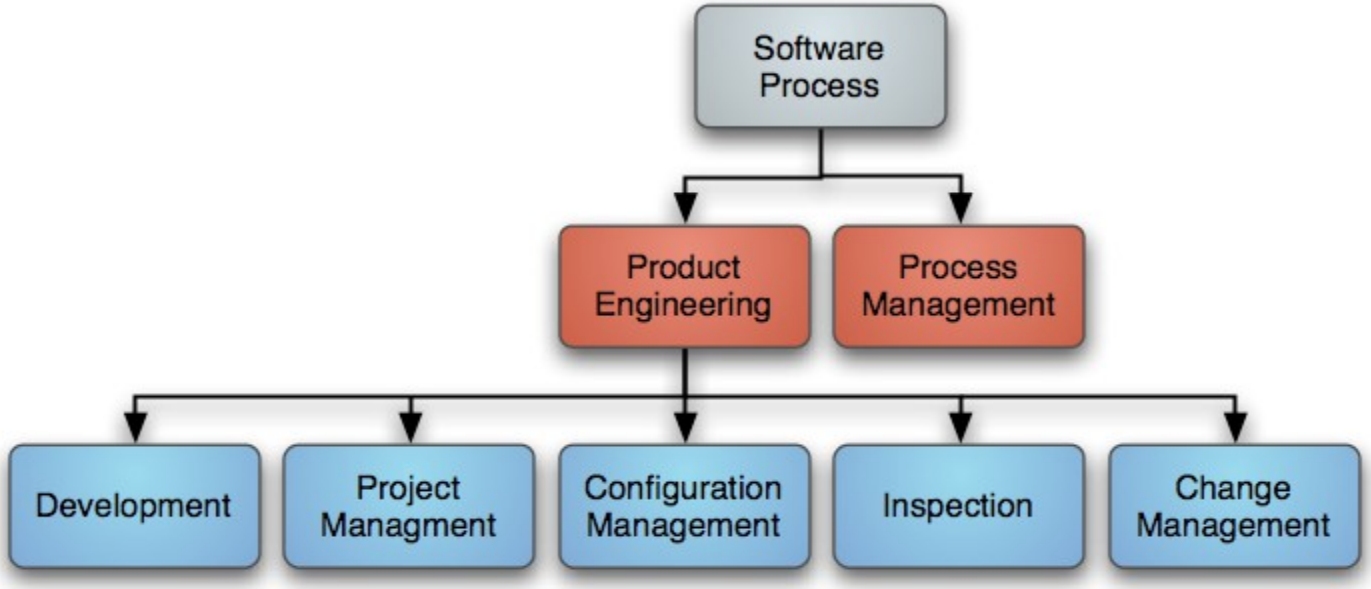
INTRODUÇÃO

- Tentativas de lidar com essa complexidade e de minimizar os problemas envolvidos no desenvolvimento de software envolvem a definição de *processos de desenvolvimento de software*.



INTRODUÇÃO

- Um processo de desenvolvimento de software compreende todas as atividades necessárias para definir, desenvolver, testar e manter um produto de software.



INTRODUÇÃO

- Eis alguns objetivos de um processo de desenvolvimento:
  - Definir quais as atividades a serem executadas ao longo do projeto;
  - Especificar quando, como e por quem tais atividades serão executadas;
  - Prover pontos de controle para verificar o andamento do desenvolvimento;
  - Padronizar a forma de desenvolver software em uma organização.

INTRODUÇÃO

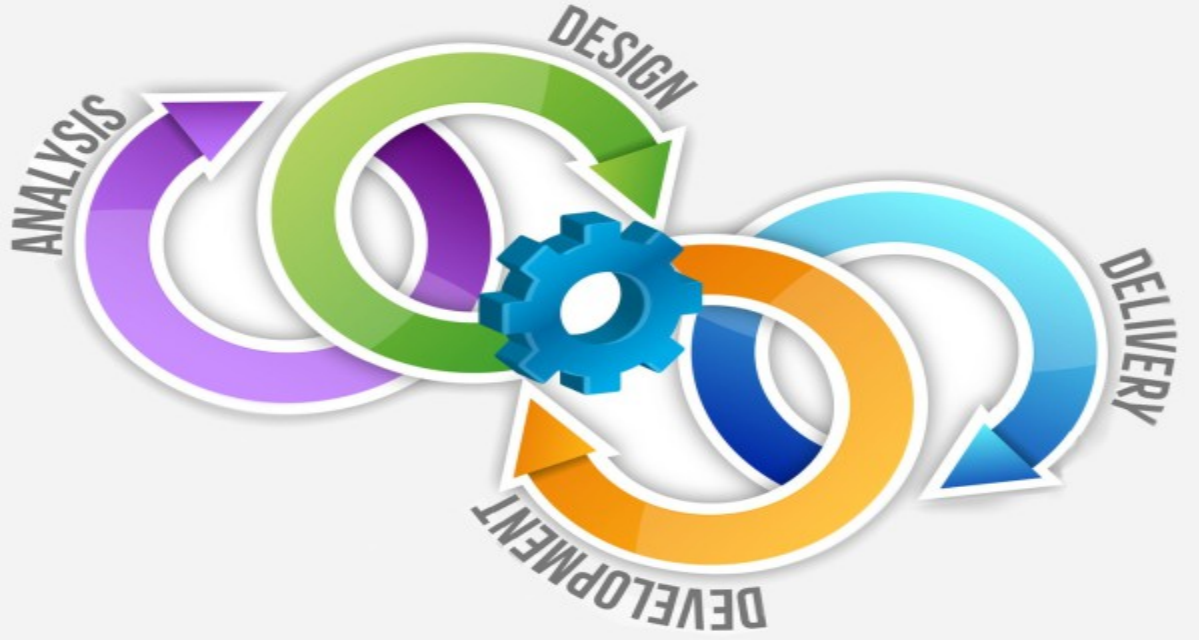
Alguns Exemplos de processos de desenvolvimento propostos são:

- ICONIX;
- RUP (*Rational Unified Process);*
- *EUP (Enterprise Unified Process);*
- *XP (Extreme Programming);*
- *OPEN (Objectoriented Process, Environment and Notation).*

ETAPAS DE DESENVOLVIMENTO

ETAPAS DE DESENVOLVIMENTO

- Um processo de desenvolvimento classifica em atividades as tarefas realizadas durante a construção de um sistema de software.



ETAPAS DE DESENVOLVIMENTO

- Há vários processos de desenvolvimento propostos.

- Sendo um consenso na comunidade de desenvolvimento de software o fato de que não existe o melhor processo de desenvolvimento.

ETAPAS DE DESENVOLVIMENTO

- Entretanto, podem-se distinguir atividades que, com uma ou outra modificação, são comuns à maioria dos processos existentes:
  - Levantamento de Requisitos;
  - Análise de Software;
  - Projeto de Software;
  - Implementação;
  - Testes;
  - Implantanção.

ETAPAS DE DESENVOLVIMENTO

- A atividade de Levantamento de Requisitos (também conhecida como elicitação de requisitos) corresponde à etapa de compreensão do problema aplicada ao desenvolvimento de software.

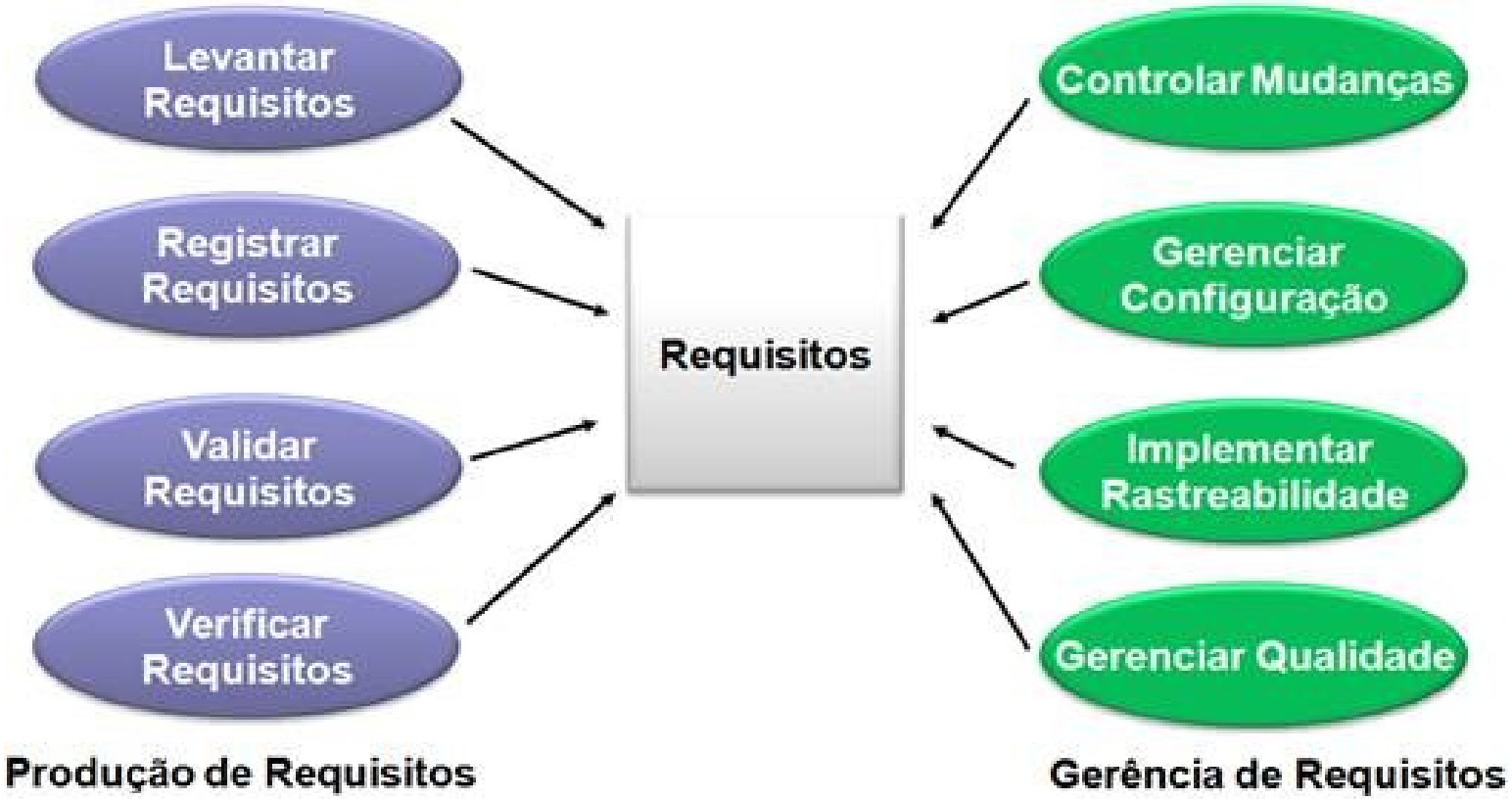
ETAPAS DE DESENVOLVIMENTO

- O principal objetivo do Levantamento de Requisitos é que usuários e desenvolvedores formalizem e tenham a mesma visão do problema a ser resolvido.

- É válido ressaltar que há três tipos de requisitos:
  - Funcionais;
  - Não Funcionais;
  - De Domínio.

ETAPAS DE DESENVOLVIMENTO

- As etapas de Levantamento de Requisitos e de Análise recebem o nome de *Engenharia de Requisitos*.



ETAPAS DE DESENVOLVIMENTO

- Formalmente, o termo *Análise corresponde a "quebrar"* um sistema em seus componentes e estudar como tais componentes interagem com o objetivo de entender como esse sistema funciona

- Nesta etapa realiza-se um estudo detalhado dos requisitos levantados na atividade anterior e a partir deste, são construídos modelos para representar o sistema a ser construído.

ETAPAS DE DESENVOLVIMENTO

- Muitos consideram que a etapa de Análise pode ser decomposta em:
  - *Análise do Domínio (ou Análise do Negócio)* - objetiva identificar e modelar os objetos do mundo real que, de alguma forma, serão processados pela aplicação em desenvolvimento;
  - *Análise da Aplicação:* objetivo identificar objetos de análise que normalmente não fazem sentido para os especialistas do domínio, mas que são necessários para suprir as funcionalidades do sistema em questão.

ETAPAS DE DESENVOLVIMENTO

- É válido ressaltar que na fase de Análise, não se deve considerar o ambiente tecnológico a ser utilizado.

- Nesta atividade, o foco deve ser tentar construir uma estratégia de solução sem se preocupar com a maneira *como* essa estratégia será realizada.

ETAPAS DE DESENVOLVIMENTO

- Em outras palavras, é necessário *saber* e *validar* o que o sistema proposto deve fazer para, então, definir *como* esse sistema irá fazê-lo.

ETAPAS DE DESENVOLVIMENTO

- Na fase de *Projeto*, determina-se "como" o sistema funcionará para atender aos requisitos, de acordo com os recursos tecnológicos existentes

- Também são considerados os aspectos físicos e dependentes de implementação

ETAPAS DE DESENVOLVIMENTO

- Nesta etapa, o analista deve produzir uma descrição computacional do que o software deve fazer.

- O mesmo, deve também verificar se o que foi validado na fase de análise está sendo de fato seguido.

ETAPAS DE DESENVOLVIMENTO

- Como já mencionado em aulas anteriores, o projeto de software pode ser decomposto em duas sub-etapas:

  - *Projeto de Arquitetura*: consiste em distribuir as classes e componentes inerentes ao sistema em subsistemas ou recursos de hardware.

  - *Projeto Detalhado*: consiste na modelagem de colaborações entre os objetos de cada módulo com o objetivo de realizar as funcionalidades do módulo.

ETAPAS DE DESENVOLVIMENTO

- Na fase de *Implementação*, o sistema é codificado;

- Ou seja, ocorre a tradução da descrição computacional obtida na fase de projeto em código executável mediante o uso de uma ou mais linguagens de programação.

ETAPAS DE DESENVOLVIMENTO

- Também na fase de *Implementação*, a equipe pode optar também pela reutilização de componentes de software, bibliotecas de classes e frameworks para agilizar a atividade.

ETAPAS DE DESENVOLVIMENTO

- Na etapa de *Testes*, diversas atividades são realizadas para verificação do sistema construído, levando-se em conta a especificação feita na fase de projeto.

- O principal produto dessa fase é o relatório de testes, com informações sobre erros detectados no software.

ETAPAS DE DESENVOLVIMENTO

- Após a atividade de testes, os diversos módulos do sistema são integrados, resultando finalmente no produto de software.

ETAPAS DE DESENVOLVIMENTO

- Na etapa de *Implantação*, o sistema é empacotado, distribuído e instalado no ambiente do usuário.

- Também devem ser escritos manuais do sistema e os usuários treinados para utilizar o sistema corretamente.

COMPONENTE HUMANO

COMPONENTE HUMANO

- O desenvolvimento de software é uma tarefa altamente cooperativa.

- Tecnologias complexas demandam especialistas em áreas específicas.

COMPONENTE HUMANO

- Comumente uma equipe de desenvolvimento de software é composta por:

  - Gerente;
  - Analistas;
  - Projetistas;
  - Programadores;
  - Clientes e grupos de avaliação de qualidade.

COMPONENTE HUMANO

- O Gerente de Projetos é o profissional responsável pela coordenação das atividades e orçamentos inerentes à construção do sistema.

- Também deve estimar o tempo necessário para o desenvolvimento do sistema, definir qual o processo de desenvolvimento e o cronograma de execução das atividades.

COMPONENTE HUMANO

- O Analista de sistemas é o profissional que deve ter conhecimento do domínio do negócio no intuito de conceber os requisitos do sistema.

- Assim sendo o responsável por entender as necessidades dos clientes em relação ao sistema a ser desenvolvido e repassar esse entendimento aos demais desenvolvedores do sistema.

COMPONENTE HUMANO

- O Projetista de sistemas é o integrante da equipe de desenvolvimento cujas funções são:

  - Avaliar as alternativas de solução (da definição) do problema resultante da análise;

  - Gerar a especificação de uma solução computacional detalhada.

COMPONENTE HUMANO

- Há diversos tipos de Projetistas:
  - de Interface (especializados nos padrões de uma interface gráfica, como o *Windows* ou o *MacOS),*
  - *de Redes (especializados no projeto de redes de comunicação);*
  - de Bancos de Dados (especializados no projeto de bancos de dados);

- E apesar dos diferentes fins, a atividade comum a qualquer projetista é adicionar os aspectos tecnológicos a aos modelos concebidos na análise.

COMPONENTE HUMANO

- O Programador é o responsável pela implementação do sistema.

- Este deve ser proficiente em uma ou mais linguagens de programação, além de ter conhecimento sobre Bancos de Dados e poder ler os modelos resultantes do trabalho do Projetista.

COMPONENTE HUMANO

- Os Avaliadores de Qualidade asseguram a adequação do processo de desenvolvimento e do produto de software sendo desenvolvido aos padrões de qualidade estabelecidos pela organização.

MODELOS DE CICLO DE VIDA

MODELOS DE CICLO DE VIDA

- Como já mencionado, processo de desenvolvimento de um *software* envolve diversas etapas;

- E ao encadeamento específico dessas etapas para a construção do sistema dá-se o nome de Modelo de Ciclo de Vida.

- O que diferencia os diferentes Modelos Ciclos de Vida é a forma como as etapas são encadeadas

MODELOS DE CICLO DE VIDA

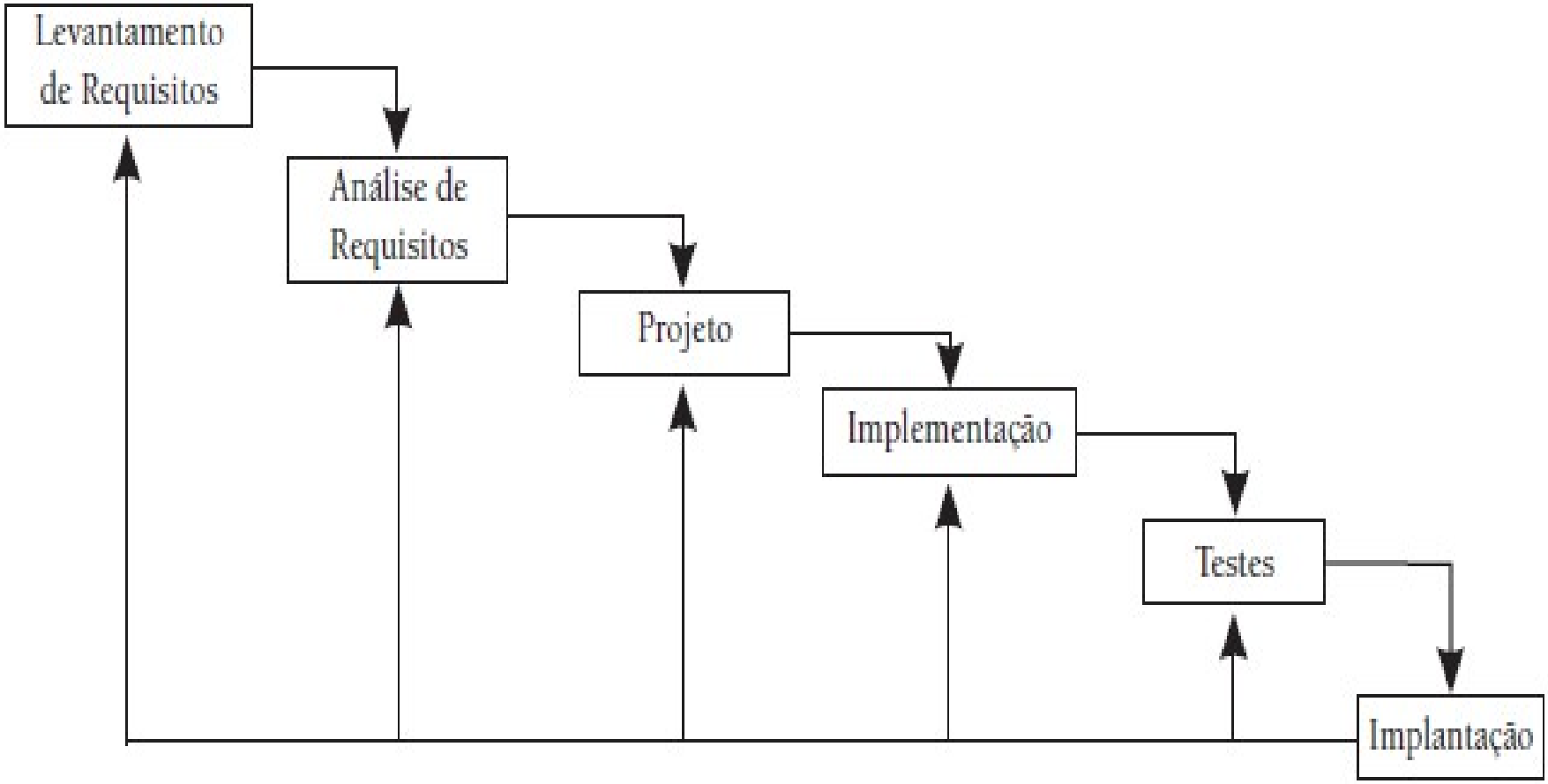
- Dois Modelos de Ciclo de Vida se destacam como os principais:
  - Modelo em Cascata;
  - Modelo Iterativo e Incremental.

MODELOS DE CICLO DE VIDA

- O modelo em Cascata, também chamado de *clássico ou linear, se caracteriza por possuir* uma tendência na progressão sequencial entre uma fase e a seguinte.

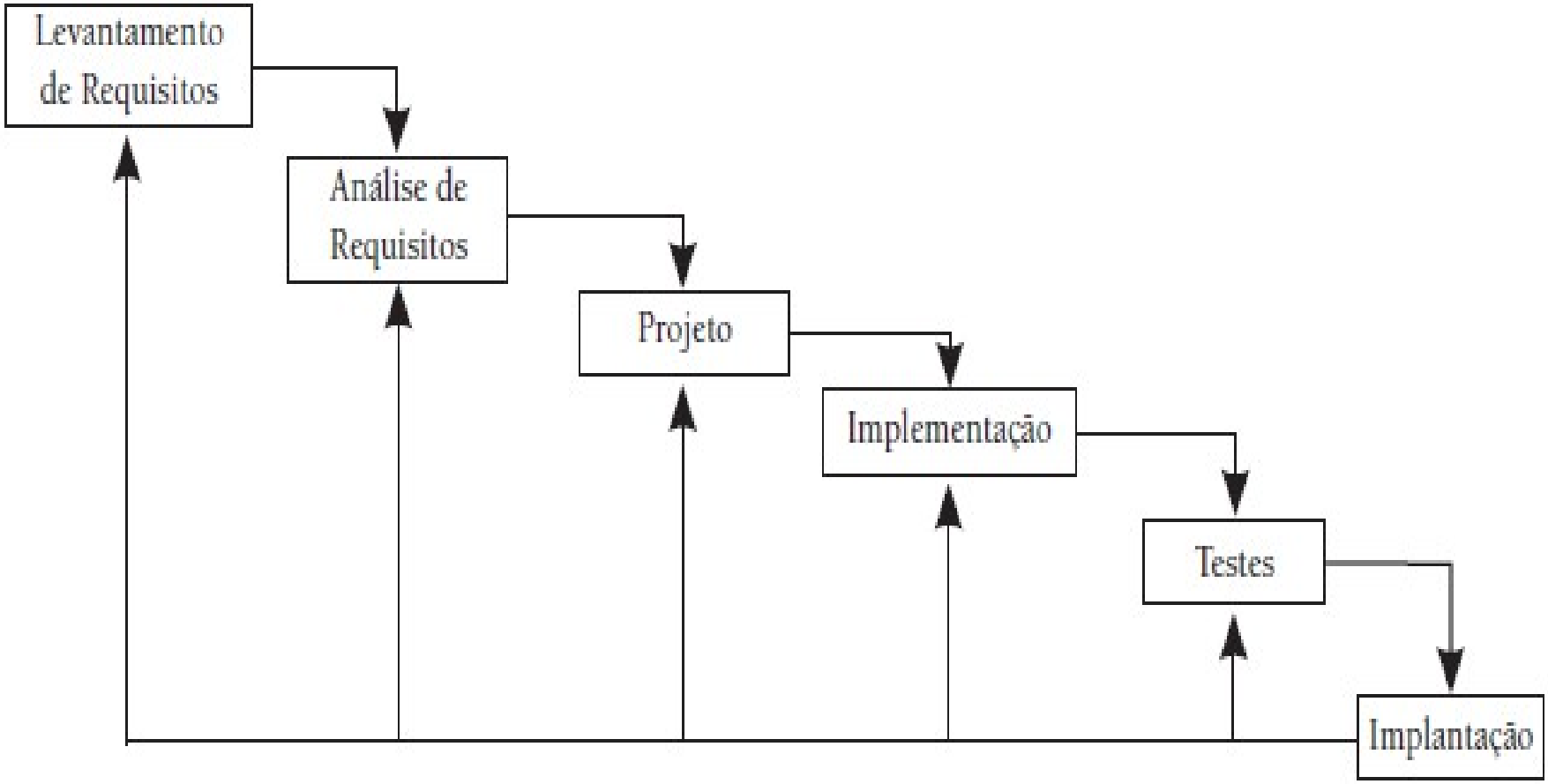
- Eventualmente, pode haver uma retroalimentação de uma fase para a fase anterior, mas, de um ponto de vista macro, as fases seguem sequencialmente.

MODELOS DE CICLO DE VIDA



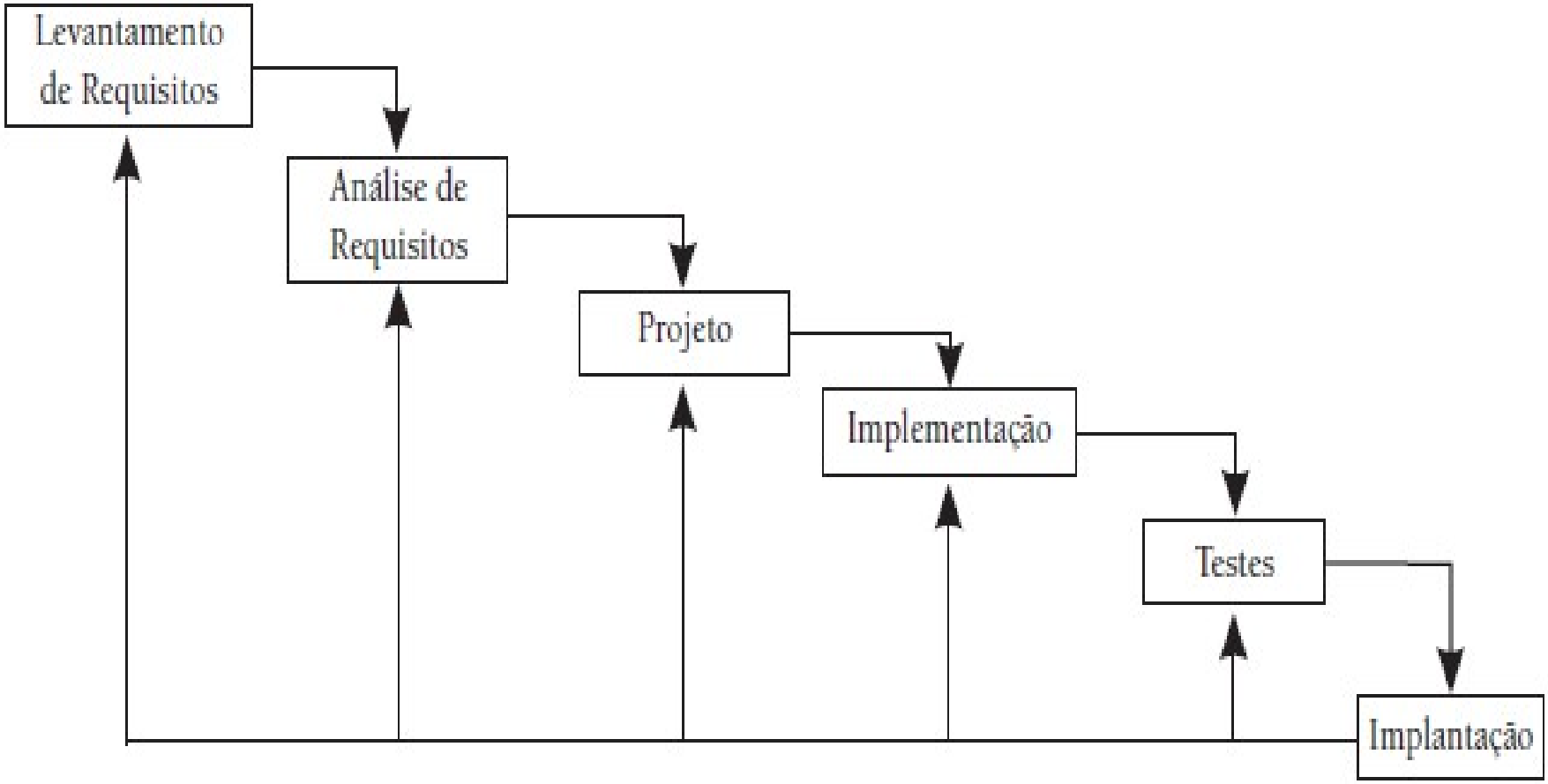
- Especificações: Projetos de desenvolvimento reais raramente seguem o fluxo sequencial que esse modelo propõe.

MODELOS DE CICLO DE VIDA



- Especificações: Esta abordagem presume que é possível declarar detalhadamente todos os requisitos antes do início das demais fases do desenvolvimento.

MODELOS DE CICLO DE VIDA



- Especificações: Uma versão de produção do sistema não estará pronta até que o ciclo do projeto de desenvolvimento chegue ao final.

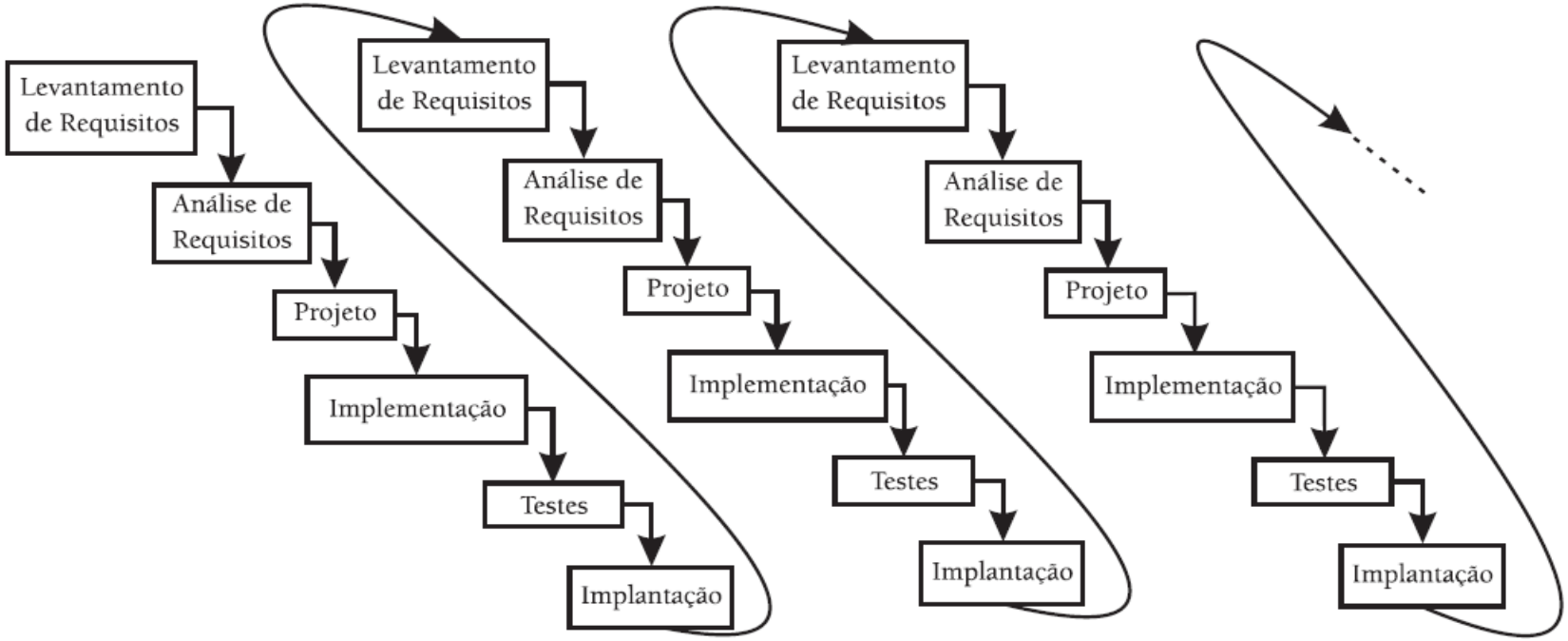
MODELOS DE CICLO DE VIDA

- Proposto com uma resposta aos problemas do modelo em Cascata, o modelo Iterativo e Incremental tende a dividir o desenvolvimento de um produto de software em ciclos.

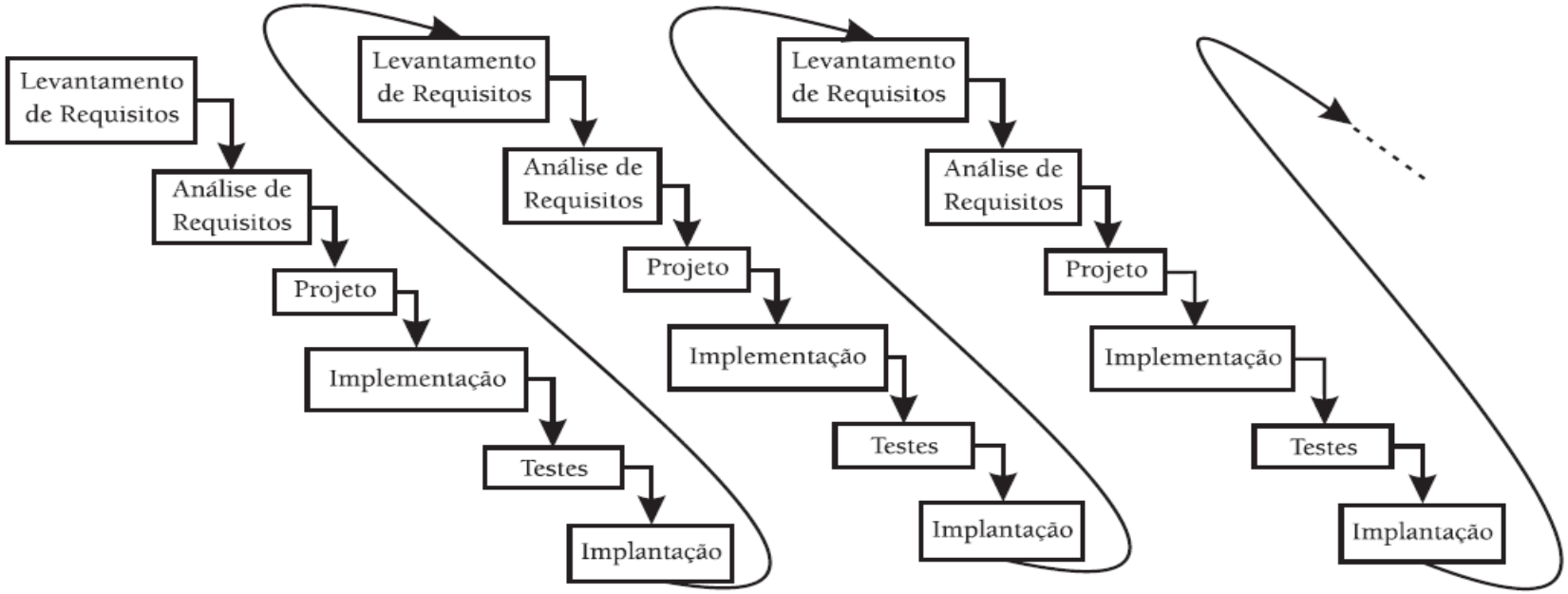
- Diferente de outras abordagens do gênero, para cada ciclo é alocado um subconjunto de requisitos.

MODELOS DE CICLO DE VIDA

- Logo, cada ciclo (a qual pode ser entendido como um conjunto de requisitos a ser tratado) tratado produz um novo incremento do sistema que contém extensões e refinamentos sobre o incremento (clico) anterior.



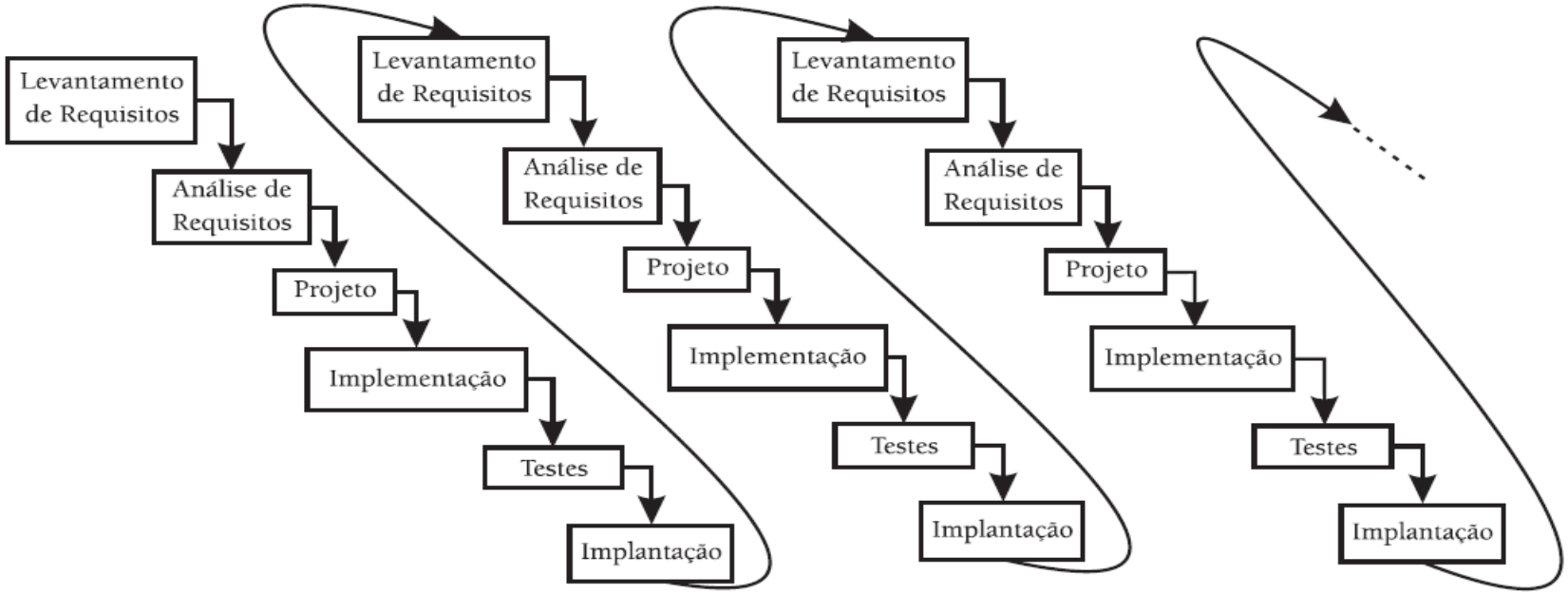
MODELOS DE CICLO DE VIDA



Observações:

1. Em cada ciclo de desenvolvimento, podem ser identificadas as fases de análise, projeto, implementação e testes.

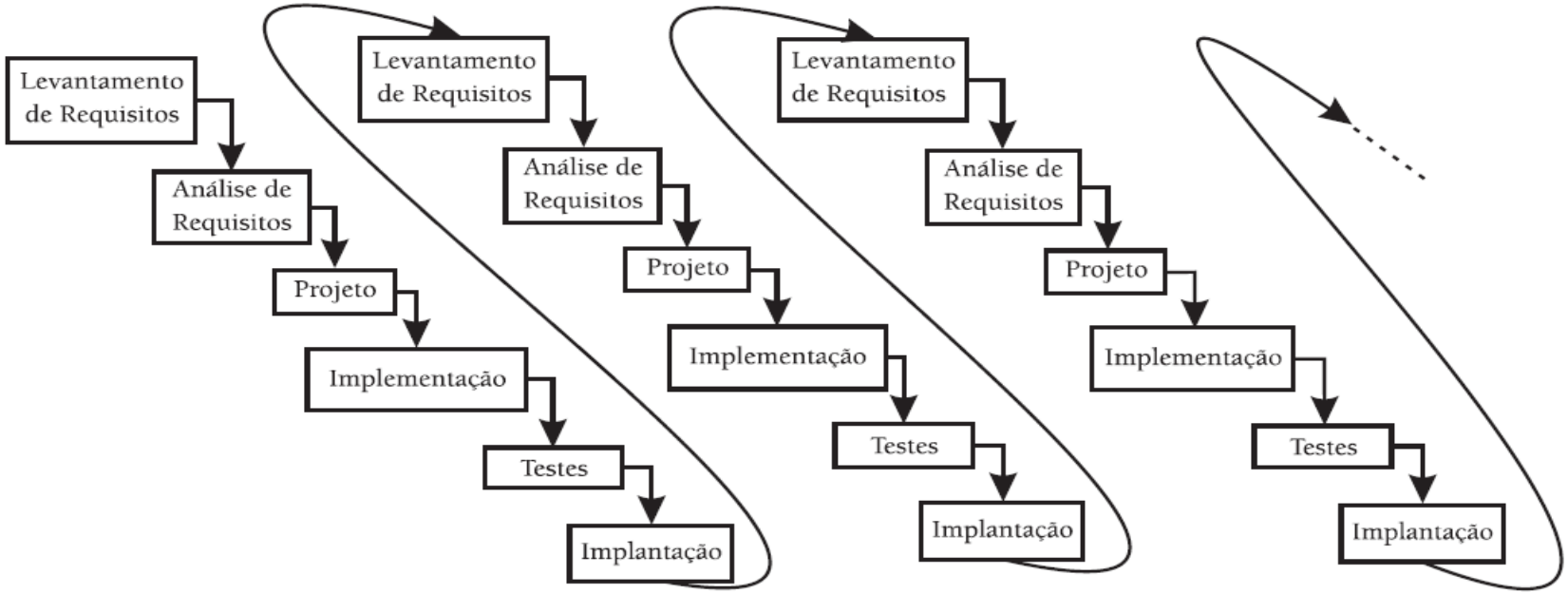
MODELOS DE CICLO DE VIDA



Observações:

2. Contrasta com a abordagem clássica, na qual as fases de análise, projeto, implementação e testes são realizadas uma única vez.

MODELOS DE CICLO DE VIDA



Observações:

3. Cada iteração é desenvolvida em cascata;

4. Este modelo só pode de ser adotado caso haja um mecanismo de divisão de requisitos com base prioridades e riscos.

MODELOS DE CICLO DE VIDA

- Os riscos de projeto não podem ser eliminados por completo.

- Portanto, na abordagem Incremental e Iterativa, os requisitos mais arriscados são considerados primeiramente.

MODELOS DE CICLO DE VIDA

“Se você não atacar os riscos [do projeto] ativamente, então estes irão ativamente atacar você” (Tom Gilb, 1988).

MODELOS DE CICLO DE VIDA

- Classificação de Riscos em um Projeto de Software segundo o *Project Management Institue (PMI)*:
  - Projeto de Software: Define os parâmetros operacionais, organizacionais e contratuais do desenvolvimento de software.
  - Riscos de Processo de Software. Relacionam-se os problemas técnicos e de gerenciamento.
  - Riscos de Produto de Software. Contém as características intermediárias e finais do produto.

ORGANIZAÇÃO GERAL DE UM PROCESSO INCREMENTAL E ITERATIVO

ORGANIZAÇÃO DO MODELO ITERATIVO

- O ciclo de vida de processo incremental e iterativo pode ser estudado segundo duas dimensões:
  - Dimensão temporal;
  - Dimensão de atividades.

ORGANIZAÇÃO DO MODELO ITERATIVO

- Na Dimensão Temporal, o processo está estruturado em *fases*.

- *Em cada uma* dessas *fases*, há uma ou mais *iterações*. *Cada iteração tem uma duração preestabelecida* (de duas a seis semanas).

- Ao final de cada iteração, é produzido um *incremento*, ou seja, uma parte do sistema final.

ORGANIZAÇÃO DO MODELO ITERATIVO

- Cada fase é concluída com um *marco;*

- Ou seja, quando decisões importantes são tomadas e objetivos críticos são alcançados.

ORGANIZAÇÃO DO MODELO ITERATIVO

- A Dimensão de Atividades compreende as atividades realizadas durante a iteração de uma *fase*:
  - Levantamento de Requisitos;
  - Análise de Requisitos;
  - Projeto;
  - Implementação;
  - Testes e Implantação.

PROTOTIPAGEM

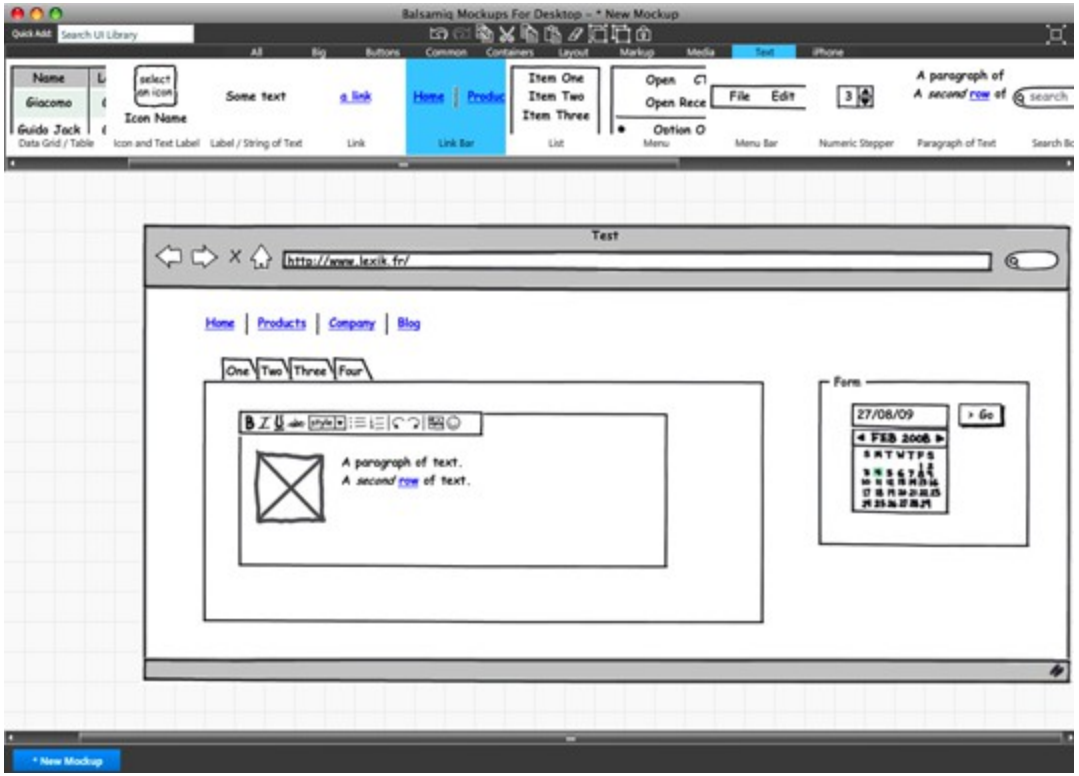
PROTOTIPAGEM

- Prototipagem refere-se a construção de protótipos e é uma atividade realizada para complementar à Análise de Requisitos.

- No âmbito de desenvolvimento de software, um protótipo é um esboço de alguma parte do sistema e geralmente serve para elucidar requisitos.

PROTOTIPAGEM

- Há diversos *softwares(wireframes)* e IDE's úteis para a realização de prototipagem. Exemplos:
  - *NetBEANS* e *Visual Studio*;
  - Axure;
  - *Pencil*;
  - *Fluid*.



FERRAMENTAS CASE

FERRAMENTAS CASE

- Existem diversos tipos de aplicações voltadas ao suporte das atividades inerentes ao ciclo de vida de desenvolvimento de software.

- Dois tipos de aplicações que têm esse objetivo são as ferramentas *CASE* e os IDE's (*Integrated Development Environment*).

FERRAMENTAS CASE

- É comum a quase todas as ferramentas CASE a possibilidade de produzir a perspectiva gráfica dos modelos de software.

- E normalmente, essas ferramentas usam os formatos UXF(*UML eXchange Format)* e XMI(*XML Metadata Interchange*) para a geração de modelos segundo a UML, garantindo assim a portabilidade.

FERRAMENTAS CASE

- É válido ressaltar que a partir de ferramentas CASE, pode-se:
  - Usufruir do conceito de Engenharia *Round-Trip* - capacidade de gerar código fonte a partir de diagramas e vice-versa;
  - Rastrear Requisitos - significa localizar os artefatos de software gerados como consequência da existência daquele requisito;
  - Verificar a consistência/validade dos modelos gerados.

FERRAMENTAS CASE

Quanto as IDE's, seu uso traz os seguintes benefícios:

- Depuração - capacidade de encontrar erros de lógica em partes de um programa;
- Verificação de Erros - a facilidade de compilação do programa em paralelo à escrita do mesmo;
- Refatoração - corresponde a alteração do código-fonte ou um componente de uma aplicação de tal forma que não altere o comportamento da mesma.

FERRAMENTAS CASE

- Além de IDE's e Ferramentas CASE, durante a progressão do processo de desenvolvimento de software as seguintes ferramentas são comumentes utilizadas:
  - Ferramentas de relatório de cobertura de testes e de testes automatizadas;
  - Ferramenta de Controle de Versão;
  - Ferramentas de Gerenciamento de Tarefas e Projetos;
  - Ferramentas de Averiguação e Monitoração de Desempenho.

FIM.