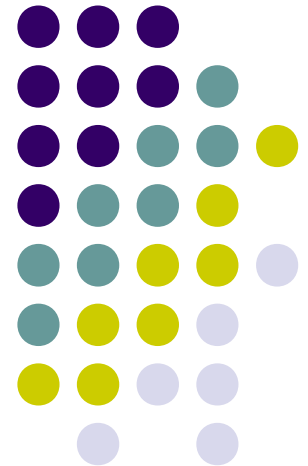


**Introdução aos Sistemas de Informação (ISI) (025607)  
e  
Engenharia de Software 1 (ES1) (1001530)**



# **Histórico da Engenharia de Software**



Prof. Fabiano Cutigi Ferrari  
2º semestre de 2020

ENPE – Ensino não presencial emergencial

# Recados Iniciais

---



- Já sabem né?

Mantenham seus e-mails atualizados no Moodle.

# Roteiro

---

- Características e Evolução do Software
- Crise e Mitos do Software
- Engenharia de Software

# Características e Evolução do Software

# Características do Software

---

- Composto por três elementos principais:
  - **INSTRUÇÕES**, que quando executadas produzem a função e o desempenho desejados,
  - **ESTRUTURAS DE DADOS**, que possibilitam que os programas manipulem adequadamente a informação; e
  - **DOCUMENTOS**, que descrevem a operação e o uso dos programas.

# Características do Software

---

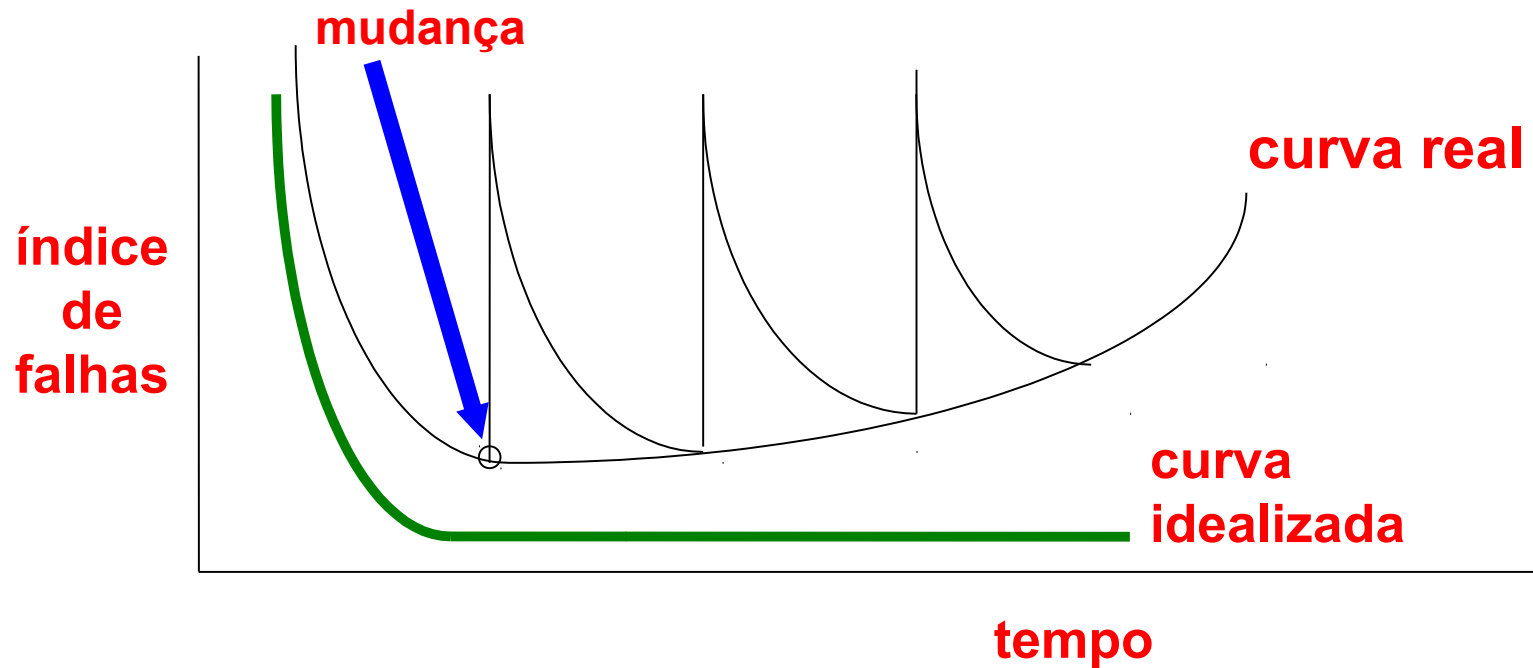
- Desenvolvido ou projetado por engenharia, não manufaturado no sentido clássico.
- Não se desgasta, se deteriora.
- A maioria é feita sob medida em vez de ser montada a partir de componentes existentes.
- Parcialmente reutilizável, mas dificilmente por completo.

# Características do Software



**CURVA DE FALHAS DO HARDWARE**

# Características do Software



**CURVA DE FALHAS DO SOFTWARE**



# Evolução do Software

---

(1950 - 1965)

- O hardware sofreu contínuas mudanças.
- O software era uma arte "secundária" para a qual havia poucos métodos sistemáticos.
- O hardware era de propósito geral.
- O software era específico para cada aplicação.
- Não havia documentação.

# Evolução do Software

---

(1965 - 1975)

- Multiprogramação e sistemas multiusuários.
- Técnicas interativas.
- Sistemas de tempo real.
- 1ª geração de SGBD's.
- Produto de software - software houses.
- Bibliotecas de Software.

# Evolução do Software

---

(1965 - 1975) (continuação..)

- Cresce o número de sistemas baseado em computador.
- Manutenção quase impossível.

..... ***CRISE DE SOFTWARE***

mais adiante...

# Evolução do Software

---

(1975 - 2000 ?)

- Sistemas distribuídos.
- Redes locais e globais.
- Uso generalizado de microprocessadores - produtos inteligentes.
- Hardware de baixo custo.
  - Impacto de consumo.

# Evolução do Software

---

## (2000... Quarta era do software)

- Tecnologias orientadas o objetos.
- Sistemas especialistas e software de inteligência artificial usados na prática.
- Software de rede neural artificial.
- Computação Paralela.
  - Grids, clouds, etc.

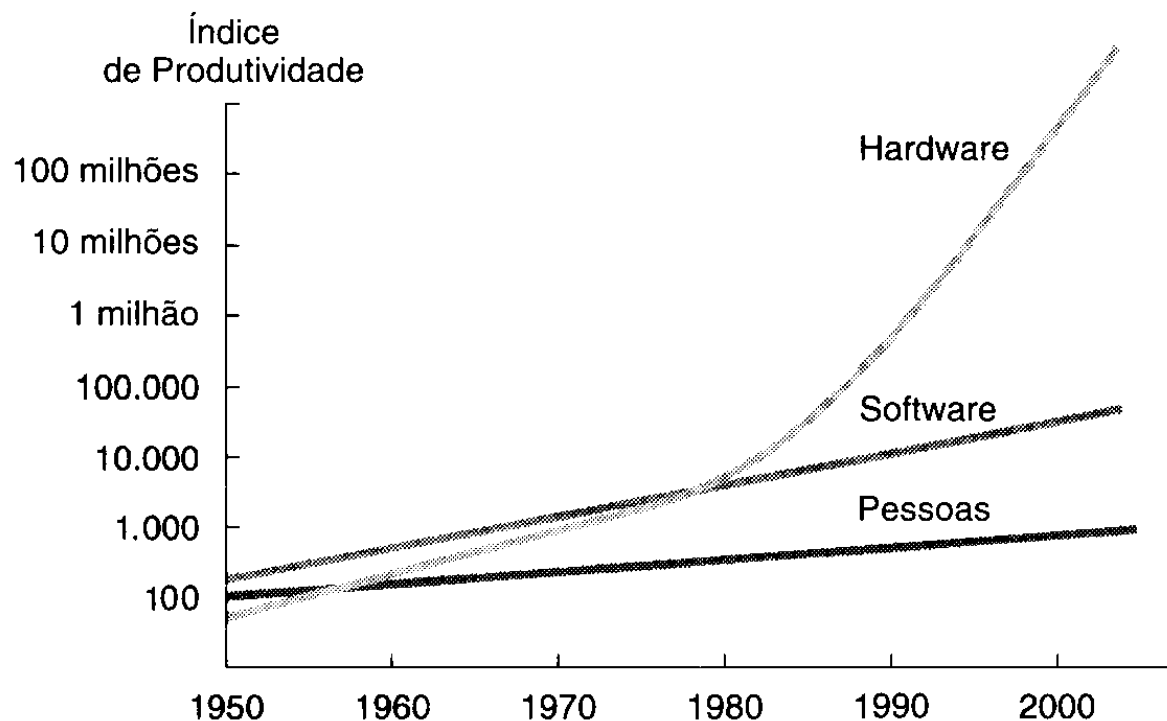
# Perguntas

---

- Quem utilizou um software hoje?
- Qual?
- Como?
- Qual a influência disso no seu dia?

# Características do Software

- Apesar da evolução do software...
- A habilidade em construir software deixa a desejar em relação ao potencial do hardware.



# Características do Software

---

- Apesar da evolução do software...
  - A construção de software não é rápida o suficiente para atender as necessidades do mercado.
  - A sociedade depende cada vez mais de software confiável;
    - Falhas podem causar prejuízos enormes.
  - O esforço para construir software confiável e de qualidade é muito grande.
  - Programas existentes originaram-se de projetos pobres e recursos inadequados.



# Crise e Mitos do Software

# Crise do Software

---

- Refere-se a um conjunto de **problemas** observados no desenvolvimento de software identificado nos anos 60 e 70.
- Originalmente discutida na NATO Software Engineering Conference em 1968
  - <<http://homepages.cs.ncl.ac.uk/brian.randell/NATO/>>
    - Relatórios das 2 edições (1968 e 1969) disponíveis no website.

# Crise do Software

---

- Refere-se a um conjunto de problemas observados no desenvolvimento de software identificado nos anos 60 e 70.
  - Estimativas imprecisas de prazo e custo.
  - Insatisfação do cliente.
  - Qualidade questionável.
  - Difícil manutenção.

# Crise do Software

- Refere-se a um conjunto de problemas encontrados no desenvolvimento de software identificado nos anos 60 e 70.

- **Estimativas imprecisas de prazo e custo.**

- Insatisfação

"Não dedicamos tempo para coletar dados

- Quando sobre o processo de desenvolvimento de software."

- Dificuldade

"Sem nenhuma indicação sólida de produtividade, não podemos avaliar com precisão a eficácia de novas ferramentas, métodos ou padrões."

# Crise do Software

- Refere-se a um conjunto de problemas encontrados no desenvolvimento de software identificado nos anos 60 e 70.
- Estimativas imprecisas de prazo e custo.
- **Insatisfação do cliente.**
- Qualidade questionável.
- Dificuldade de comunicação.
  - “Os projetos de desenvolvimento de software normalmente são efetuados apenas com um vago indício das exigências do cliente.”

# Crise do Software

- Refere-se ao período de crise no desenvolvimento de software nos anos 60.

Só recentemente começam a surgir conceitos quantitativos sólidos de garantia de qualidade de software

- Estimativas imprecisas de prazo e custo.
- Insatisfação do cliente.
- **Qualidade questionável.**
- Difícil manutenção.

# Crise do Software

- Refere-se a um conjunto de problemas encontrados no desenvolvimento de software nos anos 60.
- Estimativa incorreta do tempo necessário para a manutenção.
- Insatisfação com a qualidade do software.
- Qualidade questionável.
- **Difícil manutenção.**

"A tarefa de manutenção devora o orçamento destinado ao software."

"A facilidade de manutenção não foi enfatizada como um critério importante."

# Crise do Software: Causas

---

- **Falhas das pessoas responsáveis pelo desenvolvimento do software:**
  - Gerentes sem nenhum *background* em software.
  - Profissionais da área de software têm pouco treinamento formal em novas técnicas para o desenvolvimento de software.
  - Resistência a mudanças.



# Crise do Software: Causas

---

- **Mitos do software**, que propagam desinformação e confusão:
  - Sob o ponto de vista dos **administradores**.
  - Sob o ponto de vista dos **clientes**.
  - Sob o ponto de vista dos **desenvolvedores**.

# Mitos do Software:

## Administrativos

---

### Mito 1:

- “Já temos um manual repleto de padrões e procedimentos para a construção de software. Isso oferecerá ao meu pessoal tudo o que eles precisam saber.”

### Realidade:

- Será que o manual é usado?
- Os profissionais sabem que ele existe?
- Ele reflete a prática moderna de desenvolvimento de software?
- Ele é completo?

# Mitos do Software: Administrativos

---



## Mito 2:

- “Meu pessoal tem ferramentas de desenvolvimento de software de última geração.”

## Realidade:

- É preciso muito mais do que os mais recentes computadores e ferramentas para se fazer um desenvolvimento de software de alta qualidade.

# Mitos do Software:

## Administrativos

---

### Mito 3:

- “Se nós estamos atrasados nos prazos, podemos adicionar mais programadores e recuperar o atraso.”

### Realidade:

- O desenvolvimento de software não é um processo mecânico igual à manufatura. Acrescentar pessoas em um projeto pode torná-lo ainda mais atrasado.
- Pessoas podem ser acrescentadas, mas somente de forma planejada.

# Mitos do Software: Cliente

---

## Mito 1:

- “Uma declaração geral dos objetivos é suficiente para começar a escrever programas . Podemos preencher os detalhes mais tarde.”

## Realidade:

- Uma definição inicial ruim é a principal causa de fracassos dos esforços de desenvolvimento de software.
- É fundamental uma descrição formal e detalhada do domínio da informação, função, desempenho, interfaces, restrições de projeto e critérios de validação.

# Mitos do Software: Cliente

---



## Mito 2:

- “Os requisitos de projeto modificam-se continuamente, mas as mudanças podem ser facilmente acomodadas, porque o software é flexível.”

## Realidade:

- Uma mudança, quando solicitada tardiamente num projeto, pode ser maior do que a ordem de magnitude mais dispendiosa da mesma mudança solicitada nas fases iniciais.

# Mitos do Software: Cliente

## Mito 2:

### ***MAGNITUDE DAS MUDANÇAS***

FASES	CUSTO DE MANUTENÇÃO
D EFINIÇÃO	1 x
D ESENVOLVIMENTO	1.5 - 6x
M ANUTENÇÃO	60 - 100x

fases iniciais.

# Mitos do Software: Desenvolvedores

---

## Mito 1:

- “Assim que escrevermos o programa e o colocarmos em funcionamento nosso trabalho estará completo.”

## Realidade:

- Os dados da indústria indicam que entre 50 e 70% de todo esforço gasto num programa serão despendidos depois que ele for entregue pela primeira vez ao cliente.



# Mitos do Software: Desenvolvedores

---

## Mito 2:

- “Enquanto não tiver o programa 'funcionando', eu não terei realmente nenhuma maneira de avaliar sua qualidade.”

## Realidade:

- Um programa funcionando é somente uma parte de uma Configuração de Software que inclui todos os itens de informação produzidos durante a construção e manutenção do software.

# Sob a Perspectiva da Indústria

- Já há algum tempo, o software passou a custar mais do que o hardware.
- Gerentes e técnicos se perguntam:
  - Por que é preciso tanto tempo para terminar os programas?
  - Por que os custos são tão altos?
  - Por que não se consegue encontrar todos os defeitos antes que o software seja liberado para os clientes?
  - Por que existe uma dificuldade em medir o progresso à medida que o software está sendo construído?
- A preocupação em resolver essas questões tem levado à adoção das práticas da **Engenharia de Software**.

Engenharia de Software

# Engenharia de Software:

## Uma Resposta à Crise do Software

---



- Definição\*:

A aplicação de uma abordagem sistemática, disciplinada e possível de ser medida para o desenvolvimento, operação e manutenção do software.

\* IEEE - *Institute of Electrical and Electronics Engineers*

# Engenharia de Software:

## Uma Resposta à Crise do Software

---

- Definição\*:

A aplicação de uma **abordagem sistemática**, disciplinada e possível de ser medida para o desenvolvimento, operação e manutenção do software.

\* IEEE - 1016

**Processo de Software**

# Fases Genéricas dos Modelos de Processo de ENGENHARIA

---

- **Especificação** - estabelecer os requisitos e restrições do sistema.
- **Projeto** - produzir um modelo documentado do sistema.
- **Implementação** - construir o sistema.
- **Teste** - verificar se o sistema atende às especificações requeridas.
- **Instalação** (ou implantação) - liberar o sistema para o cliente e garantir que ele se torne operacional.
- **Manutenção** - eliminar defeitos e evoluir o sistema conforme demanda.

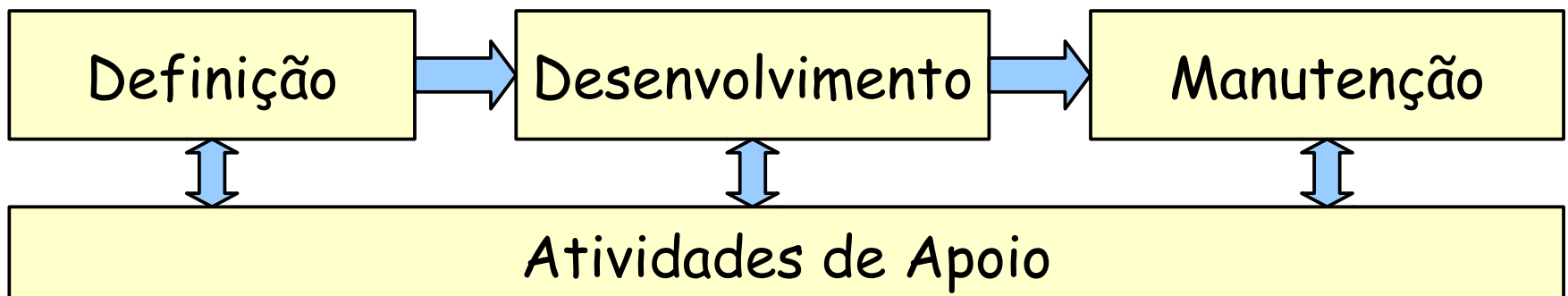
# Fases Genéricas dos Modelos de Processo de ENGENHARIA

- Exemplo: Processo de construção de uma casa

Fase	Execução
Especificação	Casa com 3 dormitórios, 1 sala de jantar, 1 sala de estar, 2 banheiros e 1 área de serviço.
Projeto	Planta baixa e alta da casa, planta hidráulica, planta elétrica.
Implementação	Executar a obra de construção da casa.
Teste	Apresentar a casa para aprovação do cliente.
Instalação	Entrega da casa ao cliente.
Manutenção	Correção de defeitos e implementação de melhorias (e.g. fazer uma cobertura na área externa, consertar um vazamento).

# Fases Genéricas dos Modelos de Processo de SOFTWARE

- Independentemente da natureza do projeto e aplicação, os modelos de processo de software possuem:
  - fase de definição
  - fase de desenvolvimento
  - fase de manutenção
  - atividades de apoio





# Fases Genéricas dos Modelos de Processo de SOFTWARE



**Focaliza "o que" será desenvolvido**

- Qual dado/informação será processado?
- Qual função e desempenho são desejados?
- Qual comportamento pode ser esperado do sistema?

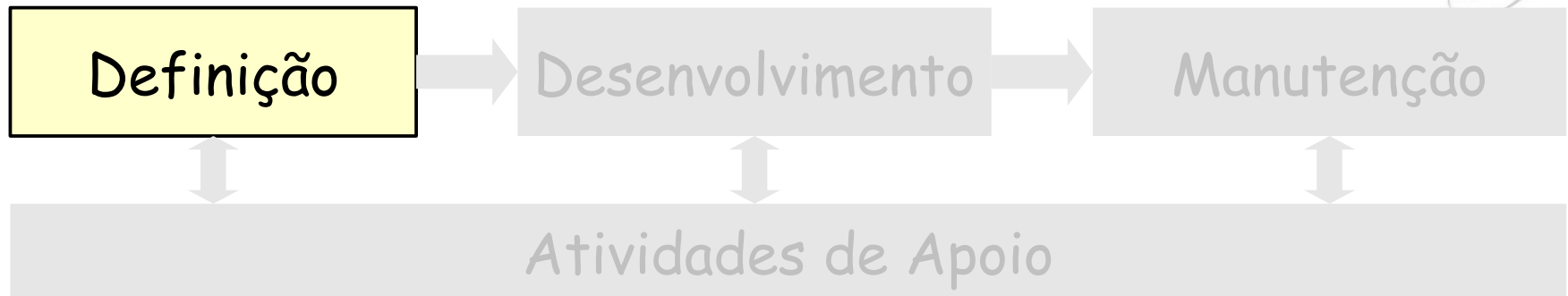
# Fases Genéricas dos Modelos de Processo de SOFTWARE



Focaliza "o que" será desenvolvido

- Quais interfaces serão estabelecidas?
- Quais restrições de projeto existem?
- Quais critérios de validação são exigidos para definir um sistema bem sucedido?

# Fases Genéricas dos Modelos de Processo de SOFTWARE



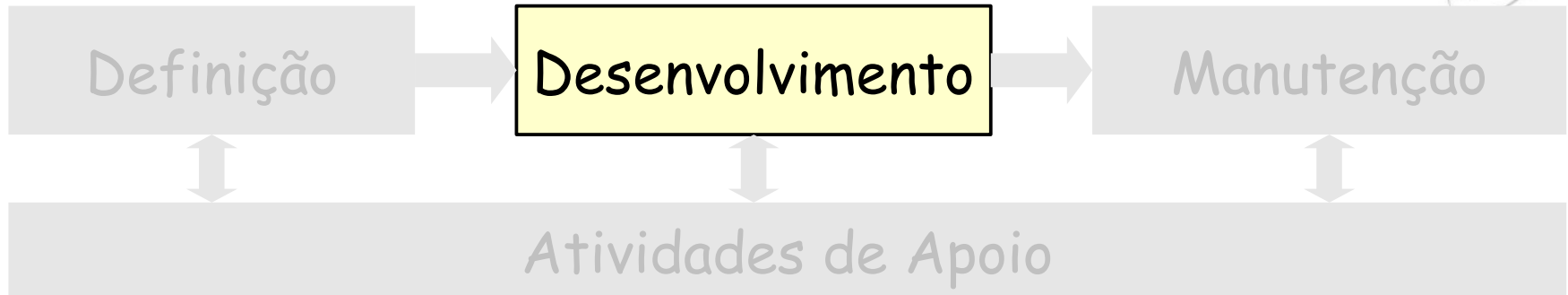
Três tarefas principais ocorrem de alguma forma:

engenharia de sistemas

planejamento do projeto de software

análise de requisitos

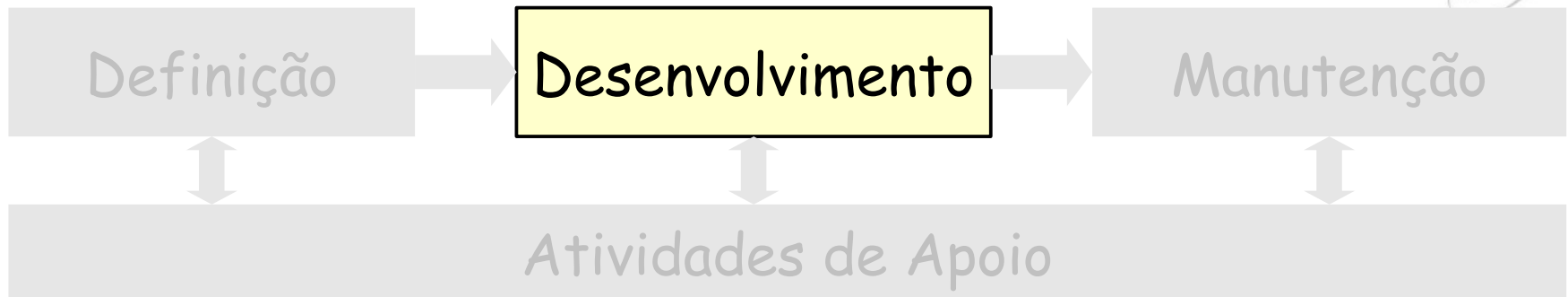
# Fases Genéricas dos Modelos de Processo de SOFTWARE



Focaliza "como" o software será desenvolvido.

- Como os dados serão estruturados?
- Como a função será implementada em uma arquitetura de software?
- Como os detalhes procedimentais serão implementados?

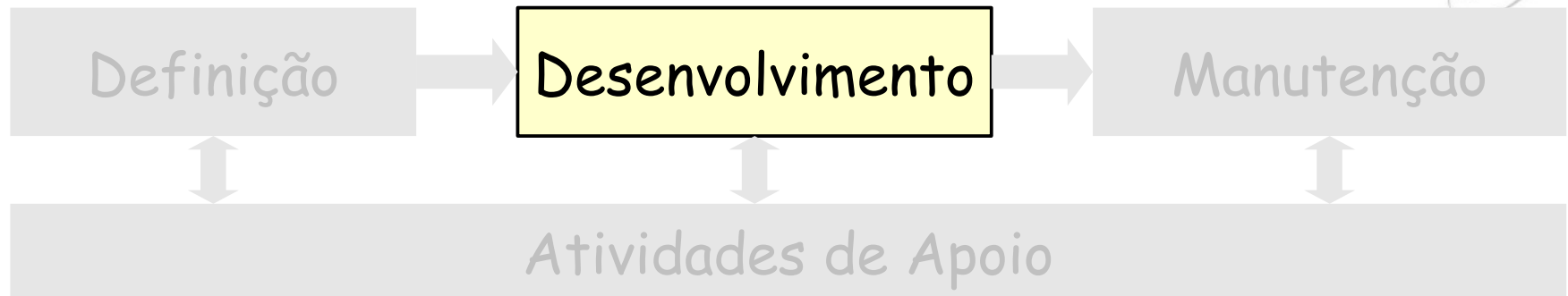
# Fases Genéricas dos Modelos de Processo de SOFTWARE



Focaliza "como" o software será desenvolvido.

- Como as interfaces serão caracterizadas?
- Como o projeto será traduzido em uma linguagem de programação?
- Como os testes serão efetuados?

# Fases Genéricas dos Modelos de Processo de SOFTWARE



Três tarefas técnicas específicas ocorrem sempre:

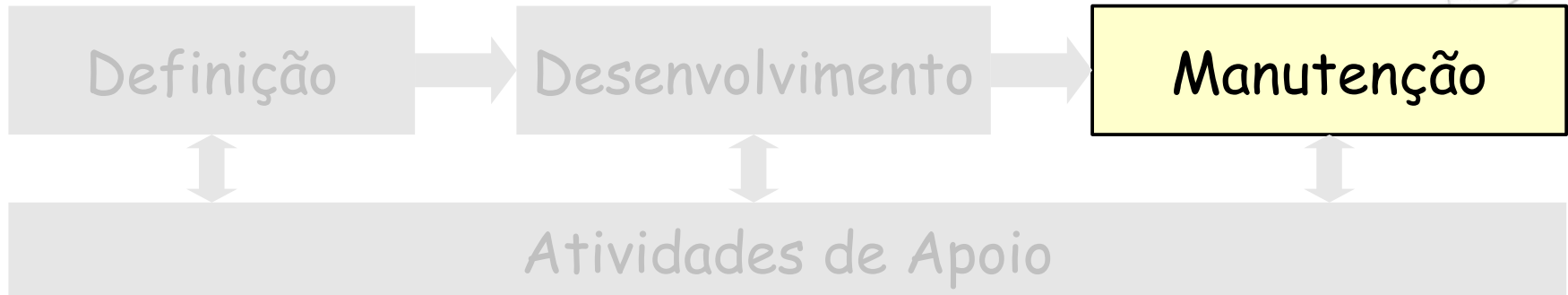
projeto de software

produção de código

inspeção e teste

instalação

# Fases Genéricas dos Modelos de Processo de SOFTWARE



Focaliza as "mudanças" que ocorrerão depois que o software for liberado para uso:

- A fase de manutenção reaplica os passos das fases de definição e desenvolvimento, mas faz isso no contexto de um software existente.

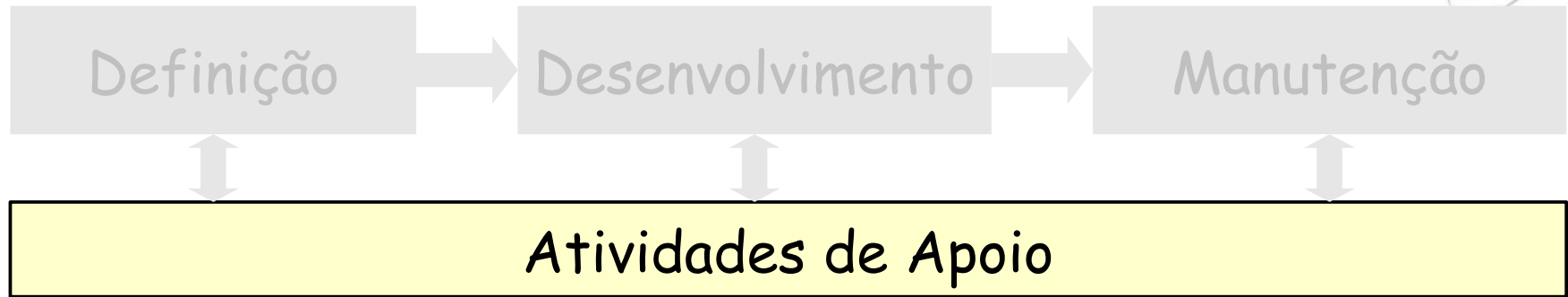
# Fases Genéricas dos Modelos de Processo de SOFTWARE



- As **mudanças** estão associadas com:
  - Correção de defeitos.
  - Adaptações exigidas conforme o ambiente do software evolui.
  - Aperfeiçoamentos para incluir funções adicionais
  - Modificações preventivas para evitar erros e facilitar modificações futuras.

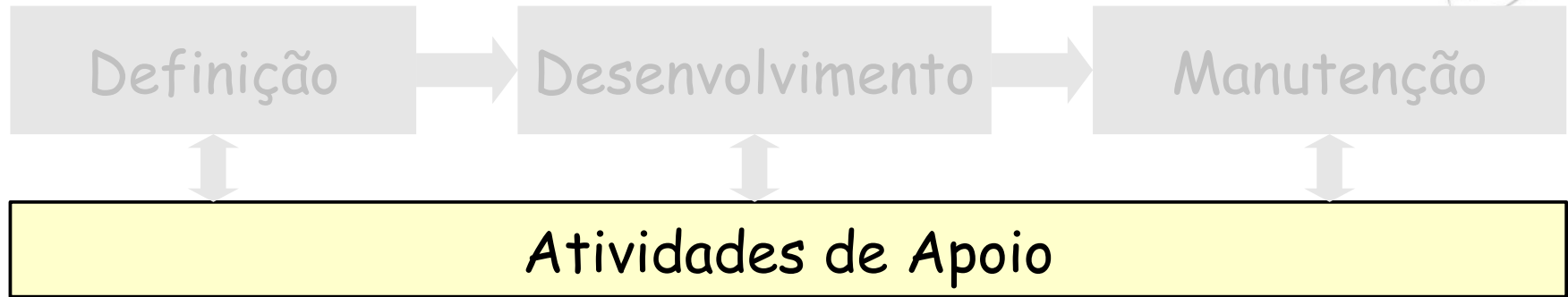


# Fases Genéricas dos Modelos de Processo de SOFTWARE



- Complementam as três fases genéricas de um processo de software.
- São aplicadas durante toda a engenharia do software.

# Fases Genéricas dos Modelos de Processo de SOFTWARE



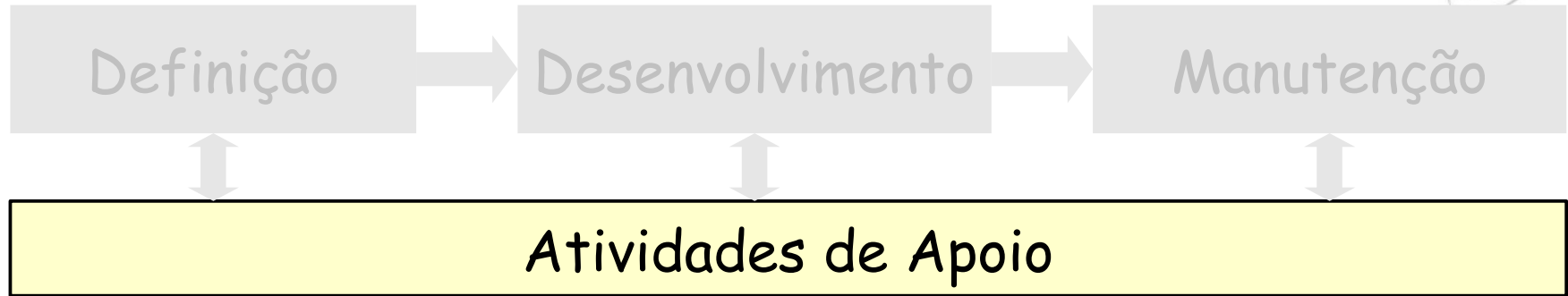
Atividades típicas:

Controle e Acompanhamento do Projeto de Software

Revisões com Técnicas Formais

Gestão de Configuração de Software

# Fases Genéricas dos Modelos de Processo de SOFTWARE



Atividades típicas:

Preparação e Produção de Documentos

Gestão de reutilização

Medições

Gestão de risco

# Próxima aula...

---

- Existem vários **modelos de processo de software** (ou paradigmas de engenharia de software).
- Cada um representa uma tentativa de colocar ordem em uma atividade inerentemente caótica.
- É escolhido com base:
  - Na natureza do projeto e da aplicação.
  - Nos métodos e ferramentas a serem utilizados.
  - Nos controles e produtos que precisam ser entregues.