
BANG!
IDADE DA PEDRA
BANG!
IDADE DO BRONZE
BANG!
IDADE DO FERRO
BANG!
IDADE MÉDIA
BANG!
IDADE MODERNA
BANG!
IDADE DA INFORMÁTICA

Crise do Software

- **Crise de tecnologia** - hardware caminha mais rápido que o software
- **Crise de oferta** - demanda é maior que a capacidade de desenvolvimento
- **Crise de manutenção** - projeto mal feito e recursos escassos não permitem manutenção.

É possível enxergar a crise de sistemas de software nos exemplos: Arianne V, Aeroporto de Denver e American Airlines?

Alguns Exemplos de Problemas de Software

- **Explosão do foguete Arianne V** : possuía no seu código uma rotina destinada ao Arianne IV, e que não deveria ter sido ativada.
- **Aeroporto de Denver ficou fechado durante anos** esperando o funcionamento do sistema de bagagens

  >> Solução: simulação
- **Avião da American Airlines (1995) bateu em uma montanha matando 163 pessoas**, devido à presunção do sistema de que os códigos de navegação do local (América do Sul) eram os mesmos que os da origem (América do Norte).

Engenharia de Software:
Definições

"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" [NAU 69]

"A aplicação prática do conhecimento científico para o projeto e a construção de programas computacionais e a documentação necessária à sua operação e manutenção." [Boehm, 76]

"Abordagem sistemática para o desenvolvimento, a operação e a manutenção de software" [Afnor, 83]

"Conjunto de métodos, técnicas e ferramentas necessárias à produção de software de qualidade para todas as etapas do ciclo de vida do produto." [Krakowiak, 85]

Produto de Software

- **Entidade Abstrata**
- **É um produto transformador de informações***
  - De acordo com a **semiótica**:
    . **dados** são **símbolos com uma sintaxe** e
    . **informação** são **dados com uma semântica**
- **É um veículo para distribuir informações***
  - Exs.: controle do computador (SO), comunicação de informações (redes), criação e controle de outros programas (ferramentas)

Produto de software é um conjunto de ...

(1) ***instruções*** *(programas de computador) que, quando executadas, proveem as características, funcionalidades e desempenho desejados;*

(2) ***estruturas de dados*** *que permitem aos programas manipularem informação de forma eficiente; e*

(3) ***informação descritiva****, tanto impressa quanto digital, descrevendo operação, uso e manutenção dos programas*

Artefatos

- Quaisquer produtos intermediários produzidos durante um projeto de desenvolvimento de software:
  - Diagramas
  - Programas
  - Documentos de texto
  - Desenhos
  - Contratos
  - Projetos
  - Planos
  - …
- Dependendo do Modelo de Ciclo de Vida adotado, cada artefato pode ter somente um dono (o único que pode modificar ou permitir sua modificação)
- Devem estar submetidos a um sistema de Gerenciamento de Configuração para controlar em versões as mudanças sobre os artefatos

Engenharia de Software

No Departamento de Informática:

- Engenharia de Requisitos
- Projeto (Design) de Software
- Sistemas de Informação em Saúde
- Engenharia de Software
- Optativas

Camadas da Engenharia de Software

Camadas da ES

- Ferramentas
- Processos
- Métodos
- Foco na Qualidade

O Processo

- **Processo**: conjunto de atividades, ações e tarefas realizadas na criação do produto
- **Método** (s. m.) (..)

  3. Processo racional para chegar a determinado fim.

  4. Maneira de proceder.

  5. Processo racional para chegar ao conhecimento ou demonstração da verdade.

  6. Obra que contém disposta numa ordem de progressão lógica os principais elementos de uma ciência, de uma arte.

Processo de Software

- No contexto de software (wikipedia, Wazlawick, 11):

  Um processo de engenharia de software é formado por um conjunto de atividades estruturadas em passos parcialmente ordenados, relacionados com artefatos, pessoas, recursos, estruturas organizacionais e restrições, tendo como objetivo produzir e manter os produtos de software finais requeridos

  - Um processo é um conjunto de atividades:

  1) Interdependentes.

  2) Com responsáveis.

  3) Com entradas e saídas definidas.

CMM
Capability Matury Model

- O CMM procura orientar a organização a implementar a melhoria contínua do processo de software:
  - Através de um modelo de 5 níveis, priorizado de forma lógica as ações a serem realizadas.
  - Quanto maior o nível, maior a maturidade da organização, o que se traduz em maior qualidade dos produtos final, prazos e custos mais baixos e maior previsibilidade em cronogramas e orçamentos.

CMM
Capability Matury Model

- O **CMM** procura orientar a organização a implementar a melhoria contínua do processo de software:
  - Através de um **modelo de 5 níveis**, priorizado de forma lógica as ações a serem realizadas.
  - Quanto **maior o nível**, **maior a maturidade** da organização, o que se traduz em **maior qualidade** dos produtos final, **prazos e custos mais baixos** e **maior previsibilidade** em cronogramas e orçamentos.

CMM
Capability Matury Model

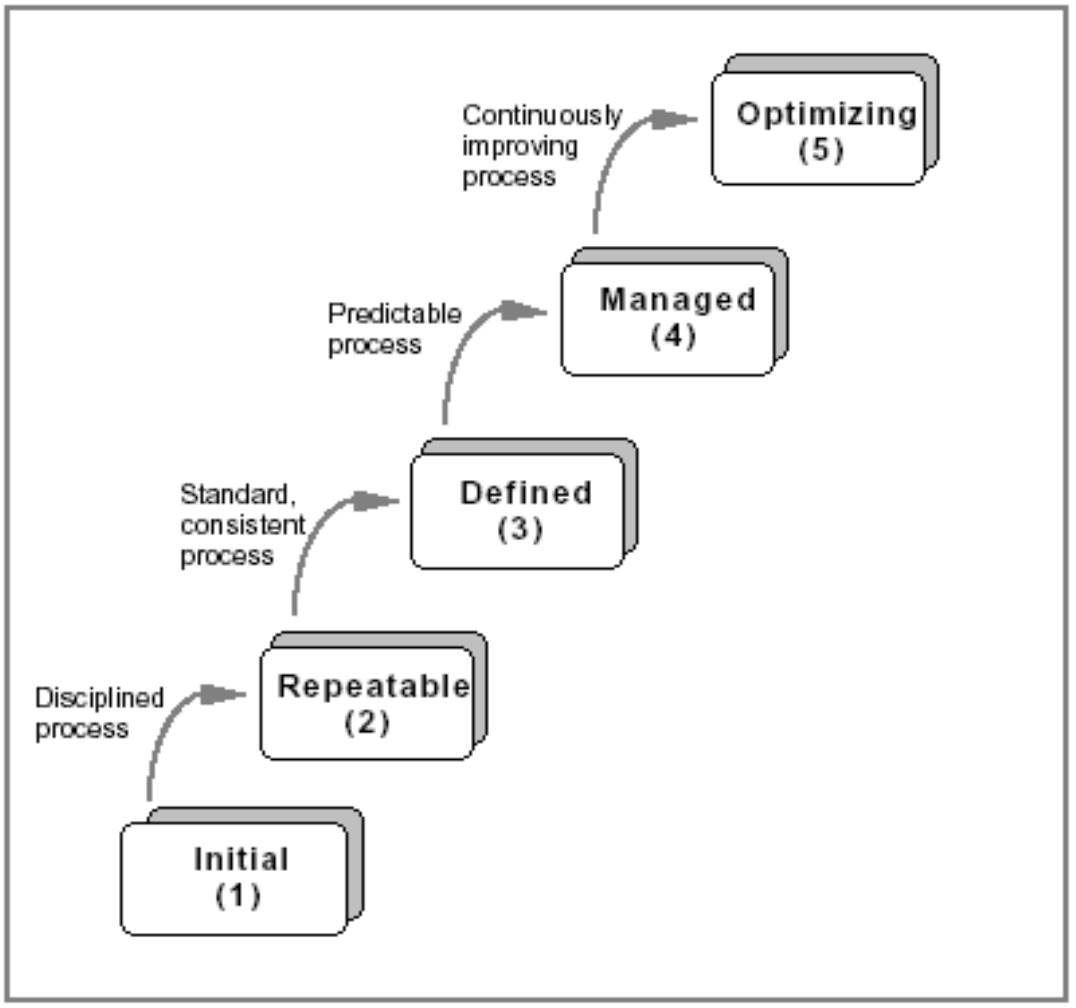


Figure 2.1 The Five Levels of Software Process Maturity

CMM
Capability Matury Model

- No **nível 1 (Inicial)** o desenvolvimento é **caótico (*ad hoc)***.
- Sem procedimentos padronizados, estimativas de custos e planos de projeto.
- Cada qual desenvolve como quer, não existe documentação e não há mecanismos de controle que permitam ao gerente saber o que está acontecendo, identificar problemas e riscos e agir de acordo.
- Os desvios não são corrigidos. Ocorrem problemas como prazos não cumpridos, orçamentos estourados, software sem qualidade e usuários insatisfeitos.
- Cronogramas e orçamentos inexistentes, rudimentares ou inexatos.
- ¾ dos projetos

CMM
Capability Matury Model

- **Para passar ao nível 2 (Repetitível)**, a organização deve **instituir controles básicos de projeto**:
  - **Gerenciamento de Requisitos**
  - **Gerenciamento de Projetos**: técnicas para planejar e estimar o esforço em projetos e controlar o progresso do desenvolvimento + **Controle Gerencial:** verificação pela Gerência do progresso do projeto em momentos pré-determinados + qualidade dos artefatos/produtos
  - instituição de um **Grupo de Garantia de Qualidade**
  - instituição de procedimentos básicos de **Gerenciamento de Configuração** (para garantir que mudanças no projeto e manutenções solicitadas não destruam o que já foi feito, garantindo um mínimo de estabilidade no desenvolvimento).

CMM
Capability Matury Model

- **Chegando ao nível 2 (Repetitível)**:
  - A organização está em **condições de ter maior controle sobre seus projetos. Estimativas** podem ser **mais precisas** (já que se desenvolve uma base histórica intuitiva) e a **qualidade** pode ser **maior**.
- **Problema em permanecer aqui**:
  - caso a empresa enfrente o desafio de desenvolver **projetos de características distintas das que está acostumada** (usando uma nova tecnologia, por exemplo), esta informação será irrelevante, e **a empresa poderá regredir ao nível 1**.

CMM
Capability Matury Model

- **Para passar ao nível 3 (Definido)**:
  - Introduzir uma **Metodologia de Desenvolvimento** formal padronizada, com:
    - um ciclo de vida definido + métodos, técnicas e ferramentas apropriadas, como inspeções e técnicas abrangentes de teste.
  - Estabelecer o **time** encarregado exclusivamente da **melhoria contínua** do processo de software.

CMM
Capability Matury Model

- **Chegando ao nível 3 (Definido):**
  - a empresa terá um **fundamento claro para desenvolver sistemas** e também para **melhorar o próprio processo**, especialmente **quando surgirem crises**.
- **Problema em permanecer aqui**:
  - os **controles** ainda são basicamente **qualitativos. Não há meios de quantificar** a qualidade dos produtos e a eficiência do processo.

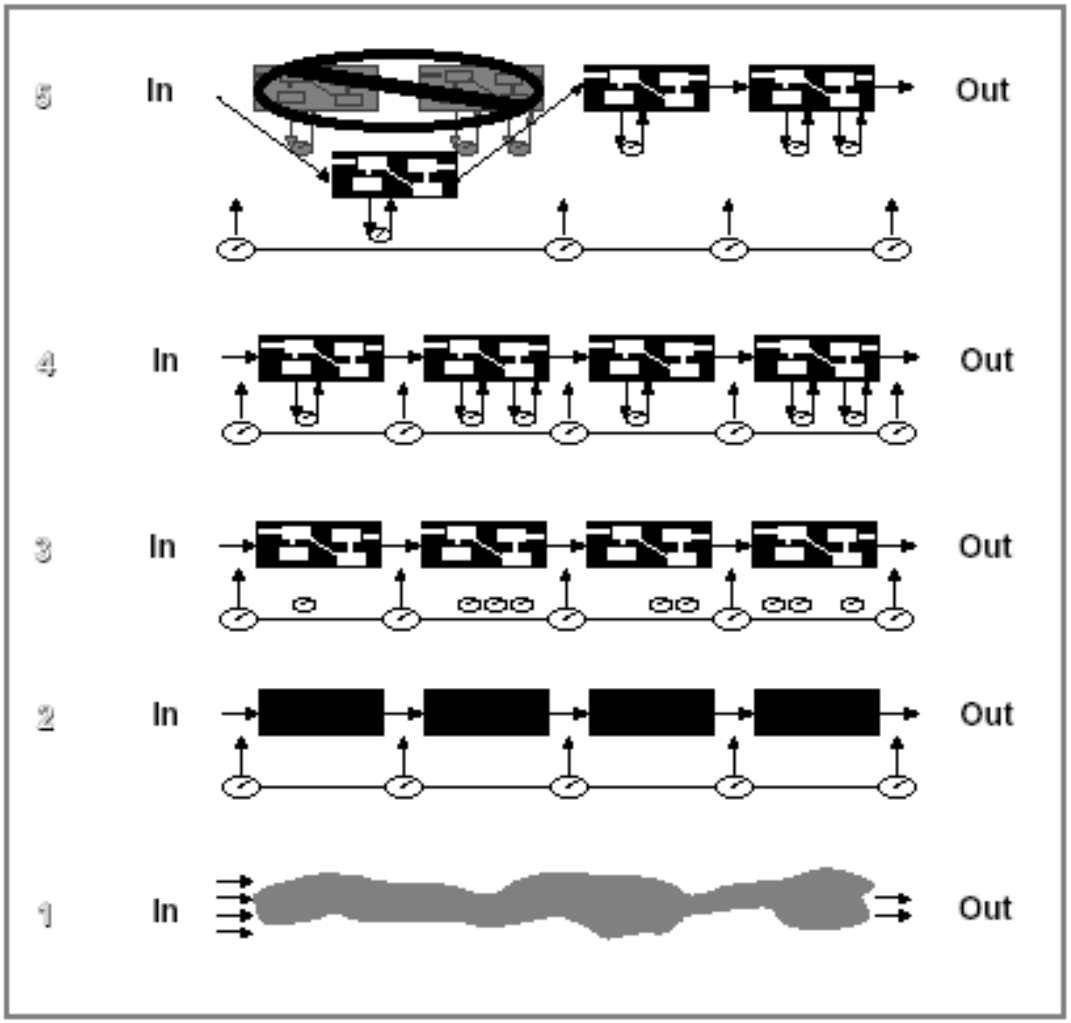
CMM
Capability Matury Model

- **Para passar para o nível 4 (Gerenciado)**
  - A empresa deve **estabelecer métricas** de forma a medir características específicas dos **produtos**.
  - Devem-se **definir a forma de coletar, armazenar e analisar** as métricas e, com base nesta informação, pode-se sugerir melhorias específicas nos produtos.

CMM
Capability Matury Model

- **Para subir para o nível 5 (Otimização):**
  - **Estabelecer meios para a coleta automática de métricas** e para a utilização da informação coletada de forma a prevenir problemas.
  - A idéia é **analisar as causas dos problemas e resolvê-las** para evitar que voltem a ocorrer.
- Enquanto os dados coletados no nível 4 podem informar, por exemplo, quantos erros existem em um programa, a preocupação no nível 5 é melhorar o processo para evitar que tais erros aconteçam no próximo projeto.

CMM
Capability Matury Model



Processo de Software

- Um **processo completo de ES** abrange:
  - Atividades estruturais (estrutura o projeto-project)
  - Atividades de apoio (ao longo do processo de sw)

O Processo de Software

- **Atividades Estruturais**
  1. **Comunicação:** início do projeto, levantamento de requisitos
  2. **Planejamento**: estimar, escalonar, acompanhar (plano de projeto)
  3. **Modelagem:** análise e esboço (desenho) geral e detalhado
  4. **Construção:** geração e testes de código
  5. **Implantação:** entrega do software

O Processo de Software

- **Atividades de Apoio**
  - Controle e acompanhamento do projeto
  - Administração de riscos
  - Garantia de qualidade de software
  - Revisões técnicas
  - Medição
  - Gerenciamento da configuração de software
  - Gerenciamento da reusabilidade
  - Preparo e produção de artefatos do sofware (para cada fase distinta: modelos, documentos, logs, formulários, listagens)

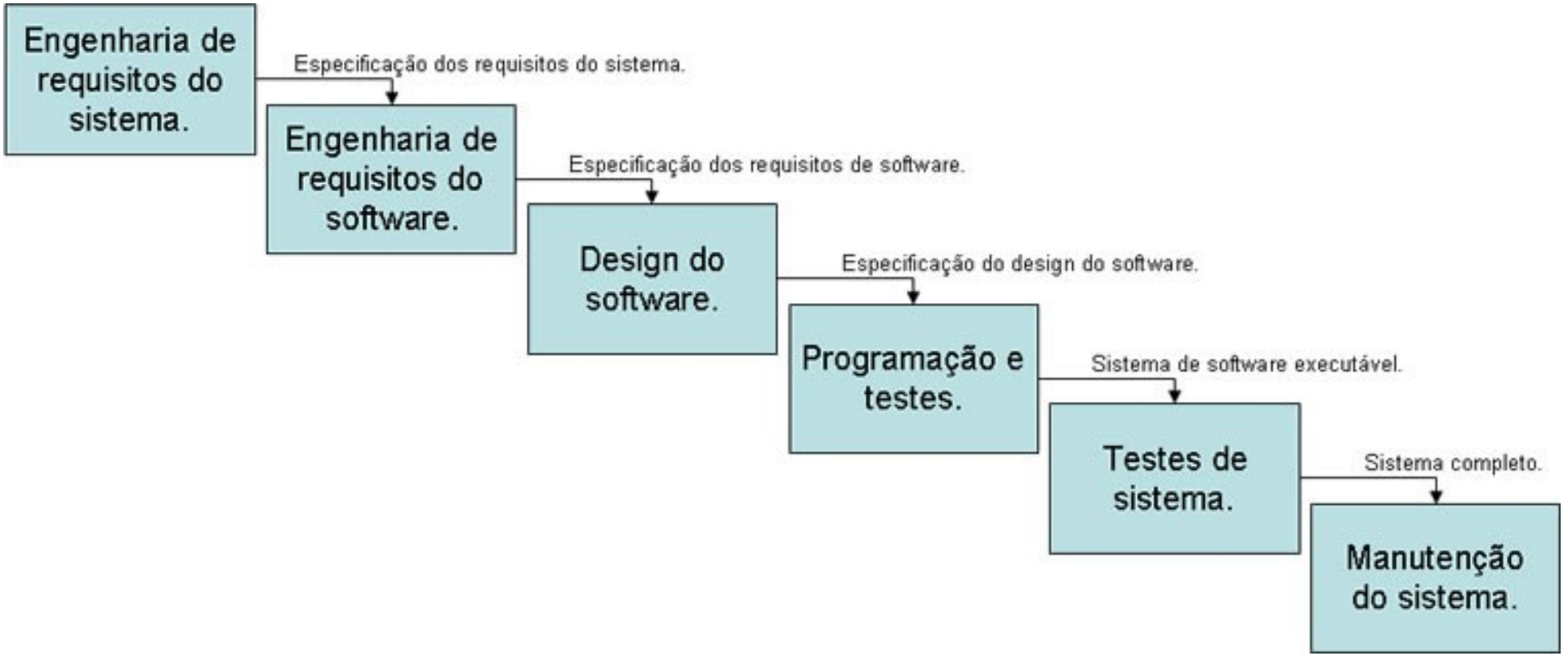
O Processo de Software

- **E a manutenção?**
  - **Atividades estruturais**
  - **Atividades de apoio**

(10 min)

* Qual o **processo** usado por você para **criar um programa**?

O processo clássico de software

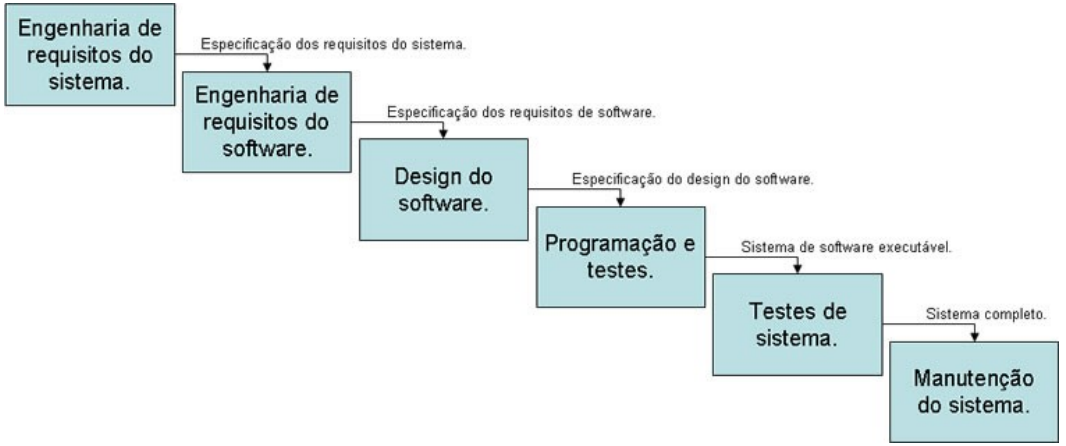


O que fazemos?

- Requisitos do **sistema**: equipes **multidisciplinares + usuários** → COMUNICAÇÃO + MODELAGEM

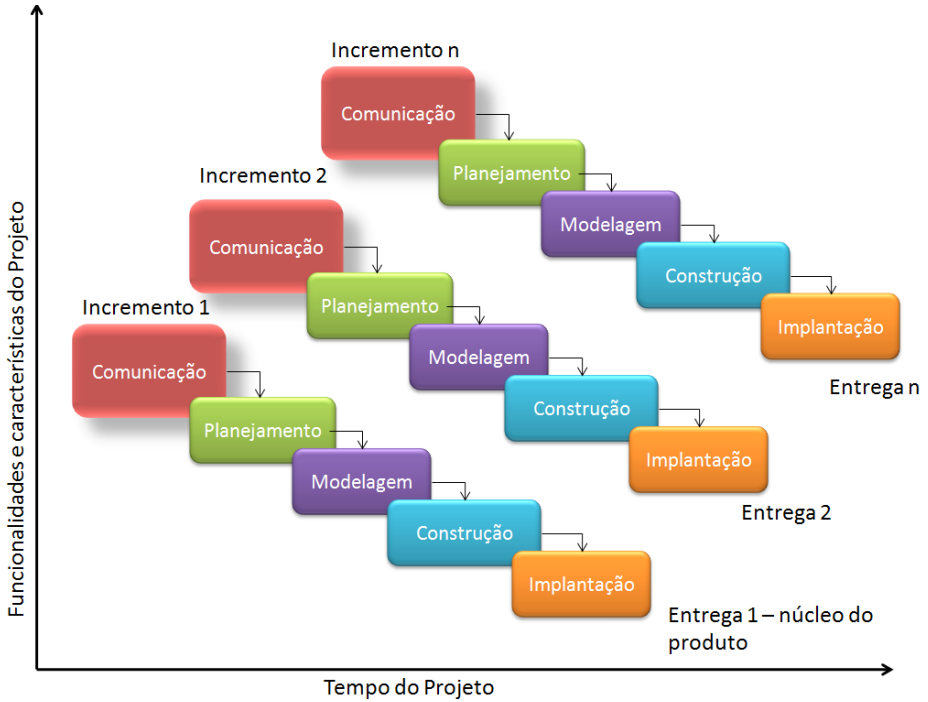
- Requisitos do **software**: equipe de **software + usuários** → COMUNICAÇÃO + MODELAGEM

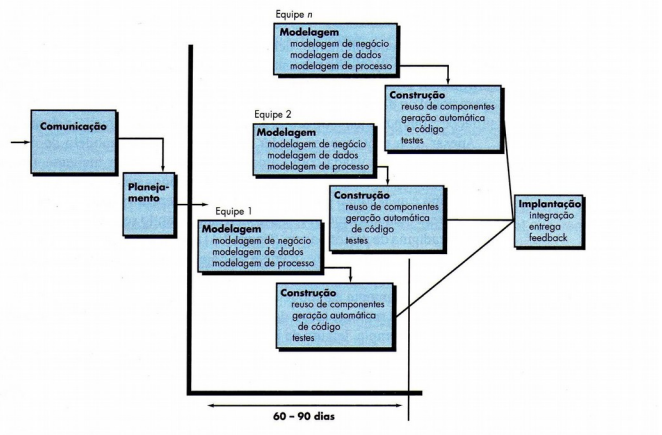
O processo clássico de software

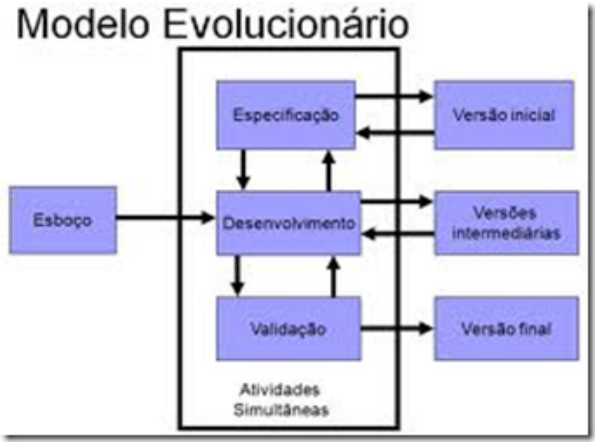


- **Design do software** → equipe de **software.**
  MODELAGEM DA SOLUÇÃO
- **Programação e testes** → equipe de **software.**
  CONSTRUÇÃO E VERIFICAÇÃO
- **Testes de sistema** → equipe de **software + usuários**
  VERIFICAÇÃO
- **Manutenção do sistema** → equipes **multidisciplinares**

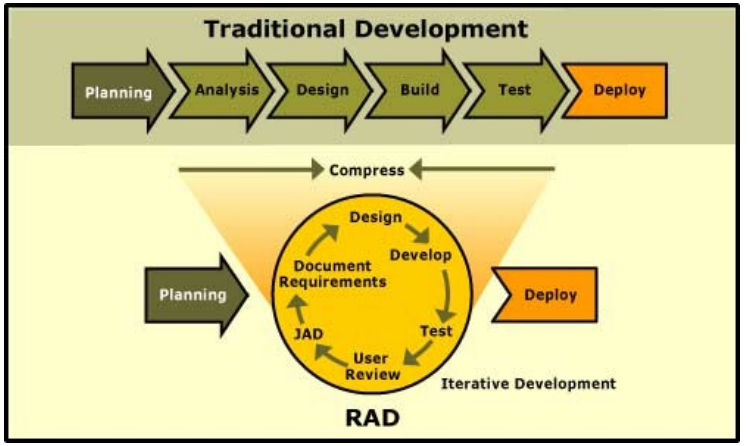
Outros processos...

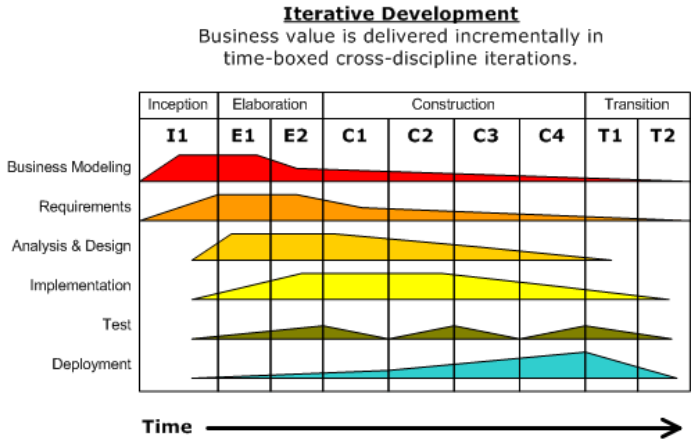


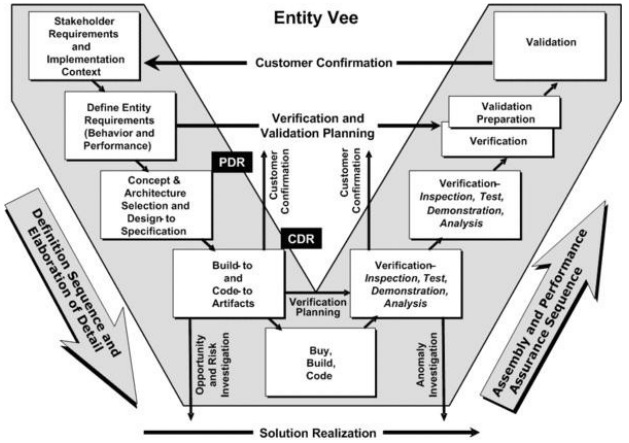


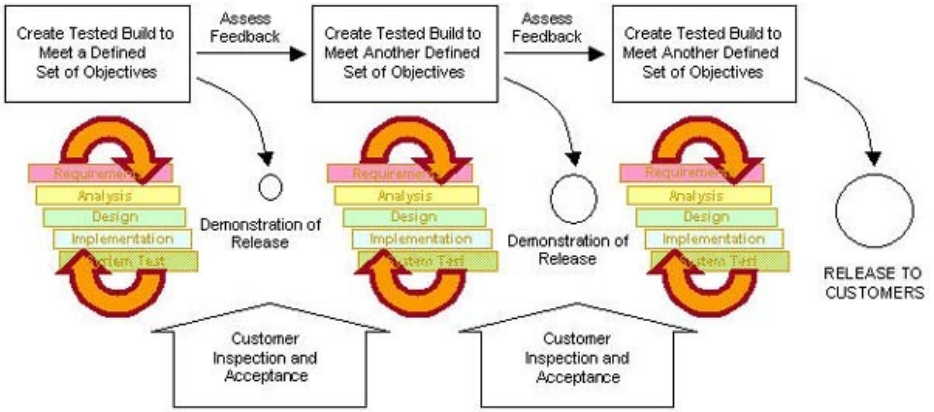


Outros processos...

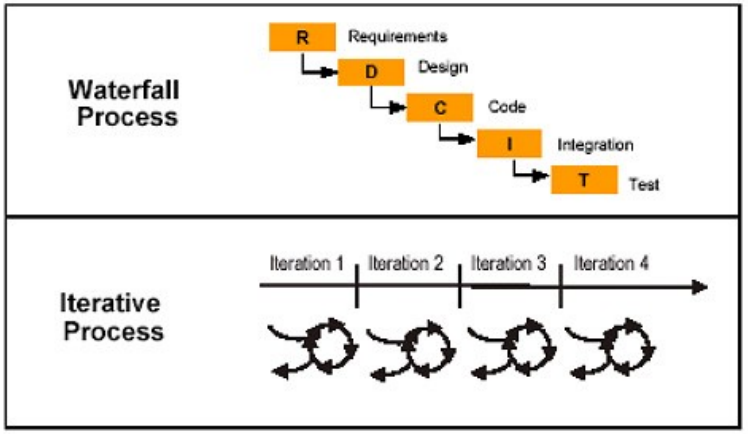


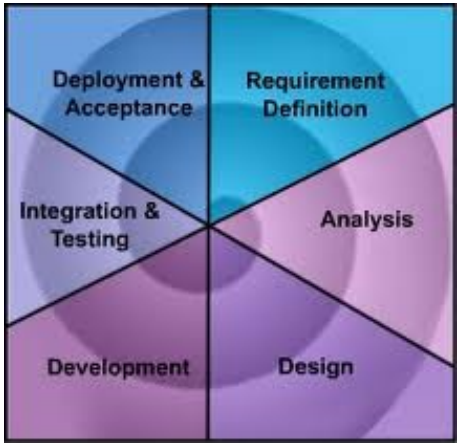






Outros processos...





Fluxos de processos

(Figura: Pressman, 2.2)

- Fluxo de processo linear
- Fluxo de processo iterativo
- Fluxo de processo evolucionário
- Fluxo de processo paralelo