

ENGENHARIA DE SOFTWARE

Professor Me. Vladimir Geraseev Junior

**Curso: TECNOLOGIA EM DEFESA
CIBERNÉTICA**

Disciplina: Engenharia de Software

Sejam Bem Vindos!

Prof. Vladimir Geraseev Junior

PROF VLADIMIR GERASEEV JUNIOR

- **GRADUAÇÃO EM CIÊNCIAS DA COMPUTAÇÃO**
- **MESTRADO EM ENGENHARIA - UNITAU**
- **EXPERIÊNCIA PROFISSIONAL:**
 - COMPANHIA DE INFORMÁTICA DE JUNDIAÍ – ARQUITETO DE REDES
 - EMBRAER – GRUPO DE ENGENHARIA
 - INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS – INPE – ANALISTA DE REDES E SEGURANÇA
 - IMPLEMENTAÇÃO DOS REQUISITOS DE SEGURANÇA CIBERNÉTICA PARA EQUIPAMENTOS PARA TELECOMUNICAÇÕES (ATO 77) – ANATEL
 - CONSULTORIA EM CONDUÇÃO DE PENTESTS

PROF VLADIMIR GERASEEV JUNIOR

- EXPERIÊNCIA PROFISSIONAL ACADÊMICA.
- PROFESSOR NA FATEC TAUBATÉ
- PROFESSOR NA ETEC JACAREÍ
- PROFESSOR EM UNIVERSIDADES PARTICULARES (UNIP, UNIVAP, FAPI)

EMENTA DA DISCIPLINA

- **Introdução à Análise de Sistemas.**
- **Modelos de Ciclo de Vida de Software.**
- **Modelos de Processos de Desenvolvimento de Software (Modelo em Cascata, Espiral e Prototipagem)**
- **Definição e classificação de Requisitos de Software (funcionais e não funcionais).**

EMENTA DA DISCIPLINA

- **Técnicas de Levantamento de Requisitos.**
- **Modelo de Negócios aplicado ao levantamento de Requisitos (Canvas)**
- **Estudo de Viabilidade.**
- **Técnicas de documentação.**
- **Metodologias para desenvolvimento de sistemas.**

OBJETIVOS DA DISCIPLINA

- Identificar as características de Sistemas de Informação, seus tipos, viabilidade técnica, características de custo, valor e qualidade da informação;
- Explicar as características de um sistema, seus componentes e relacionamentos;
- Compreender o ciclo de vida utilizando concepções do modelo cascata;
- Utilizar conceitos da UML na análise de requisitos e na elaboração de diagramas focando na modelagem de sistemas.

BIBLIOGRAFIA

BIBLIOGRAFIA BÁSICA:

- BEZERRA, EDUARDO. **PRINCÍPIOS DE ANÁLISE E PROJETO DE SISTEMAS COM UML**. 3 ED. RIO DE JANEIRO: ELSEVIER, 2015.
- PRESSMAN, ROGER; MAXIM, BRUCE. **ENGENHARIA DE SOFTWARE**. 8 ED. SÃO PAULO: MCGRAW HILL BRASIL, 2016.
- SOMMERVILLE, IAN. **ENGENHARIA DE SOFTWARE**. 10 ED. SÃO PAULO: PEARSON BRASIL, 2019.

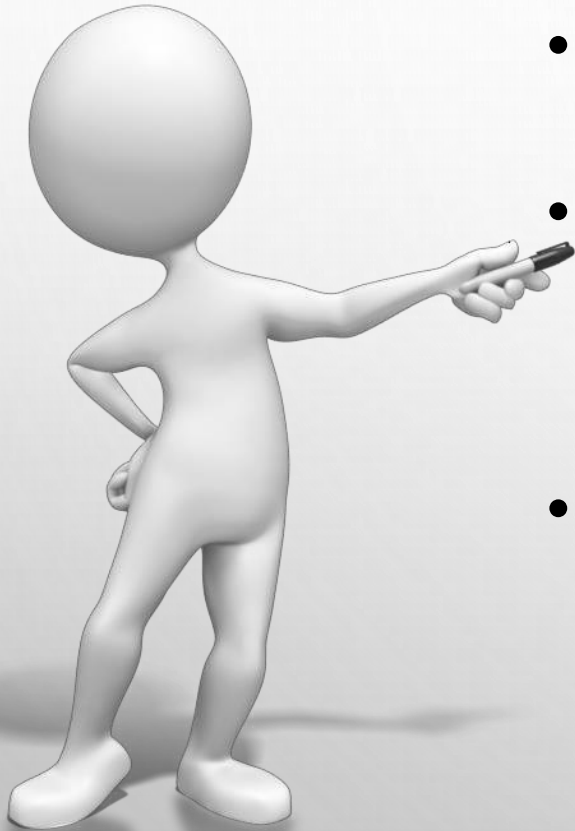
BIBLIOGRAFIA COMPLEMENTAR:

- LARMAN, CRAIG. **UTILIZANDO UML E PADRÕES**. 3 ED. PORTO ALEGRE: BOOKMAN, 2007.
- REZENDE, DENIS ALCIDES. **ENGENHARIA DE SOFTWARE E SISTEMAS DE INFORMAÇÃO**. 3 ED. RIO DE JANEIRO: BRASPORT, 2005.
- WASLAWICK RAUL. **ANÁLISE E PROJETO DE SISTEMAS DE INFORMAÇÃO ORIENTADOS A OBJETOS**. 2 ED. RIO DE JANEIRO: ELSEVIER, 2010

INTRODUÇÃO A ANÁLISE DE SISTEMAS

ANÁLISE DE SISTEMAS = ENGENHARIA DE SOFTWARE

HISTÓRICO DO SOFTWARE



- Anos 40 – Evolução dos Sistemas computadorizados
 - Desenvolvimento de Hardwares
- Anos 50 – Desenvolvimento de Sist. Operacionais
 - Poupar o usuário de conhecer as questões de funcionamento interno da Máquina
 - Surgimento das linguagem de “alto nível”: FORTRAN e COBOL
- Anos 60 – Desenvolvimento De Grandes Sistemas
 - Queda de Preços do Hardware
 - Substituição de “pequenos programas”
 - Necessidade de Grandes Sistemas

HISTÓRICO DO SOFTWARE



- Entre 60 e 70
 - A eficiência e utilidade dos sistemas computacionais tiveram um considerável crescimento.
- PORÉM
 - Havia falta de experiência e não adequação dos métodos de desenvolvimento existentes para pequenos programas.
- COMO CONSEQUÊNCIA surge a então chamada “CRISE DE SOFTWARE”

HISTÓRICO DO SOFTWARE



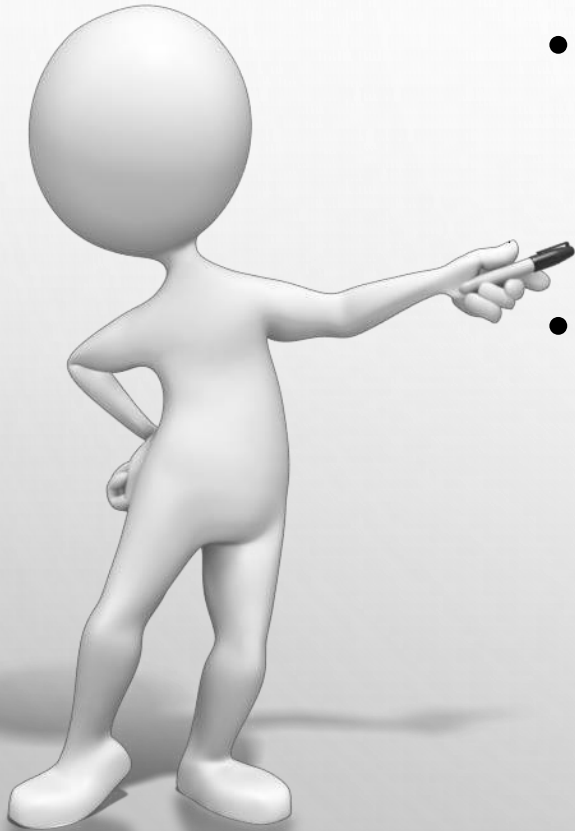
- Os Projetos apresentavam:
 - Anos de atraso na entrega;
 - Custo superava as previsões;
 - Não era confiável;
 - Difícil de manter;
 - Desempenho era insatisfatório.
- Em resposta à “CRISE DE SOFTWARE”
 - Nasceu o termo “Engenharia de Software (1968)”

HISTÓRICO DO SOFTWARE



- **Década de 80**
 - Computador desktop - A década foi marcada pelo surgimento da Microsoft e pela evolução dos computadores pessoais - desktop.
 - Unix -O Unix (sistemas operacionais em rede) avançou pelo mundo.
 - Evolução da internet - A internet começa a despontar como meio comunicação.
- **Década de 90**
 - Internet - A internet é amplamente utilizada.
 - Linux -Surge o núcleo do futuro Linux.
 - JAVA - Nasce a linguagem JAVA, que revoluciona o paradigma da orientação a objetos.

HISTÓRICO DO SOFTWARE



- Década de 2000
 - Internet - Softwares que utilizam a web como plataforma de funcionamento.
 - Computação on-premise– Empresas com datacenter próprios
- Década de 2010-2020
 - Computação em nuvem - Utilizada em larga escala por diversos seguimentos de empresas.
 - Aplicativos para dispositivos móveis - Com a evolução dos dispositivos móveis os softwares para aplicativos tornaram-se essenciais.
 - Inteligência Artificial - Utilização de algoritmos para a Deep Learning (aprendizagem profunda) e Machine Learning (aprendizado de máquina).

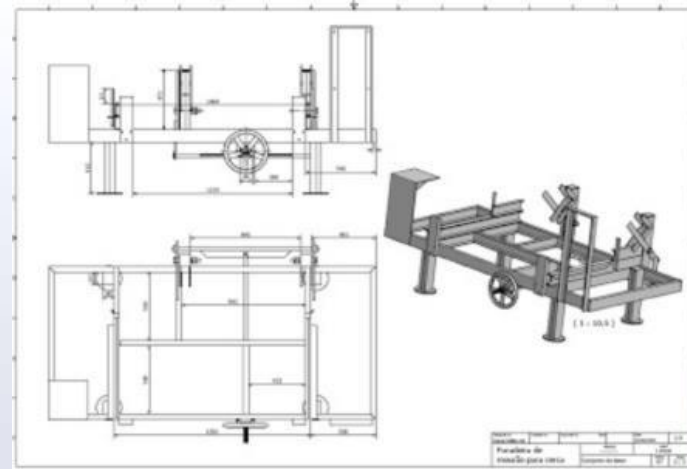
INTRODUÇÃO A ANÁLISE DE SISTEMAS



- Mas o que realmente significa Engenharia de Software ?
 - Estudo ou aplicação de abordagens sistemáticas para o desenvolvimento, operação e manutenção de software de qualidade.

INTRODUÇÃO A ANÁLISE DE SISTEMAS

- Engenheiros mecânicos desenham antes de produzirem suas máquinas



INTRODUÇÃO A ANÁLISE DE SISTEMAS

- Engenheiros civis fazem plantas antes de construírem suas edificações



INTRODUÇÃO A ANÁLISE DE SISTEMAS

- Engenheiros civis fazem plantas antes de construírem suas edificações



INTRODUÇÃO A ANÁLISE DE SISTEMAS

- O SER HUMANO CONSTRÓI PRÉDIOS, PONTES E CASAS HÁ VÁRIOS MILHARES DE ANOS.
- TAMBÉM PRODUZ MÁQUINAS E FERRAMENTAS HÁ MILHARES DE ANOS
- PORÉM O DESENVOLVIMENTO DE SOFTWARE TEM APENAS ALGUNS ANOS, APROXIMADAMENTE 60 ANOS.

INTRODUÇÃO A ANÁLISE DE SISTEMAS

- DESENVOLVER SOFTWARES COM QUALIDADE;
- PROVER PRODUTIVIDADE NO DESENVOLVIMENTO, OPERAÇÃO E MANUTENÇÃO DO SOFTWARE;
- PERMITIR QUE PROFISSIONAIS TENHAM CONTROLE SOBRE O DESENVOLVIMENTO, TAIS COMO ESCOPO, PRAZOS E CUSTOS.

E O ENGENHEIRO DE SOFTWARE ?



- Pode-se dizer que o engenheiro de software preocupa-se com os aspectos da construção de um sistema de software, desde o início de sua especificação até a manutenção, após ter sido posto em operação.

QUAL O OBJETIVO DA ENGENHARIA DE SOFTWARE?

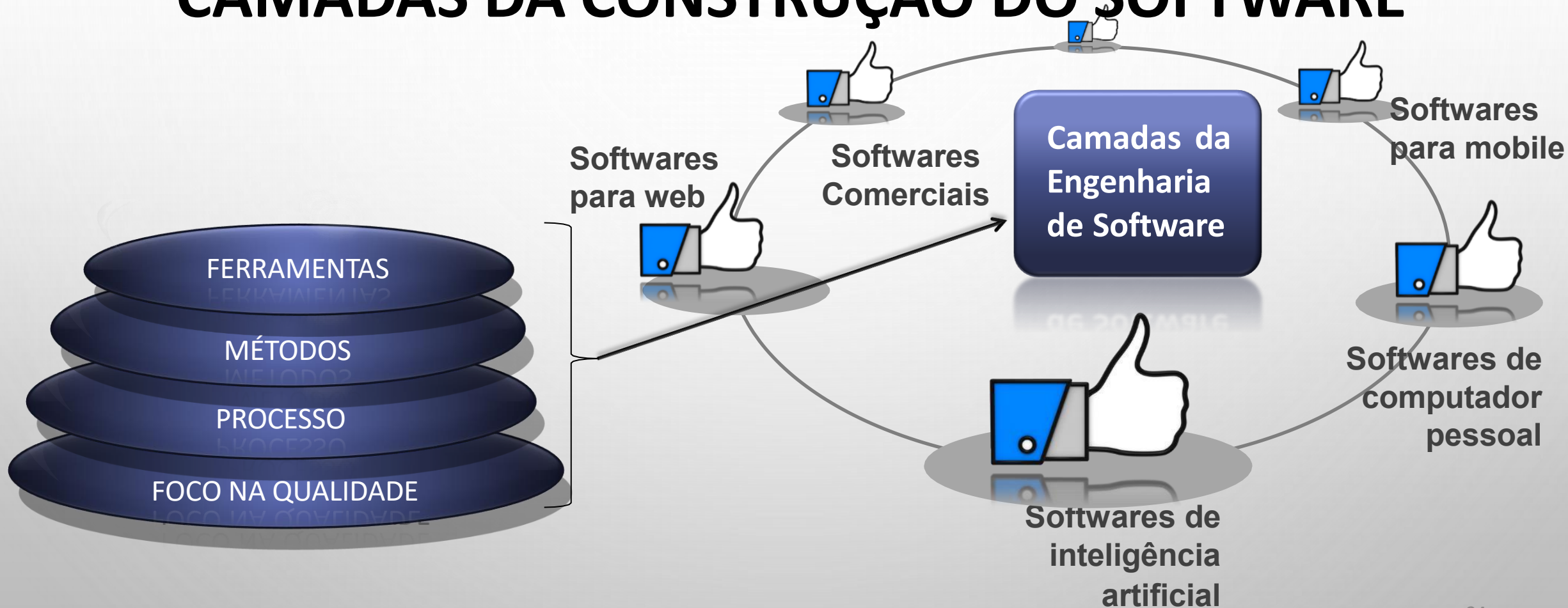
Aplicação de teoria, modelos, formalismos, técnicas e ferramentas da ciência da computação e áreas afins para a **produção** (ou desenvolvimento) sistemática de software

—Aplicação de métodos, técnicas e ferramentas para o **gerenciamento** do processo de desenvolvimento

QUAL O OBJETIVO DA ENGENHARIA DE SOFTWARE?

- Produção da **documentação** formal destinada a comunicação entre os membros da equipe de desenvolvimento bem como aos usuários;
- Apoiar o desenvolvimento profissional de software, mais do que a programação individual;

CAMADAS DA CONSTRUÇÃO DO SOFTWARE



O QUE É UM SOFTWARE?

- Software é o programa associado a todos os dados de documentação e configuração, necessários para que o programa opere corretamente.

O QUE É UM SOFTWARE?

- Sommerville (2018) define softwares como "programas de computador acompanhados de documentação associada", e esclarece que eles podem ser criados para clientes específicos ou para um público mais amplo.

O QUE É UM SOFTWARE?

Pressman (2021) descreve softwares de computador como produtos desenvolvidos e mantidos a longo prazo por profissionais de Tecnologia da Informação (TI). Ele também destaca que softwares englobam qualquer forma de mídia eletrônica e são constituídos por três elementos principais:

Instruções: quando executadas, fornecem os atributos e funções de desempenhos desejados pelos usuários.

Estruturas de dados: possibilitam aos programadores manipular as informações de forma mais adequada conforme a necessidade da aplicação.

Documentação: é toda a informação descritiva do software, a qual detalha a operação de uso dos programas, diagramas de funcionalidades, etc.

APLICAÇÕES DE SOFTWARE

- **Software Básico:** programas escritos para apoiar outros programas. Têm forte interação com o hardware:
 - *Sistema operacional, compiladores, drivers, etc.*
- **Software de Tempo Real:** software que monitora, analisa e controla eventos do mundo real:
 - *sistema de controle de tráfego aéreo, relógio digital, etc.*
- **Software Comercial:** sistemas de operações comerciais e de tomadas de decisões administrativas:
 - *folha de pagamentos, contas a pagar e a receber, sistemas de informação administrativa, etc.*

APLICAÇÕES DE SOFTWARE

- **Software Científico e de Engenharia:** caracterizado por algoritmos de processamento numérico:
 - Astronomia, vulcanologia, análise de fadiga da mecânica de automóveis, biologia molecular, CAD, etc.
- **Software Embutido (Embarcado):** usado para controlar produtos e sistemas para os mercados industriais e de consumo (produtos inteligentes):
 - Controle de micro-ondas, de combustível, sistemas de freio, etc.

APLICAÇÕES DE SOFTWARE

- **Software de Computador Pessoal:** visam atingir as necessidade do usuário de computadores pessoais:
 - processamento de textos, planilhas eletrônicas, computação gráfica, diversões, etc.
- **Software de Inteligência Artificial:** faz uso de algoritmos não numéricos para resolver problemas que não sejam favoráveis à computação ou à análise direta:
 - Sistemas especialistas, reconhecimento de padrões, redes neurais artificiais, jogos, etc.

APLICAÇÕES DE SOFTWARE

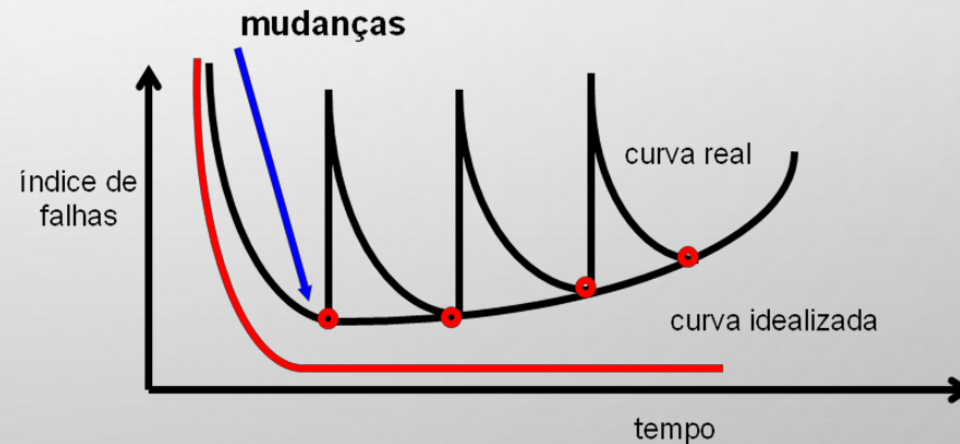
- **Software para a Internet:** envolvem a utilização de Browsers para acesso à WWW. Envolvem tecnologias utilizadas para o desenvolvimento de sites:
 - HTML, CGI, JAVA, ASP, XML, etc.

CARACTERÍSTICAS DE UM SOFTWARE

- Desenvolvido ou projetado por engenharia;
- Não se “desgasta”, mas se deteriora devido as mudanças de requisitos (mudanças exigidas pelas regras de negócio e tecnologia);
- Normalmente os produtos de software é concebida inteiramente sob medida;

CARACTERÍSTICAS DE UM SOFTWARE

- O que acontece quando um hardware se desgasta?
 - “Peça de Reposição”
- E no caso do software?
 - Toda falha indica um erro no projeto



ENGENHARIA DE SOFTWARE

- Pode ser organizada em camadas com foco em qualidade

- A engenharia de software inclui

- Processo
- Métodos
- Ferramentas

Ferramentas

Métodos de Software

Processos de Software

Qualidade

FERRAMENTAS

- Fornecem apoio automatizado (ou semi-automatizado) para o processo e para os métodos
- Exemplo: ferramentas de modelagem do processo
 - Permitem definir ações, tarefas, produtos, etc. de um processo

Ferramentas

Métodos de Software

Processo de Software

Qualidade

MÉTODOS DE SOFTWARE

- Técnicas para desenvolvimento de software

- Como fazer?

- Métodos incluem

- Modelos
 - Notações
 - Regras, etc.

Ferramentas

Métodos de Software

Processo de Software

Qualidade

PROCESSO DE SOFTWARE

- Atividades (e seus resultados) para o desenvolvimento de software

- O que fazer?

- Atividades principais

- Especificação
 - Modelagem
 - Implementação
 - Verificação
 - Evolução

Ferramentas

Métodos de Software

Processo de Software

Qualidade

QUALIDADE DE SOFTWARE

■ Atributos de um bom software

- Facilidade de manutenção
- Confiança
- Eficiência
- Usabilidade, etc.

Ferramentas

Métodos de Software

Processo de Software

Qualidade

TIPOS DE SISTEMAS

- **NATURAIS:** EXISTENTES NA NATUREZA (EX: ECOSSISTEMAS).
- • **ARTIFICIAIS:** CRIADOS PELO HOMEM (EX: SISTEMAS DE COMPUTADOR).
- • **ABERTOS:** INTERAGEM COM O AMBIENTE EXTERNO.
- • **FECHADOS:** NÃO INTERAGEM COM O AMBIENTE EXTERNO.
- • **SIMPLES:** POUCOS ELEMENTOS E INTERAÇÕES SIMPLES.
- • **COMPLEXOS:** MUITOS ELEMENTOS E INTERAÇÕES COMPLEXAS.
- • **DETERMINÍSTICOS:** COMPORTAMENTO PREVISÍVEL.
- • **ESTOCÁSTICOS:** COMPORTAMENTO IMPREVISÍVEL.
- • **ESTÁTICOS:** NÃO MUDAM AO LONGO DO TEMPO.
- • **DINÂMICOS:** MUDAM AO LONGO DO TEMPO.

SISTEMAS DE INFORMAÇÕES

- A NECESSIDADE É A MÃE DAS INVENÇÕES
 - EM CONSEQÜÊNCIA DO CRESCIMENTO DA IMPORTÂNCIA DA INFORMAÇÃO, SURTIU A NECESSIDADE DE GERENCIAR INFORMAÇÕES DE UMA FORMA ADEQUADA E EFICIENTE E, DESTA NECESSIDADE, SURTIRAM OS DENOMINADOS **SISTEMAS DE INFORMAÇÕES**.
- UM SI É UMA COMBINAÇÃO DE PESSOAS, DADOS, PROCESSOS, INTERFACES, REDES DE COMUNICAÇÃO E TECNOLOGIA QUE INTERAGEM COM O OBJETIVO DE DAR SUPORTE E MELHORAR O PROCESSO DE NEGÓCIO DE UMA ORGANIZAÇÃO COM RELAÇÃO ÀS INFORMAÇÕES.
 - VANTAGENS DO PONTO DE VISTA COMPETITIVO.
- OBJETIVO PRINCIPAL E FINAL DA CONSTRUÇÃO DE UM SI: *ADIÇÃO DE VALOR À ORGANIZAÇÃO*.

SISTEMAS DE SOFTWARE

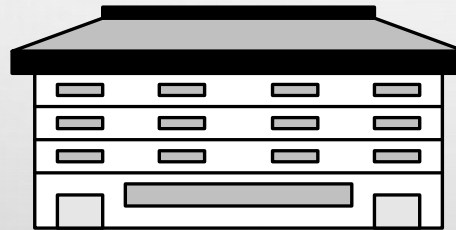
- UM DOS COMPONENTES DE UM SISTEMA DE INFORMAÇÃO É DENOMINADO **SISTEMA DE SOFTWARE**.
- COMPREENDE OS MÓDULOS FUNCIONAIS COMPUTADORIZADOS QUE INTERAGEM ENTRE SI PARA PROPORCIONAR A AUTOMATIZAÇÃO DE DIVERSAS TAREFAS.
- CARACTERÍSTICA INTRÍNSECA DO DESENVOLVIMENTO DE SISTEMAS DE SOFTWARE: **COMPLEXIDADE**.

SISTEMAS DE SOFTWARE

Uma analogia...



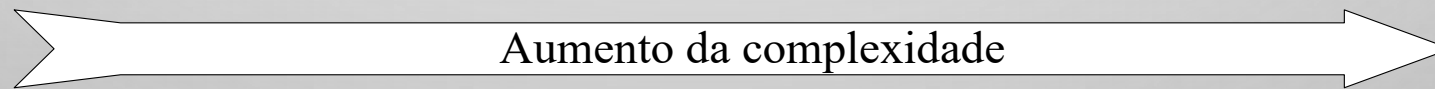
Casa de
canhorro

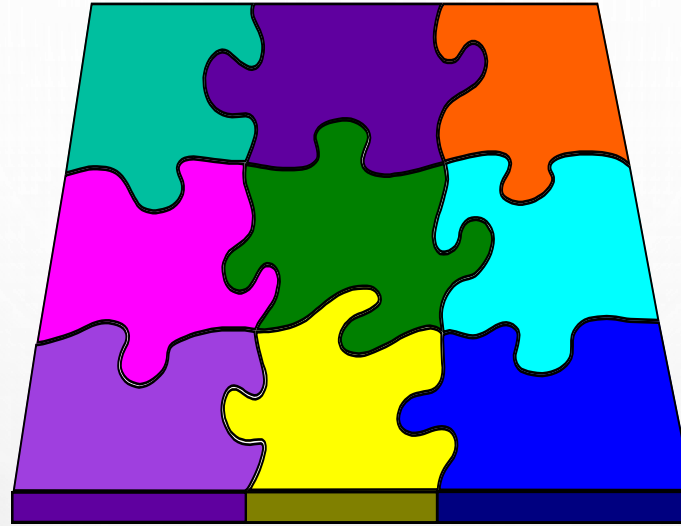


Casa



Arranha-Ceús





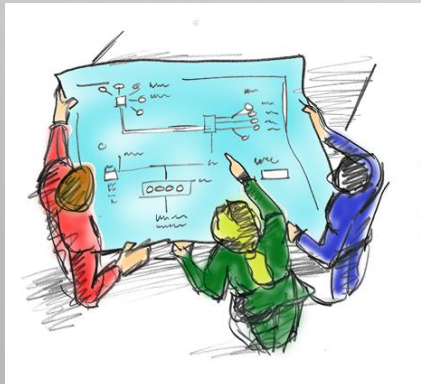
MODELAGEM DE SISTEMAS DE SOFTWARE

MODELAGEM DE SOFTWARE

A modelagem de sistemas de software consiste na utilização de notações gráficas e textuais com o objetivo de construir modelos que representam as partes essenciais de um sistema, considerando-se diversas perspectivas diferentes e complementares

MODELOS DE SOFTWARE

- NA CONSTRUÇÃO DE SISTEMAS DE SOFTWARE, ASSIM COMO NA CONSTRUÇÃO DE SISTEMAS HABITACIONAIS, TAMBÉM HÁ UMA GRADAÇÃO DE COMPLEXIDADE.
 - A CONSTRUÇÃO DESSES SISTEMAS NECESSITA DE UM PLANEJAMENTO INICIAL.
- UM MODELO PODE SER VISTO COMO UMA REPRESENTAÇÃO IDEALIZADA DE UM SISTEMA QUE SE PLANEJA CONSTRUIR.
- MAQUETES DE EDIFÍCIOSE DE AVIÕES E PLANTAS DE CIRCUITOS ELETRÔNICOS SÃO APENAS ALGUNS EXEMPLOS DE MODELOS.



RAZÕES PARA CONSTRUÇÃO DE MODELOS

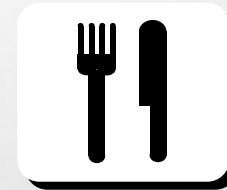
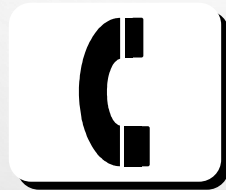
- A PRINCÍPIO, PODEMOS VER A CONSTRUÇÃO DE MODELOS COMO UMA ATIVIDADE QUE ATRASA O DESENVOLVIMENTO DO SOFTWARE PROPRIAMENTE DITO.
- MAS ESSA ATIVIDADE PROPICIA...
 - O **GERENCIAMENTO DA COMPLEXIDADE** INERENTE AO DESENVOLVIMENTO DE SOFTWARE.
 - A **COMUNICAÇÃO** ENTRE AS PESSOAS ENVOLVIDAS.
 - A **REDUÇÃO DOS CUSTOS** NO DESENVOLVIMENTO.
 - A **PREDIÇÃO DO COMPORTAMENTO** FUTURO DO SISTEMA.
- ENTRETANTO, NOTE O FATOR **COMPLEXIDADE** COMO CONDICIONANTE DESSAS VANTAGENS.

DIAGRAMAS E DOCUMENTAÇÃO

- NO CONTEXTO DE DESENVOLVIMENTO DE SOFTWARE, CORRESPONDEM A DESENHOS GRÁFICOS QUE SEGUEM ALGUM PADRÃO LÓGICO.
- PODEMOS TAMBÉM DIZER QUE UM DIAGRAMA É UMA APRESENTAÇÃO DE UMA COLEÇÃO DE ELEMENTOS GRÁFICOS QUE POSSUEM UM SIGNIFICADO PREDEFINIDO.
- DIAGRAMAS NORMALMENTE SÃO CONSTRUÍDOS DE ACORDO COM REGRAS DE NOTAÇÃO BEM DEFINIDAS.
 - OU SEJA, CADA FORMA GRÁFICA UTILIZADA EM UM DIAGRAMA DE MODELAGEM TEM UM SIGNIFICADO ESPECÍFICO.

DIAGRAMAS E DOCUMENTAÇÃO

- DIAGRAMAS PERMITEM A CONSTRUÇÃO DE UMA REPRESENTAÇÃO CONCISA DE UM SISTEMA A SER CONSTRUÍDO.
 - “UMA FIGURA VALE POR MIL PALAVRAS”



- NO ENTANTO, MODELOS TAMBÉM SÃO COMPOSTOS DE INFORMAÇÕES TEXTUAIS.
- DADO UM MODELO DE UMA DAS PERSPECTIVAS DE UM SISTEMA, DIZ-SE QUE O SEU DIAGRAMA, JUNTAMENTE COM A INFORMAÇÃO TEXTUAL ASSOCIADA, FORMAM A **DOCUMENTAÇÃO** DESTE MODELO.

PROCESSO DE DESENVOLVIMENTO DE SOFTWARE

“SOFTWARE IS HARD...”

- 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.

Fonte: Chaos Report (1994)

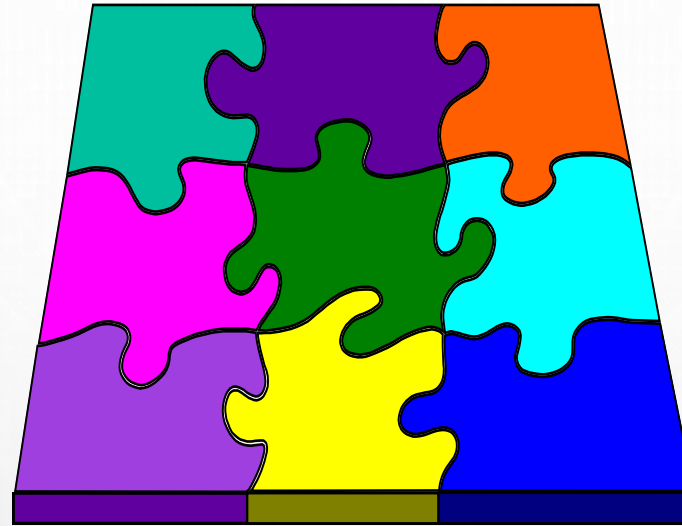


PROCESSO DE DESENVOLVIMENTO

- TENTATIVAS DE LIDAR COM A COMPLEXIDADE E DE MINIMIZAR OS PROBLEMAS ENVOLVIDOS NO DESENVOLVIMENTO DE SOFTWARE ENVOLVEM A DEFINIÇÃO DE ***PROCESSOS DE DESENVOLVIMENTO DE SOFTWARE***.
- UM PROCESSO DE DESENVOLVIMENTO DE SOFTWARE (PDS) COMPREENDE TODAS AS ATIVIDADES NECESSÁRIAS PARA DEFINIR, DESENVOLVER, TESTAR E MANTER UM PRODUTO DE SOFTWARE.

PROCESSO DE DESENVOLVIMENTO

- EXEMPLOS DE PROCESSOS DE DESENVOLVIMENTO EXISTENTES:
 - ICONIX
 - RUP
 - EUP
 - XP
 - OPEN
- ALGUNS OBJETIVOS DE UM PROCESSO DE DESENVOLVIMENTO SÃO:
 - DEFINIR *QUAIS* AS ATIVIDADES A SEREM EXECUTADAS AO LONGO DO PROJETO;
 - DEFINIR *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.



ATIVIDADES TÍPICAS DE UM PDS

ATIVIDADES TÍPICAS DE UM PDS

- LEVANTAMENTO DE REQUISITOS
- ANÁLISE DE REQUISITOS
- PROJETO
- IMPLEMENTAÇÃO
- TESTES
- IMPLANTAÇÃO

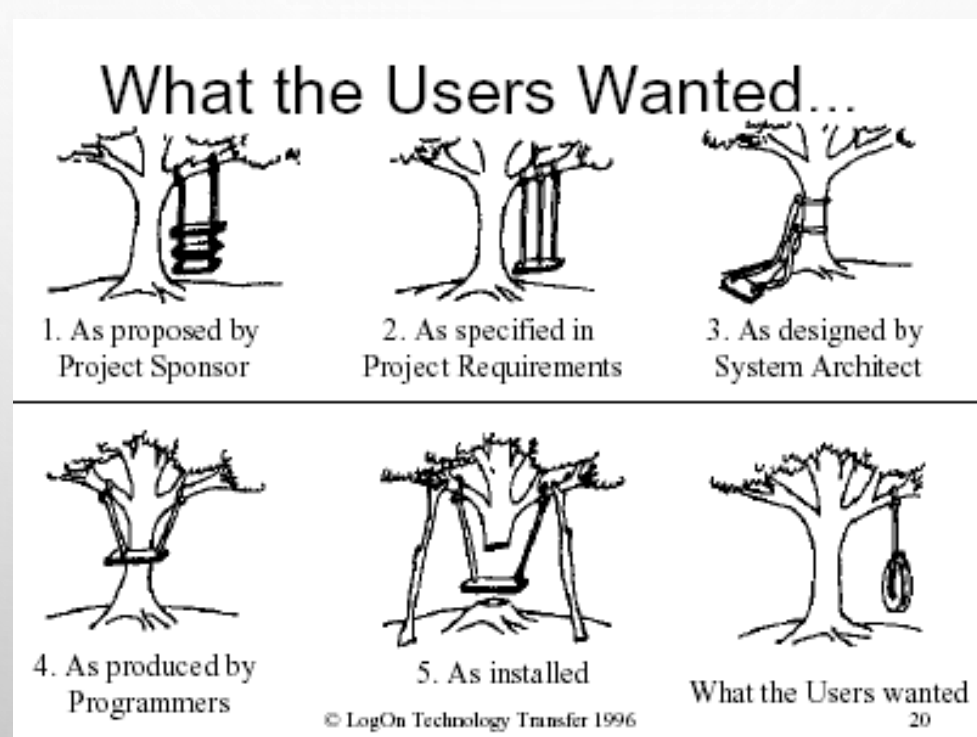
PARTICIPANTES DO PROCESSO

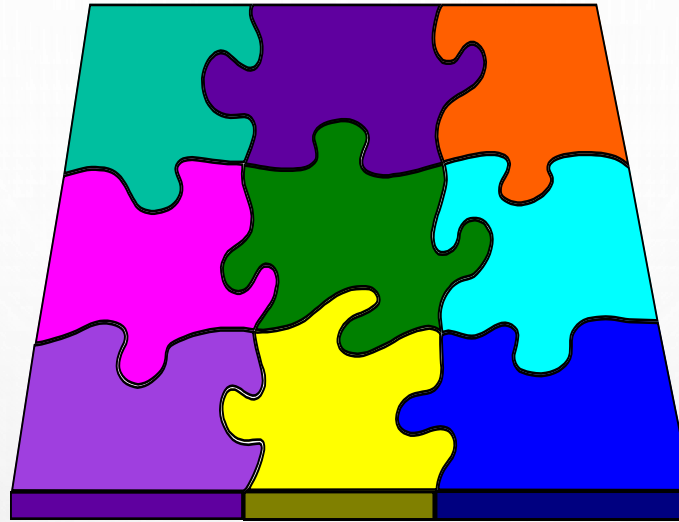
- GERENTES DE PROJETO
- ANALISTAS
- PROJETISTAS
- ARQUITETOS DE SOFTWARE
- PROGRAMADORES
- CLIENTES
- AVALIADORES DE QUALIDADE



PARTICIPAÇÃO DO USUÁRIO

- A PARTICIPAÇÃO DO USUÁRIO DURANTE O DESENVOLVIMENTO DE UM SISTEMA EXTREMAMENTE É IMPORTANTE.





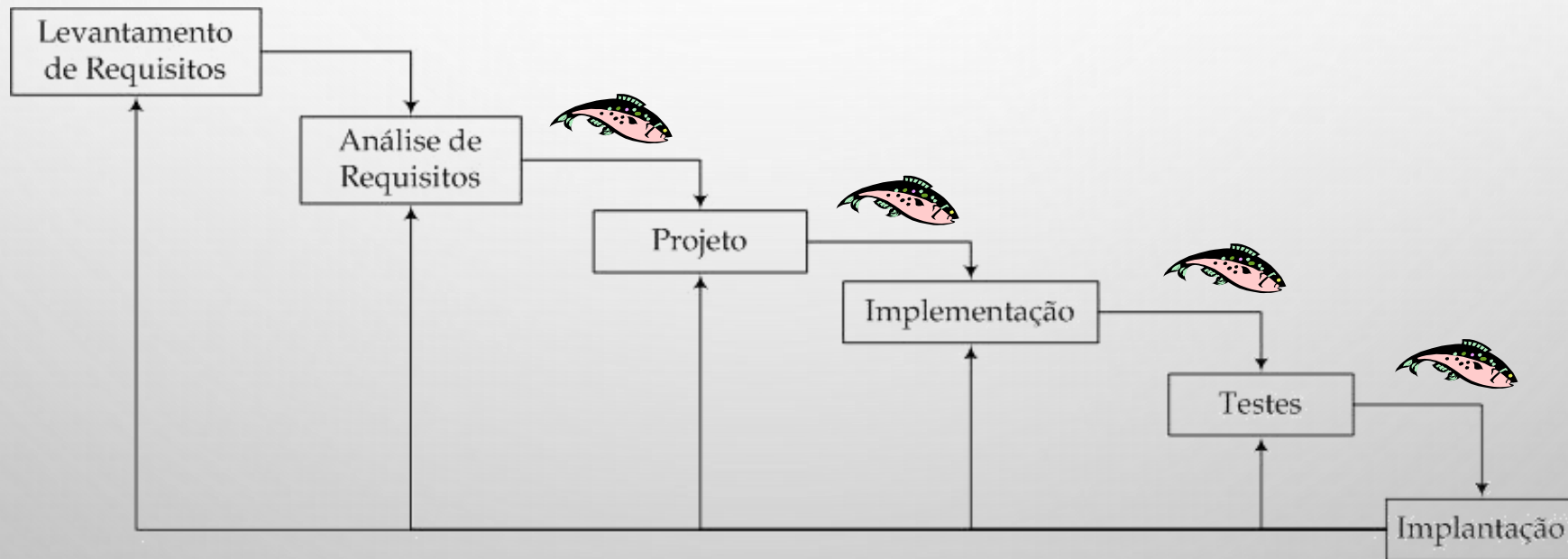
MODELOS DE CICLO DE VIDA

MODELO DE CICLO DE VIDA

- UM CICLO DE VIDA CORRESPONDE A UM ENCADEAMENTO ESPECÍFICO DAS FASES PARA CONSTRUÇÃO DE UM SISTEMA.
- DOIS MODELOS DE CICLO DE VIDA:
 - *MODELO EM CASCATA*
 - *MODELO ITERATIVO E INCREMENTAL.*

MODELO EM CASCATA

- ESSE MODELO APRESENTA UMA TENDÊNCIA PARA A PROGRESSÃO SEQUENCIAL ENTRE UMA FASE E A SEGUINTE.



MODELO EM CASCATA

- PROJETOS REAIS RARAMENTE SEGUEM UM FLUXO SEQUÊNCIAL.
- ASSUME QUE É POSSÍVEL DECLARAR DETALHADAMENTE TODOS OS REQUISITOS ANTES DO INÍCIO DAS DEMAIS FASES DO DESENVOLVIMENTO.
 - PROPAGAÇÃO DE ERROS PELAS AS FASES DO PROCESSO.
- UMA VERSÃO DE PRODUÇÃO DO SISTEMA NÃO ESTARÁ PRONTA ATÉ QUE O CICLO DO PROJETO DE DESENVOLVIMENTO CHEGUE AO FINAL.

MODELO EM CASCATA

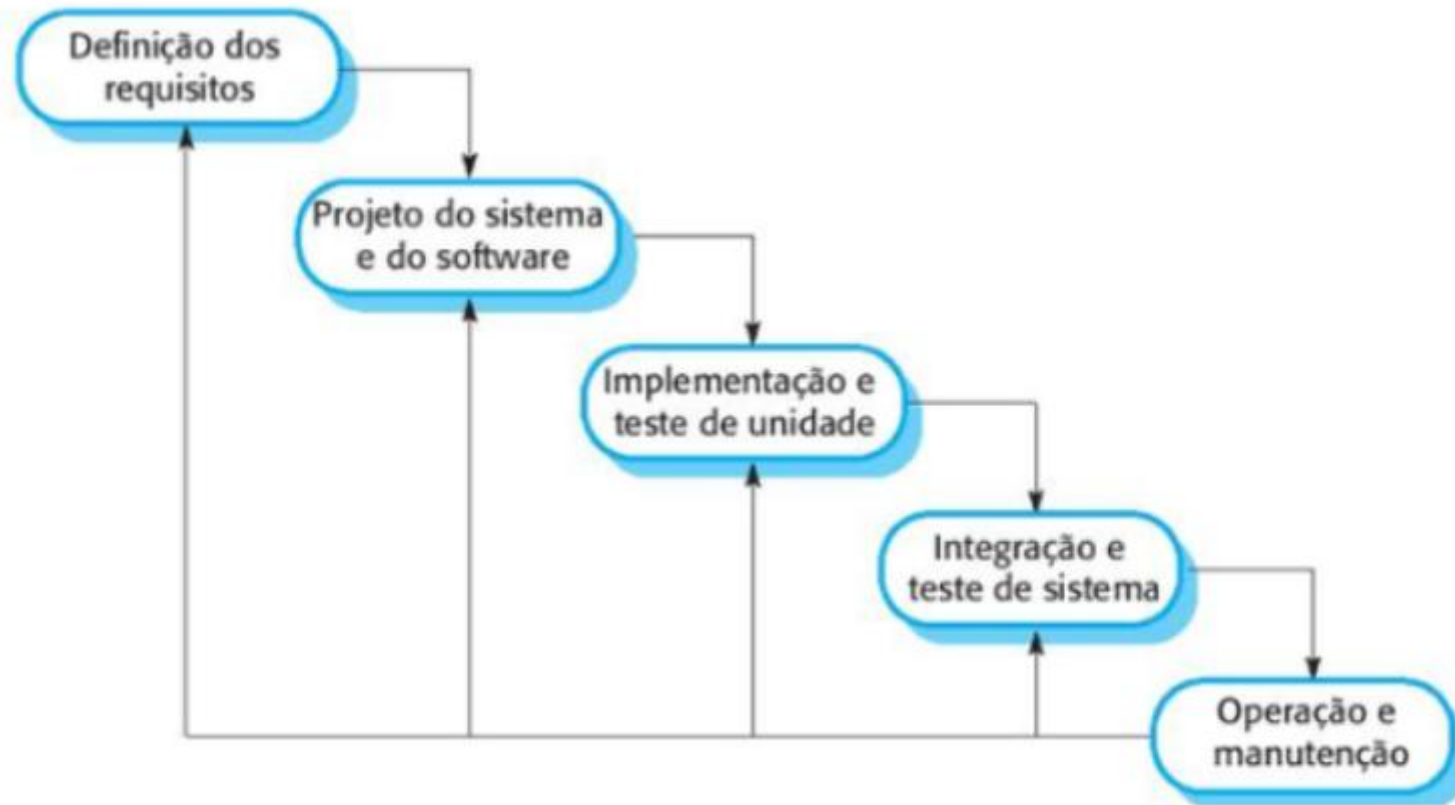


Figura 1 | Modelo cascata. Fonte: Sommerville (2018, p. 33).

MODELO EM CASCATA

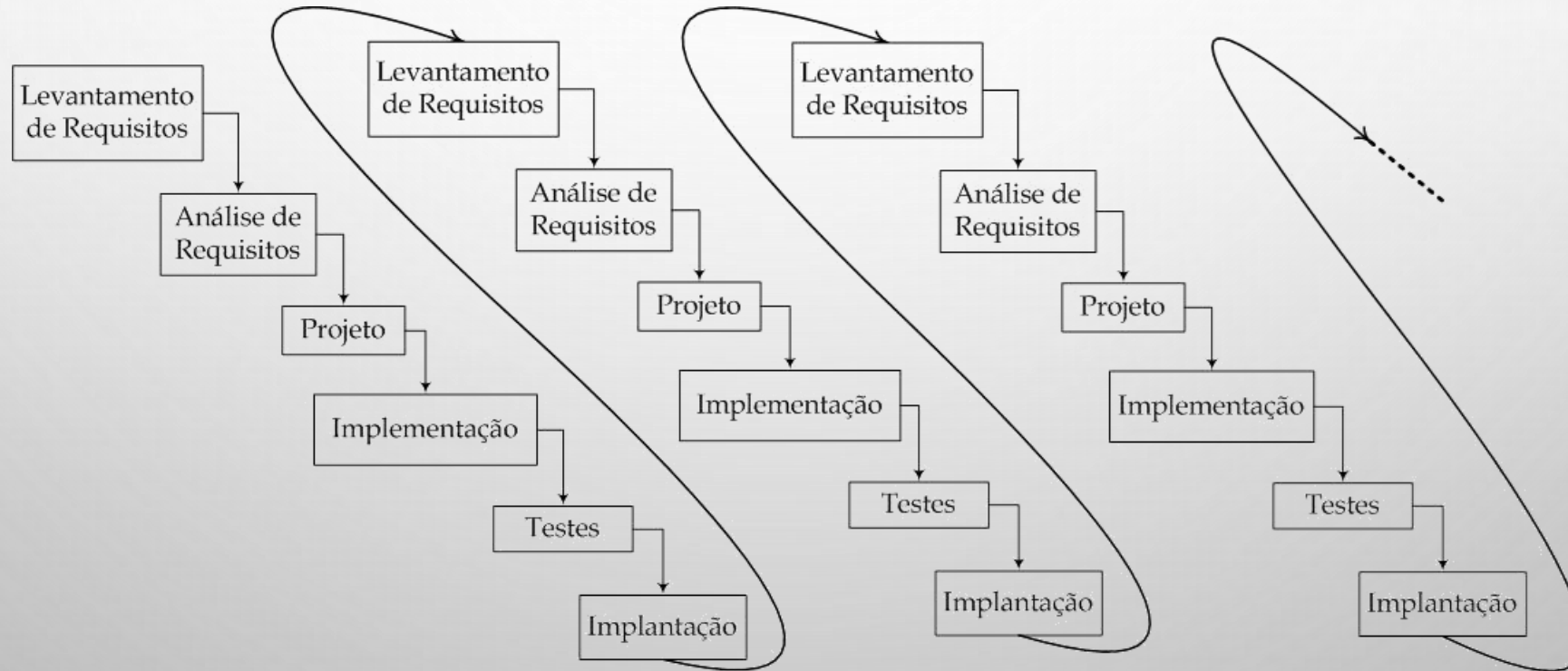
- **Definição de requisitos:** nesta etapa, as funcionalidades do sistema são definidas através de consultas aos usuários. A especificação do sistema é documentada, envolvendo a participação e a aprovação do cliente para garantir que as necessidades sejam atendidas.
- **Projeto do sistema de software:** esta fase inclui o planejamento da estrutura do sistema, designando recursos para o desenvolvimento do software. Aqui, são definidas a estrutura de dados, a arquitetura do software, interfaces gráficas, entre outros aspectos técnicos.
- **Implementação e teste de unidade:** durante a implementação, o software é codificado na linguagem de programação escolhida. Paralelamente, testes unitários são realizados em cada módulo ou parte do código para identificar e corrigir falhas de programação.
- **Integração e teste de sistemas:** após a codificação, todos os módulos do software são integrados e submetidos a testes conjuntos, preparando o software para entrega ao cliente.
- **Operação e manutenção:** com o software em uso, surgem correções e possíveis melhorias solicitadas pelos usuários. Nesta fase, são implementadas as correções e novos requisitos identificados como necessários pelo cliente, assegurando a funcionalidade contínua e aprimoramento do software.

MODELO DE ITERATIVO E INCREMENTAL

- DIVIDE O DESENVOLVIMENTO DE UM PRODUTO DE SOFTWARE EM **CICLOS**.
- EM CADA CICLO DE DESENVOLVIMENTO, PODEM SER IDENTIFICADAS AS FASES DE ANÁLISE, PROJETO, IMPLEMENTAÇÃO E TESTES.
- CADA CICLO CONSIDERA UM SUBCONJUNTO DE REQUISITOS.
- ESTA CARACTERÍSTICA CONTRASTA COM A ABORDAGEM CLÁSSICA, NA QUAL AS FASES SÃO REALIZADAS UMA ÚNICA VEZ.

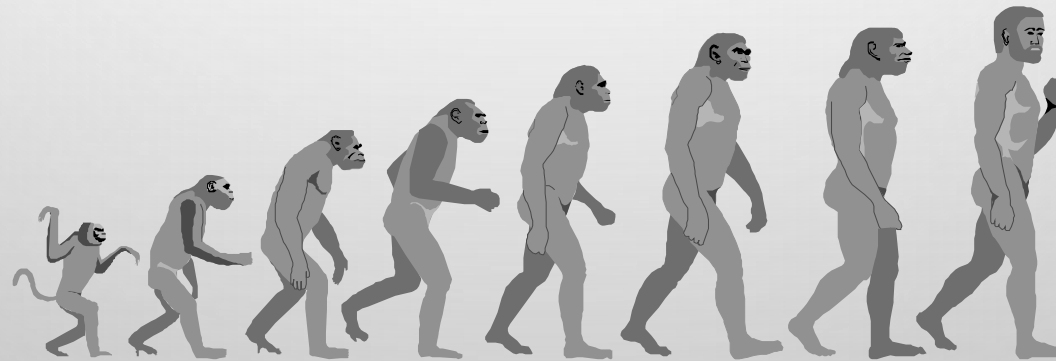
MODELO ITERATIVO E INCREMENTAL

Desenvolvimento em “mini-cascatas”.



MODELO ITERATIVO E INCREMENTAL

- **ITERATIVO:** O SISTEMA DE SOFTWARE É DESENVOLVIDO EM VÁRIOS PASSOS SIMILARES.
- **INCREMENTAL:** EM CADA PASSO, O SISTEMA É ESTENDIDO COM MAIS FUNCIONALIDADES.



MODELO ITERATIVO E INCREMENTAL

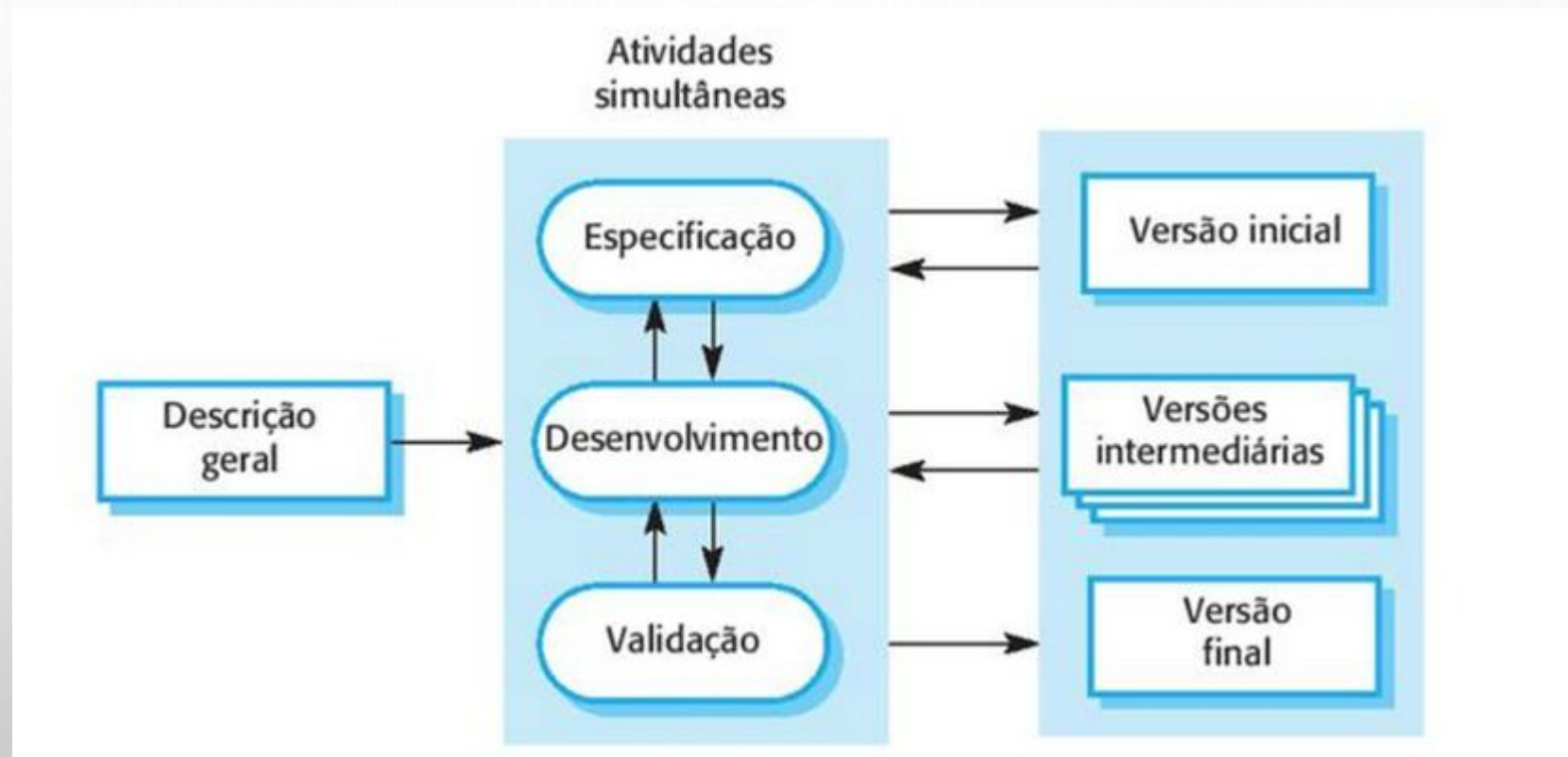


Figura 2 | Modelo de processo prescritivo: modelo incremental. Fonte: Sommerville (2018, p. 35).

MODELO INCREMENTAL

- O **MODELO INCREMENTAL** É ITERATIVO E CONSTRÓI O SOFTWARE EM ETAPAS. A PARTIR DOS REQUISITOS INICIAIS, PEQUENAS VERSÕES DO SOFTWARE SÃO DESENVOLVIDAS E ENTREGUES AO CLIENTE, EXPANDINDO-SE PROGRESSIVAMENTE ATÉ A CONSTRUÇÃO COMPLETA DO SISTEMA IDEAL.
- NESTE MODELO, CADA VERSÃO REPRESENTA UM INCREMENTO, INCLUINDO PARTE DAS FUNCIONALIDADES REQUISITADAS. DIFERENTEMENTE DO MODELO EM CASCATA, NO MODELO INCREMENTAL O CLIENTE RECEBE O SOFTWARE EM VÁRIAS ENTREGAS PARCIAIS AO LONGO DO DESENVOLVIMENTO. ESSAS ENTREGAS CONSISTEM EM MÓDULOS QUE ADICIONAM OU APRIMORAM AS FUNCIONALIDADES DO SISTEMA.

MODELO ITERATIVO E INCREMENTAL – VANTAGENS E DESVANTAGENS

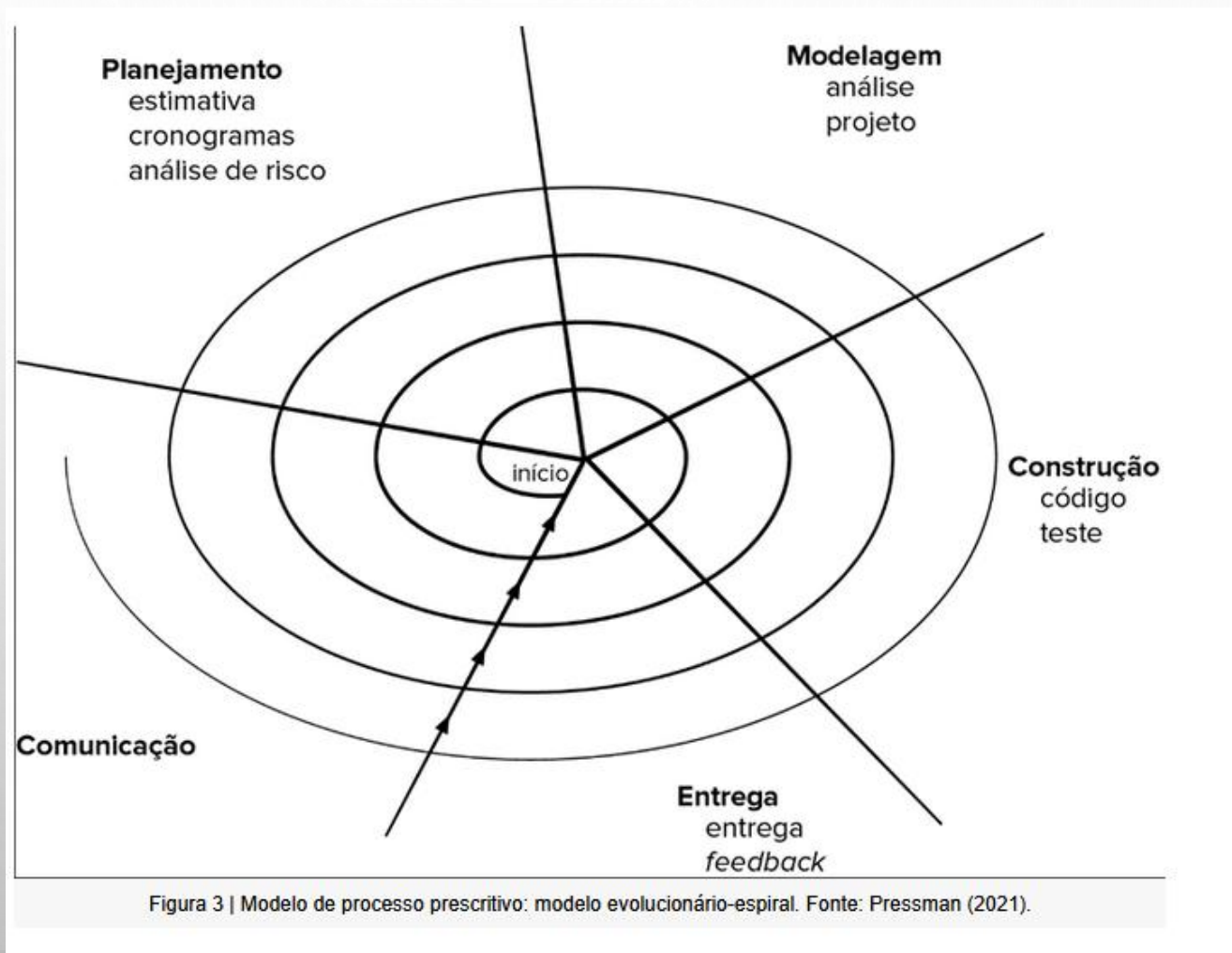
👍 INCENTIVA A PARTICIPAÇÃO DO USUÁRIO.

👍 RISCOS DO DESENVOLVIMENTO PODEM SER MAIS BEM GERENCIADOS.

- UM **RISCO DE PROJETO** É A POSSIBILIDADE DE OCORRÊNCIA DE ALGUM EVENTO QUE CAUSE PREJUÍZO AO PROCESSO DE DESENVOLVIMENTO, JUNTAMENTE COM AS CONSEQUÊNCIAS DESSE PREJUÍZO.
- INFLUÊNCIAS: CUSTOS DO PROJETO, CRONOGRAMA, QUALIDADE DO PRODUTO, SATISFAÇÃO DO CLIENTE, ETC.

☹️ MAIS DIFÍCIL DE GERENCIAR

MODELO ESPIRAL



MODELO ESPIRAL

- A O **MODELO ESPIRAL** É VARIANTE DO **MODELO EVOLUTIVO**, COMBINA A NATUREZA ITERATIVA DA PROTOTIPAÇÃO COM OS ELEMENTOS SISTEMÁTICOS E CONTROLADOS DO MODELO CASCATA, CONFORME DESCRITO POR PRESSMAN (2016). ESTE MODELO VISA O DESENVOLVIMENTO ACELERADO DE VERSÕES DO SOFTWARE, ONDE CADA CICLO ITERATIVO RESULTA EM UMA VERSÃO MAIS AVANÇADA E COMPLETA DO PRODUTO.

PROTOTIPAGEM

- A **PROTOTIPAGEM** É UMA TÉCNICA APLICADA QUANDO:
 - HÁ DIFICULDADES NO ENTENDIMENTO DOS REQUISITOS DO SISTEMA
 - HÁ REQUISITOS QUE PRECISAM SER MAIS BEM ENTENDIDOS.
- A CONSTRUÇÃO DE **PROTÓTIPOS** UTILIZA AMBIENTES COM FACILIDADES PARA A CONSTRUÇÃO DA INTERFACE GRÁFICA.
- PROCEDIMENTO GERAL DA PROTOTIPAGEM:
 - APÓS O LR, UM PROTÓTIPO É CONSTRUÍDO PARA SER USADO NA VALIDAÇÃO.
 - USUÁRIOS FAZEM CRÍTICAS...
 - O PROTÓTIPO É ENTÃO CORRIGIDO OU REFINADO
 - O PROCESSO DE REVISÃO E REFINAMENTO CONTINUA ATÉ QUE O PROTÓTIPO SEJA ACEITO.
 - APÓS A ACEITAÇÃO, O PROTÓTIPO É DESCARTADO OU UTILIZADO COMO UMA VERSÃO INICIAL DO SISTEMA.

PROTOTIPAGEM

- NOTE QUE A PROTOTIPAGEM NÃO É UM SUBSTITUTO À CONSTRUÇÃO DE MODELOS DO SISTEMA.
 - A PROTOTIPAGEM É UMA TÉCNICA COMPLEMENTAR À CONSTRUÇÃO DOS MODELOS DO SISTEMA.
 - MESMO COM O USO DE PROTÓTIPOS, OS MODELOS DO SISTEMA DEVEM SER CONSTRUÍDOS.
 - OS ERROS DETECTADOS NA VALIDAÇÃO DO PROTÓTIPO DEVEM SER UTILIZADOS PARA MODIFICAR E REFINAR OS MODELOS DO SISTEMA.