



UNIVERSIDADE FEDERAL DO PIAUÍ – UFPI  
CAMPUS SENADOR HELVÍDIO NUNES DE BARROS - CSHNB  
CURSO DE BACHARELADO EM SISTEMAS DE INFORMAÇÃO



# **Engenharia de Software I**

## **Modelo de Processo de Software – Parte 2**

**Professora Pâmela Carvalho**

**13/12/22**

# *Modelos de Processo de Software*

- 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

## *Para escolha de um Modelo de Processo de Software:*

- **Natureza** do projeto e da aplicação
- **Métodos** e **ferramentas** a serem usados
- **Controles** e **produtos** que precisam ser entregues

# MODELOS DE PROCESSO DE SOFTWARE

- O Modelo Sequencial Linear (Modelo Cascata) **OK**
- O Modelo de Prototipação **OK**
- O Modelo RAD (*Rapid Application Development*) **OK**
- Modelos Evolutivos de Processo de Software:
  - O Modelo Incremental **OK**
  - O Modelo Espiral **OK**
  - O Modelo de Montagem de Componentes
  - O Modelo de Desenvolvimento Concorrente
- Modelo de Métodos Formais
- Técnicas de Quarta Geração

# Antes de estudarmos os outros modelos...

## Comparação

### **Modelo Incremental**

- Atividades fixas do modelo em cascata são usadas em cada incremento.
- Objetiva a elaboração de um produto operacional a cada incremento, que pode ser testado.

### **Modelo Espiral**

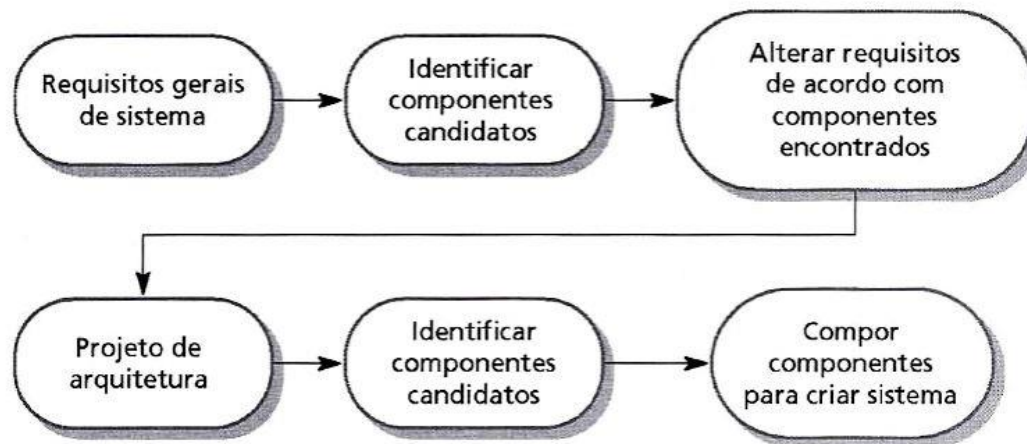
- As atividades não são fixas, cada “loop” se concentra mais em uma determinada atividade.
- A análise de riscos é uma atividade essencial no modelo.

# **Modelos Evolutivos de Processo**

**Modelo de Montagem de Componentes**

# Modelo de Montagem de Componentes

- Utiliza tecnologias **orientadas a objeto**.
- Quando projetadas e implementadas apropriadamente as **classes** orientadas a objeto são **reutilizáveis** em diferentes aplicações e arquiteturas de sistema.
- O modelo de montagem de componentes incorpora muitas das características do **modelo espiral**.





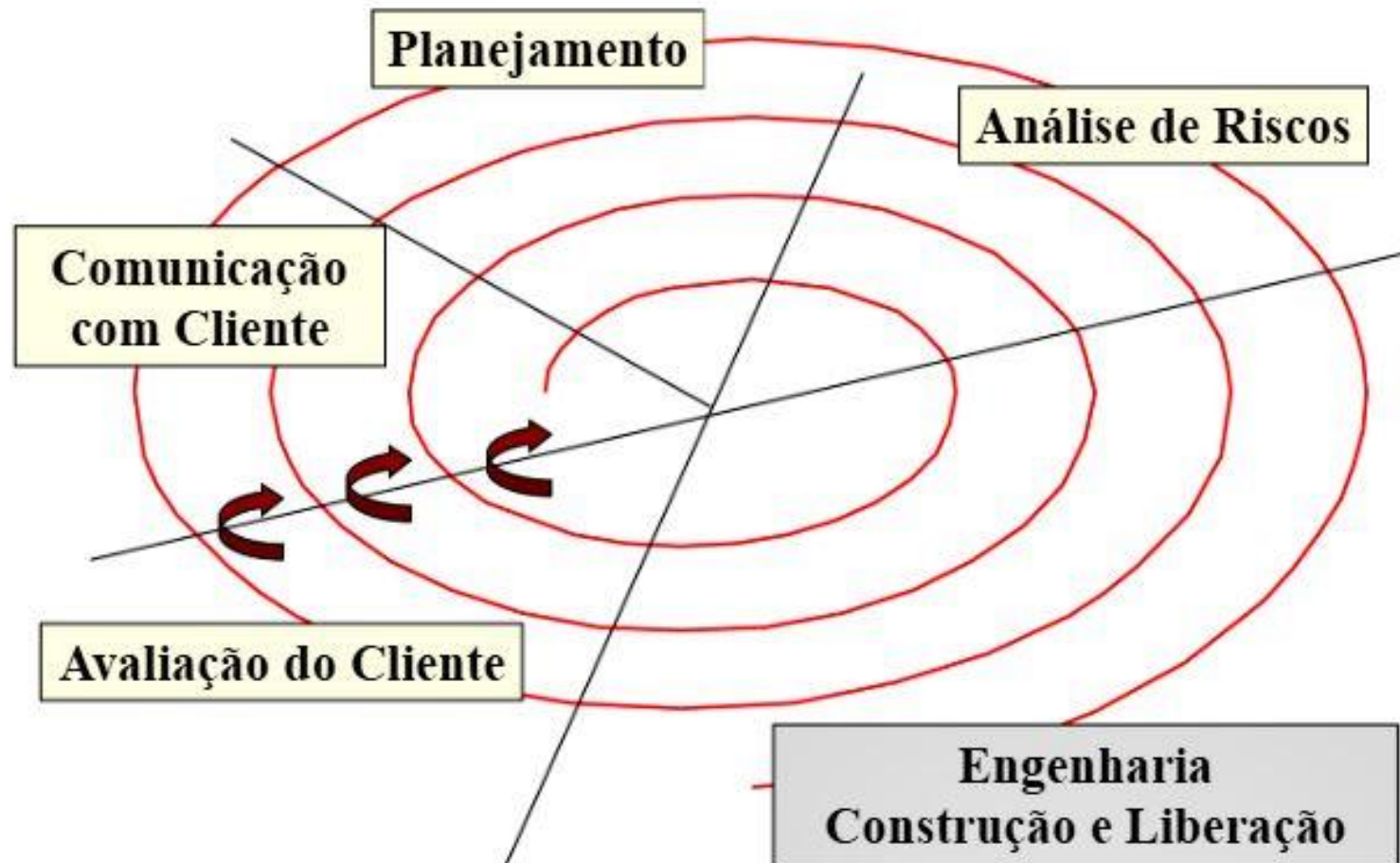
# Desenvolvimento Baseado em Componentes

- Compõe aplicações a partir de **componentes** de software previamente preparados.
- Segue os seguintes passos implantados com uma abordagem evolucionária:
  1. Pesquisa e avaliação de componentes disponíveis para o domínio em questão.
  2. Considerações sobre a integração de componentes.
  3. Projeto de arquitetura de software.
  4. Integração dos componentes à arquitetura.
  5. Testes para garantir a funcionalidade adequada.



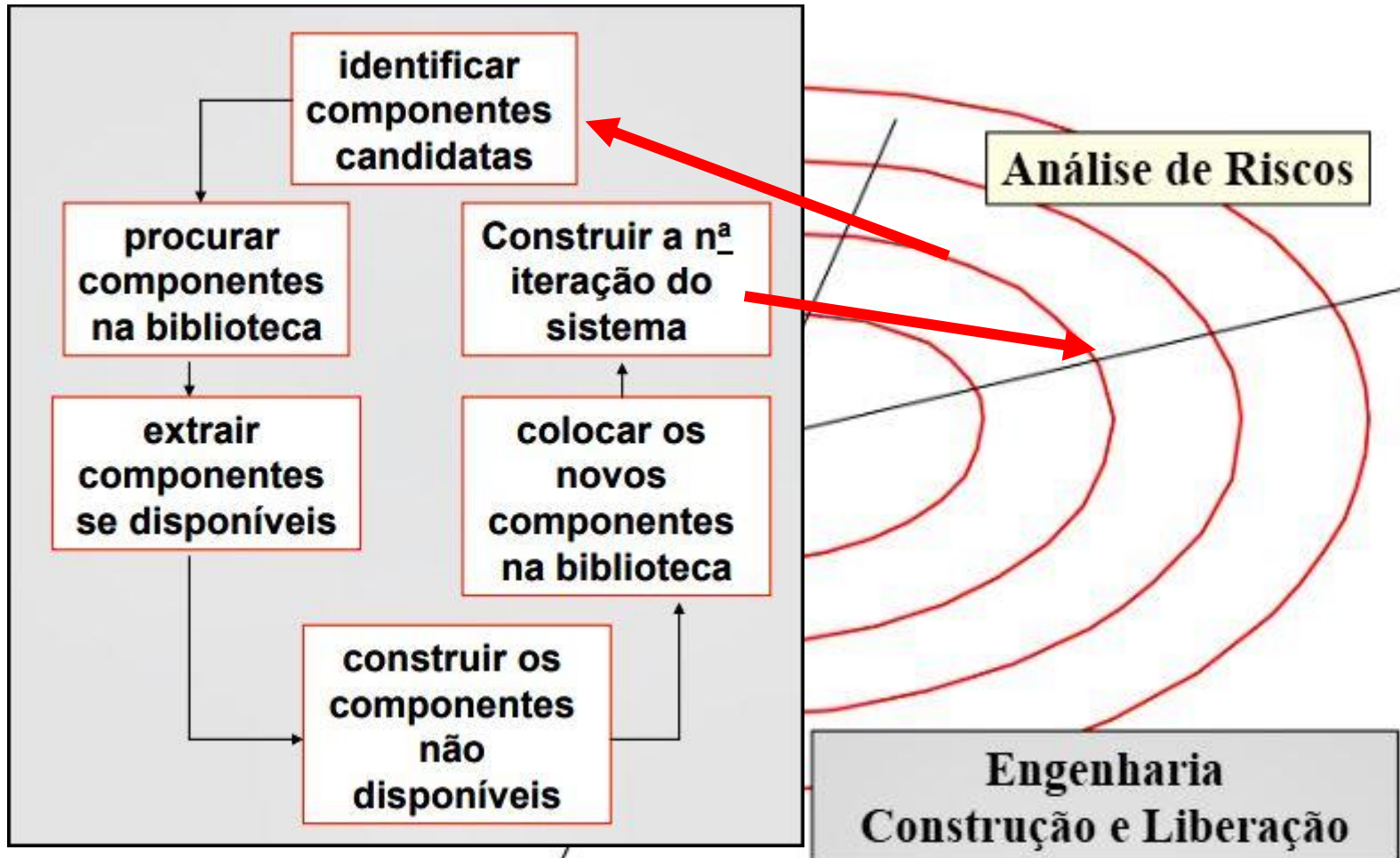
# Modelo de Montagem de Componentes

O Modelo de Montagem de Componentes



# Modelo de Montagem de Componentes

O Modelo de Montagem de Componentes



# Vantagens do desenvolvimento baseado em componentes

- Leva ao **reuso** de software, que segundo estudos tem como consequências:
  - Redução significativa do prazo de desenvolvimento.
  - Redução significativa no custo do projeto.
  - Aumento do índice de produtividade.
- Em que situações o desenvolvimento baseado em componentes **não** é adequado?
  - Aquelas em que não existam componentes padrão disponíveis ou em que não se queira pagar pelos componentes.

# **Modelos Evolutivos de Processo**

**Modelo de Desenvolvimento Concorrente**



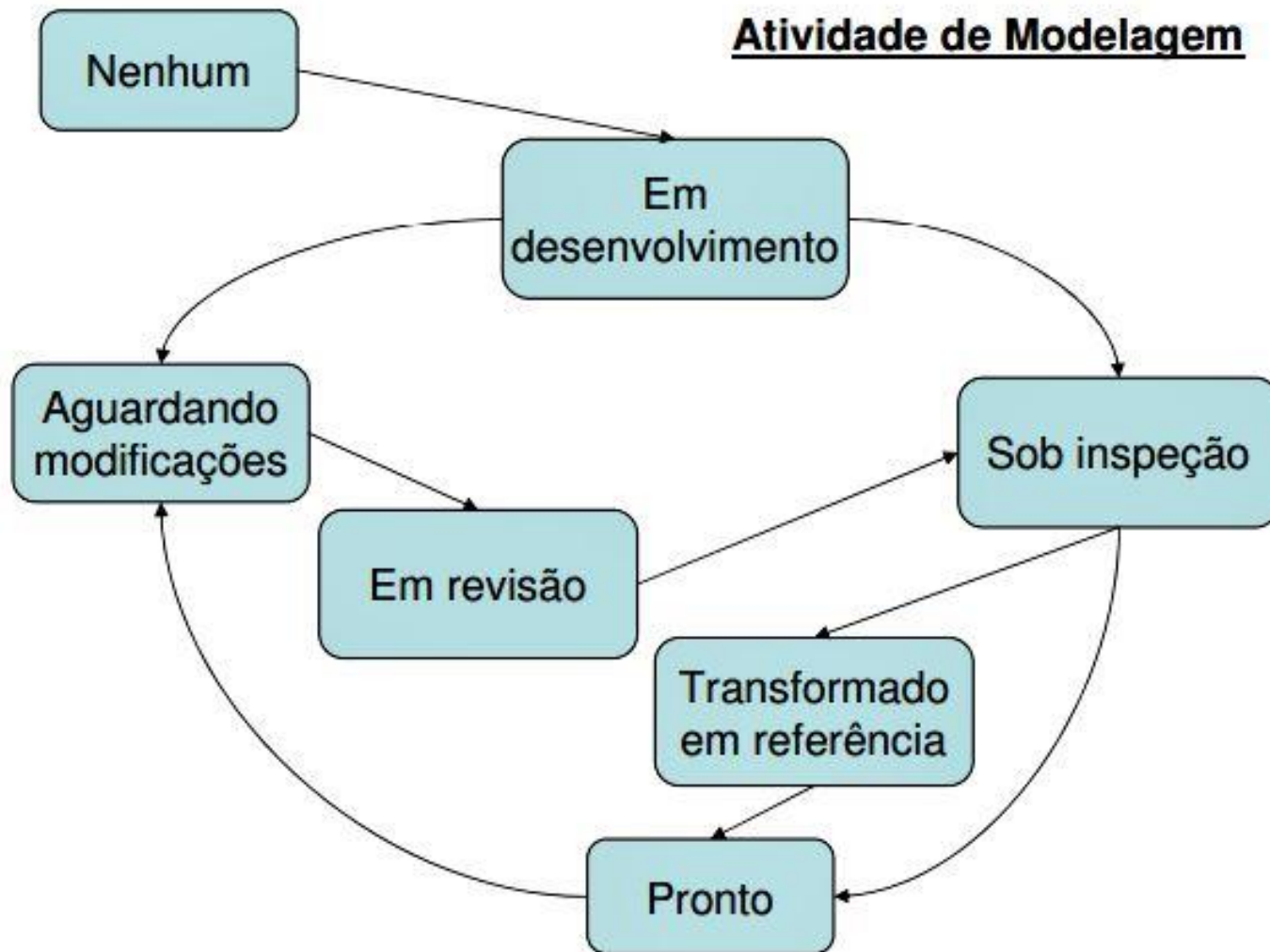
# Modelo de Desenvolvimento Concorrente

- Todas as atividades ocorrem em **paralelo** mas estão em diferentes **estados**.
- O modelo define uma série de eventos que vão disparar **transições** de estado para estado, para cada uma das atividades.
- Em vez de usar uma seqüência como o modelo cascata, ele define uma **rede** de atividades.
- Pode ser aplicado a todo tipo de desenvolvimento de software e fornece uma visão exata de como está o estado do projeto.



# Desenvolvimento Concorrente

## Atividade de Modelagem



# Desenvolvimento Concorrente

- Exemplo: Começo de projeto
  - A atividade de **comunicação** completou sua primeira iteração e está no estado **aguardando modificações**.
  - A atividade de **modelagem** passa do estado **nenhum** para o estado **em desenvolvimento**.
  - Se o cliente requiere mudança nos requisitos, a modelagem passa de **em desenvolvimento** para **aguardando modificações** e a comunicação passa de aguardando modificações para **em revisão**.

# **Modelos de Métodos Formais**

# Modelo de Métodos Formais

- Métodos formais permitem ao engenheiro de software especificar, desenvolver e verificar um sistema aplicando uma **rigorosa notação matemática**.
  - Uma variante chamada **engenharia de software sala limpa** é aplicada por algumas organizações.



# Vantagens dos métodos formais

1. **Elimina** muitos problemas encontrados nos outros modelos:
  - ambigüidade
  - incompletitude
  - inconsistência
2. Servem de base para a **verificação** de programas, oferecendo a promessa de um software livre de defeitos.
3. Adequado para softwares **críticos** (por exemplo, de aeronaves e dispositivos médicos).



## **Desvantagens dos métodos formais**

1. O desenvolvimento de modelos formais é atualmente muito **lento** e dispendioso.
2. Exige **treinamento** extensivo para dar aos desenvolvedores o preparo necessário.
3. É difícil usar os modelos como um mecanismo de **comunicação** com a maioria dos clientes.

# **Técnicas de Quarta Geração**

# Técnicas de Quarta Geração

- Concentra-se na capacidade de se especificar o software a uma máquina em um nível que esteja próximo à **linguagem natural**.
- Engloba um conjunto de ferramentas de software que possibilitam que:

O sistema seja especificado em uma **linguagem de alto nível** e o código fonte seja gerado automaticamente a partir dessas especificações.

## Linguagem de programação de quarta geração

As linguagens de programação de quarta geração, ou 4GL em sua abreviatura de origem inglesa, são linguagens de programação de alto-nível com objetivos específicos, como o desenvolvimento de softwares comerciais de negócios. Elas permitem ao programador especificar o que deve ser feito visando um resultado imediato.

# Técnicas de Quarta Geração

- O ambiente de desenvolvimento de software que sustenta o ciclo de vida de geração inclui as ferramentas:
  - Linguagens não procedimentais para consulta de banco de dados.
  - Geração de relatórios
  - Manipulação de dados
  - Interação e definição de telas
  - Geração de códigos
  - Capacidade gráfica de alto nível
  - Capacidade de planilhas eletrônicas

# *Técnicas de 4ª Geração*





## **OBTENÇÃO DOS REQUISITOS:**

- o cliente descreve os requisitos os quais são traduzidos para um protótipo operacional
  - o cliente pode estar inseguro quanto aos requisitos
  - o cliente pode ser incapaz de especificar as informações de um modo que uma ferramenta 4GL possa consumir
  - as 4GLs atuais não são sofisticadas suficientemente para acomodar a verdadeira "linguagem natural"

**Extra**

**Processo Unificado (PU)**

**ou ainda RUP (Rational Unified Process)**

# O Processo Unificado

- É uma tentativa de **unir** os melhores recursos e características dos modelos convencionais.
- Reconhece a importância da comunicação com o cliente e dos **casos de uso** para descrever a visão do cliente
- Utiliza a **UML** como a notação para modelagem e análise de projeto.
- Sugere um fluxo de processo que é **iterativo** e **incremental**.
- Também conhecido como RUP (de Rational Unified Process) – a Rational construiu ferramentas de apoio ao processo unificado.



# Processo Unificado

- É um processo incremental.
  - Enquanto acontecem as fases de construção, transição e produção, já pode ser iniciado o incremento seguinte.
- Um fluxo de trabalho de engenharia de software é distribuído ao longo de todas as fases do PU.
  - Identifica as tarefas exigidas para realizar uma ação importante de engenharia de software.

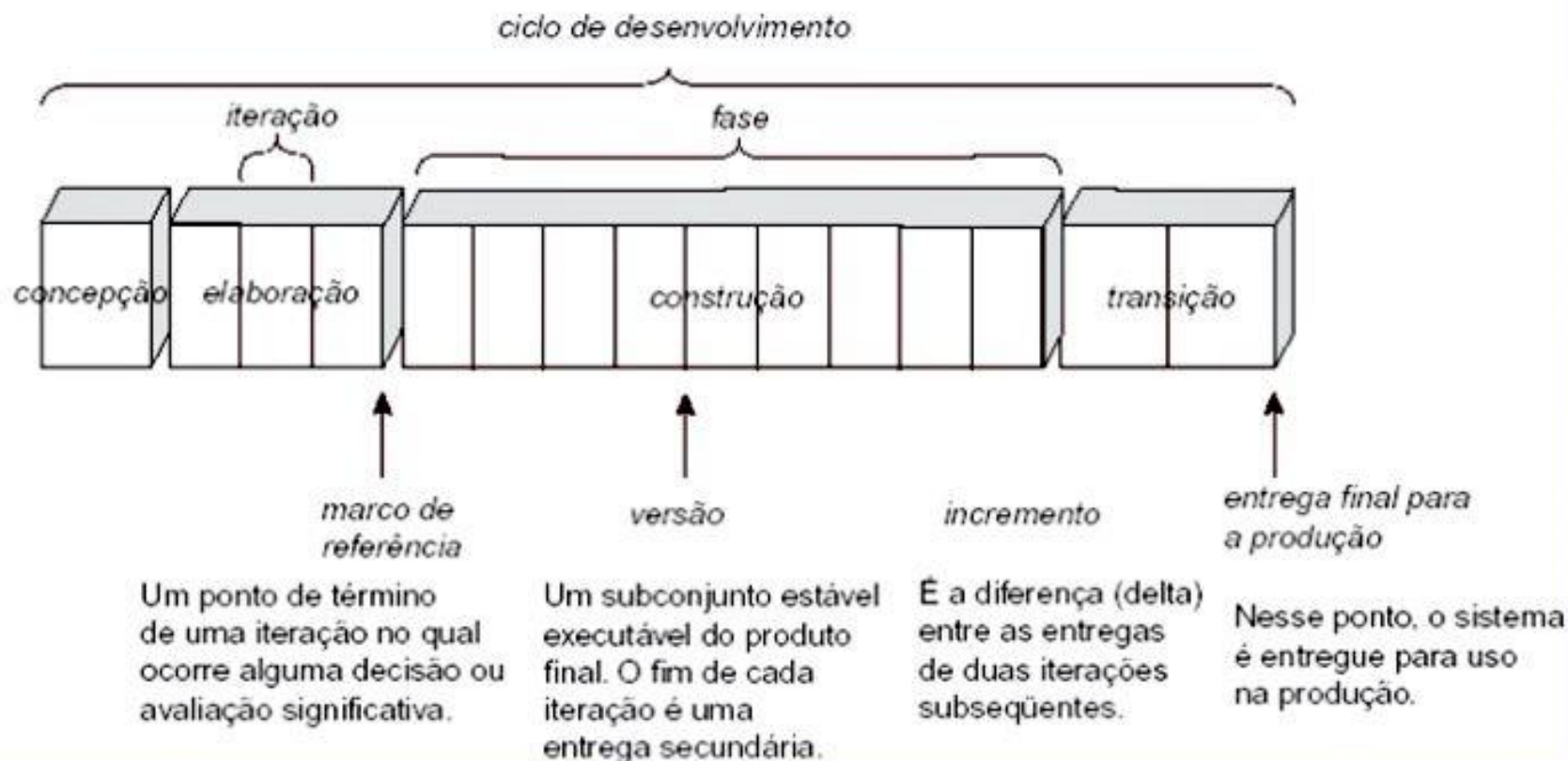
# Baseado em Casos de Uso

Um caso de uso é uma seqüência de ações, executadas por um ou mais atores e pelo próprio sistema, que produz um ou mais resultados de valor para um ou mais atores.

O PU é dirigido por casos de uso, pois os utiliza para dirigir todo o trabalho de desenvolvimento, desde a captação inicial e negociação dos requisitos até a aceitação do código (testes).



# As Fases do PU



# Vídeo

RUP – Processo Unificado da Rational



# **Estudem sobre Modelos de Processo**

Amanhã será repassado no SIGAA uma atividade valendo 1 ponto para 1ª nota.

O horário da aula será para resolução da atividade.

**Obs: Poderá ser feita em dupla**

**Até a próxima aula...**