



# Engenharia de Software I

---

Prof. Agnaldo Cieslak, MsC



# ENGENHARIA DE SOFTWARE I

Carga Horária: 40 horas

## **Competência:**

Analisar e especificar projetos de software.

## **Indicadores:**

- Desenha o projeto de sistemas de acordo com as fases do ciclo de vida de desenvolvimento de software.
- Elicita requisitos do software, conforme as necessidades do projeto.
- Especifica os requisitos funcionais, não-funcionais e requisitos inversos, de acordo com o resultado do processo de elicitação de requisitos.
- Constrói o modelo de casos de uso do sistema de acordo com a especificação de requisitos do sistema.
- Elabora User Stories de acordo com o resultado do processo de elicitação de requisitos.
- Elabora e documenta diagrama de casos de uso.
- Elabora diagrama de atividades dos casos de uso.

# ENGENHARIA DE SOFTWARE I

## Ementa

---

Ciclo de vida de desenvolvimento de software;

---

Processo de desenvolvimento de software;

---

Elicitação de requisitos;

---

Requisitos Funcionais, Não-Funcionais e Inversos;

---

Análise de requisitos (baseado em metodologias ágeis);

---

Definição de User Stories e Story points;

---

UML: modelo de Casos de Uso (diagrama e especificação);

---

UML: Diagrama de Atividades.

# Aula 1

---

## Ciclo de vida de um projeto

# Ciclo de Vida de Desenvolvimento

---

O ciclo de vida de um projeto consiste em fases e atividades que são definidas:

- de acordo com o escopo;
- recursos;
- e a natureza (características) do projeto;

visando oferecer maior controle gerencial.

# Ciclo de Vida de Desenvolvimento

---

- Cada fase do ciclo de vida do projeto, gera produtos de trabalho para as fases posteriores.
- Organização em fases permite o planejamento do projeto, e marcos para seu controle.
- Marcos previamente definidos ao se realizar o planejamento.
- É possível: técnicas de fast-tracking e collision, sem afetar os ciclos de vida.

# Ciclo de Vida de Desenvolvimento

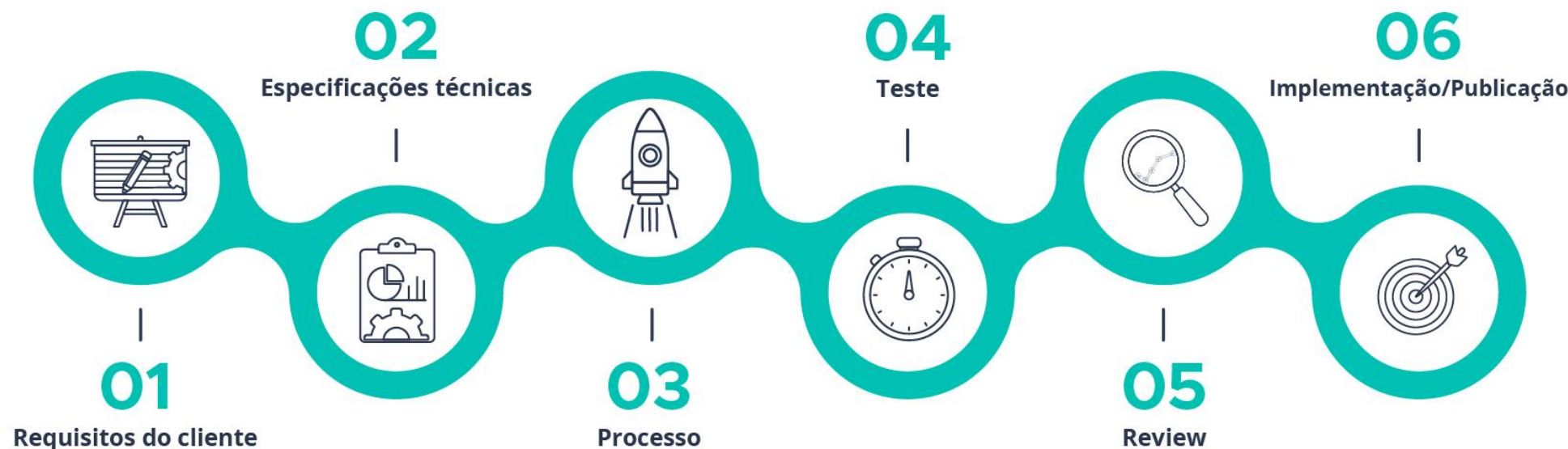
---

- Fast-tracking: atividades iniciadas antes de outra, mesmo tendo dependências, desde que o risco seja tratado.
- Collision: permite que mais pessoas sejam alocadas em uma tarefa para garantir a data de entrega, desde que o risco seja tratado.

# Criando um Software

## Quais etapas Envolvidas na criação de um software?

- Os **clientes** e **outros envolvidos** expressam a **necessidade** (01) pelo **software** de computador, os **engenheiros constroem o produto** de software (02, 03, 04 e 05) e os **usuários o utilizam** (06) para **resolver** um **problema específico** ou para tratar de uma **necessidade específica**





# Ciclo de Vida de Desenvolvimento

---

## [AWS.AMAZON] – SDLC (System Development Life Cycle)

O ciclo de vida de desenvolvimento de software (SDLC) é o processo econômico e rápido que as equipes de desenvolvimento usam para projetar e criar software de alta qualidade. O objetivo do SDLC é minimizar os riscos do projeto por meio do planejamento antecipado, para que o software atenda às expectativas do cliente durante e depois da produção. Essa metodologia descreve uma série de etapas que dividem o processo de desenvolvimento de software em tarefas que você pode atribuir, concluir e avaliar.

# Ciclo de Vida de Desenvolvimento

---

## Planejamento

- tarefas de análise do custo-benefício, a programação, a estimativa e a alocação de recursos.
- a equipe coleta requisitos de várias partes envolvidas: clientes, especialistas internos e externos e gerentes
- criar um documento de especificação de requisitos do software.
  - define as expectativas e as metas comuns que ajudarão no planejamento do projeto.
  - estima os custos, cria uma programação e tem um plano detalhado para atingir suas metas.

# Ciclo de Vida de Desenvolvimento

---

## Projeto

- os times analisam os requisitos e identificam as melhores soluções para criar o software.
- Por exemplo: considerar a integração de módulos pré-existent, fazer escolhas tecnológicas e identificar ferramentas de desenvolvimento.
- Analisar qual a melhor forma de integrar o novo software às infraestruturas de TI existentes que a organização já tiver.

# Ciclo de Vida de Desenvolvimento

---

## Implementação

- time de desenvolvimento codifica o produto.
- analisam requisitos para identificar as tarefas de codificação menores realizadas diariamente para alcançar o resultado final.

# Ciclo de Vida de Desenvolvimento

---

## Teste

- Time de desenvolvimento combina testes manuais e automatizados para identificar bugs no software.
- A análise de qualidade inclui testar o software para identificar erros e verificar se ele atende aos requisitos do cliente.
- Como muitas equipes testam o código que desenvolvem, a fase de teste pode ocorrer em paralelo à fase de desenvolvimento.

# Ciclo de Vida de Desenvolvimento

---

## Implantação

- No desenvolver do sw, cria-se e testa-se o código em uma cópia do software diferente da usada pelo usuário.
- O software do cliente é considerado em *produção*, e outras cópias são o *ambiente de compilação* ou ambiente de teste.
- Ambientes de compilação e produção separados → uso do software, mesmo quando alterado ou atualizado.
- Inclui mover a cópia mais recente para o ambiente de produção, como empacotamento, configuração de ambientes e instalação.

# Ciclo de Vida de Desenvolvimento

---

## Manutenção

- Na fase de manutenção: a equipe corrige bugs, soluciona problemas do cliente e gerencia as alterações do software.
- a equipe monitora a performance geral do sistema, a segurança e a experiência do usuário para identificar novas formas de melhorar o software existente.

# Ciclo de Vida de Desenvolvimento

---

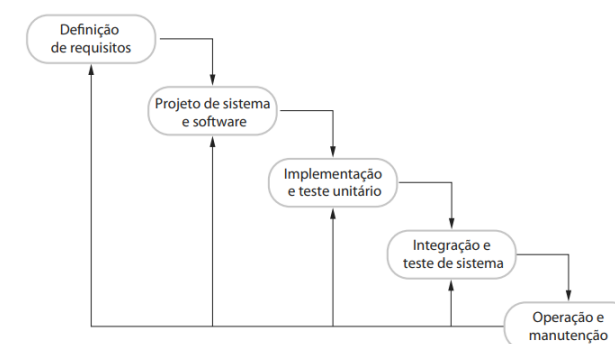
- Tipos de ciclos:
  - **Sequencial (ou Cascata)**
    - as fases são executadas em uma sequência ordenada.
    - a passagem para determinada atividade exige a finalização da atividade imediatamente anterior.
    - pontos fortes:
      - ênfase na abordagem disciplinada;
      - recomendação que todos os produtos de cada fase sejam formalmente revisados.



# Ciclo de Vida de Desenvolvimento

- Fases do modelo sequencial:
  - Iniciação (Não iterativo).
  - Planejamento (Não iterativo).
  - Análise dos Requisitos (Não iterativo).
  - Encerramento do Planejamento e Requisitos (Não iterativo).
  - Desenvolvimento (Não iterativo).
  - Entrega (Não iterativo)..

O modelo em cascata



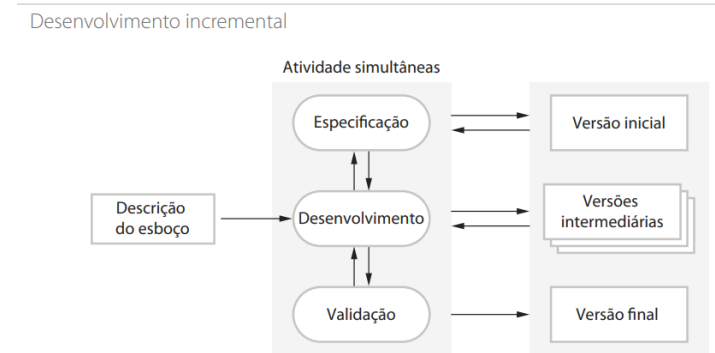
# Ciclo de Vida de Desenvolvimento

---

- **Iterativo (ou Incremental)**
  - Construção em pequenos passos (iteração) que evolui a para o produto final.
  - Organização em fases com objetivos distintos.
    - Dentro da fase faz-se atividades em ordem sequencial
    - Em cada fase ocorre múltiplas iterações
      - Pode variar com requisitos especiais
    - Iniciação e planejamento ocorre uma única iteração

# Ciclo de Vida de Desenvolvimento

- Fases do modelo iterativo:
  - Iniciação (Não iterativo).
  - Planejamento (iterativo).
  - Análise dos Requisitos (iterativo).
  - Encerramento do Planejamento e Requisitos (iterativo).
  - Desenvolvimento (iterativo).
  - Entrega (iterativo)..



# Ciclo de Vida de Desenvolvimento

---

- **RAD (ou Rapid Application Development)**
  - Adaptação em alta velocidade do modelo Cascata.
- Características deste ciclo de vida:
  - Sistema plenamente funcional entre 60 e 90 dias.
  - Incremental (ciclos curtos).
  - Baseado em componentes.
  - Modularidade (equipes distintas).
  - Requisitos precisam estar bem definidos e o escopo deve ser restrito.

# Ciclo de Vida de Desenvolvimento

---

- **As principais desvantagens:**
  - Requer recursos humanos suficientes para criar um número adequado de equipes.
  - Requer um comprometimento entre desenvolvedores e clientes:
    - atividades possam ser realizadas rapidamente;
    - sistema seja concluído em um tempo abreviado.
    - comprometimento abandonado → o projeto falhará.
    -
  - Não é apropriado quando os riscos são grandes.

# Ciclo de Vida de Desenvolvimento

- **Evolutivo**
  - SW evolui com o tempo e uso das versões iniciais.
  - É também iterativo.
  - Requisitos parcialmente definidos e depois refinados.
    - requisitos conhecidos → versão parcial
    - Conhecimento adquirido → evolução do produto



# Ciclo de Vida de Desenvolvimento

---

- **As principais desvantagens:**
  - Cliente não sabe que o software que ele vê não considerou, durante o desenvolvimento, a qualidade global e a manutenibilidade a longo prazo. Força a utilização do protótipo como produto final.
  - Desenvolvedor frequentemente faz uma implementação comprometida, com o objetivo de produzir rapidamente um protótipo. Depois de um tempo ele se familiariza com essas escolhas.

# Ciclo de Vida de Desenvolvimento

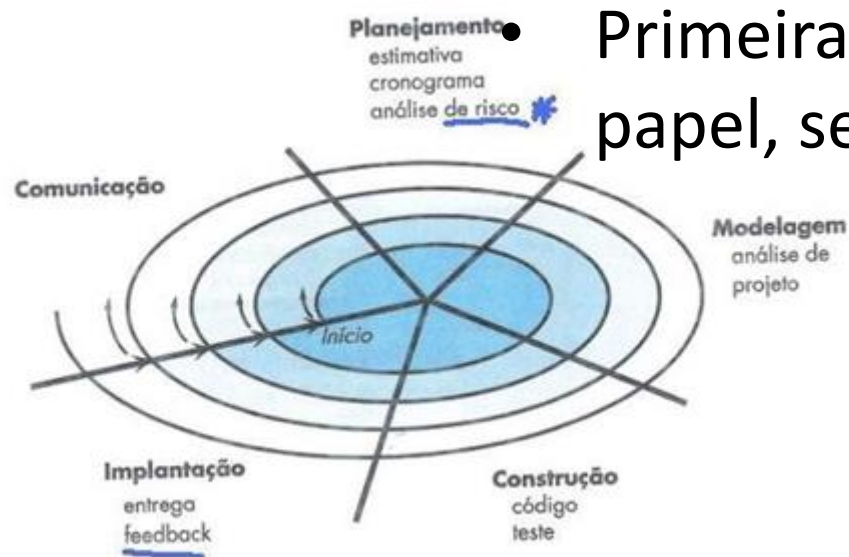
---

- **Ainda assim:**
  - este modelo de prototipação é um meio de desenvolvimento adequado.
  - definindo as regras deste jogo no começo ->sem surpresas
  - O cliente e o desenvolvedor devem ambos concordar que o protótipo → mecanismo a fim de definir os requisitos



# Ciclo de Vida de Desenvolvimento

- Espiral
- Combina a iteração da Prototipagem com aspectos do Cascata.
- Software desenvolvido em versões evolucionárias.
- Primeiras versões (ciclos) podem ser modelo de papel, seguindo um protótipo etc.



# Ciclo de Vida de Desenvolvimento

---

- **Espiral (cont...)**
- Modelo guiado por riscos.
- Abordagem cíclica.
- Aumentar incrementalmente o grau de definição e implementação de um sistema, enquanto seu grau de risco diminui.
- Marcos de ancoragem: estabelecer quando um ciclo é completado em torno da espiral e fornecem marcos de decisão antes do projeto de software prosseguir. Ex.: objetivos e arquitetura do ciclo de vida, capacidade inicial de operação.

# Ciclo de Vida de Desenvolvimento

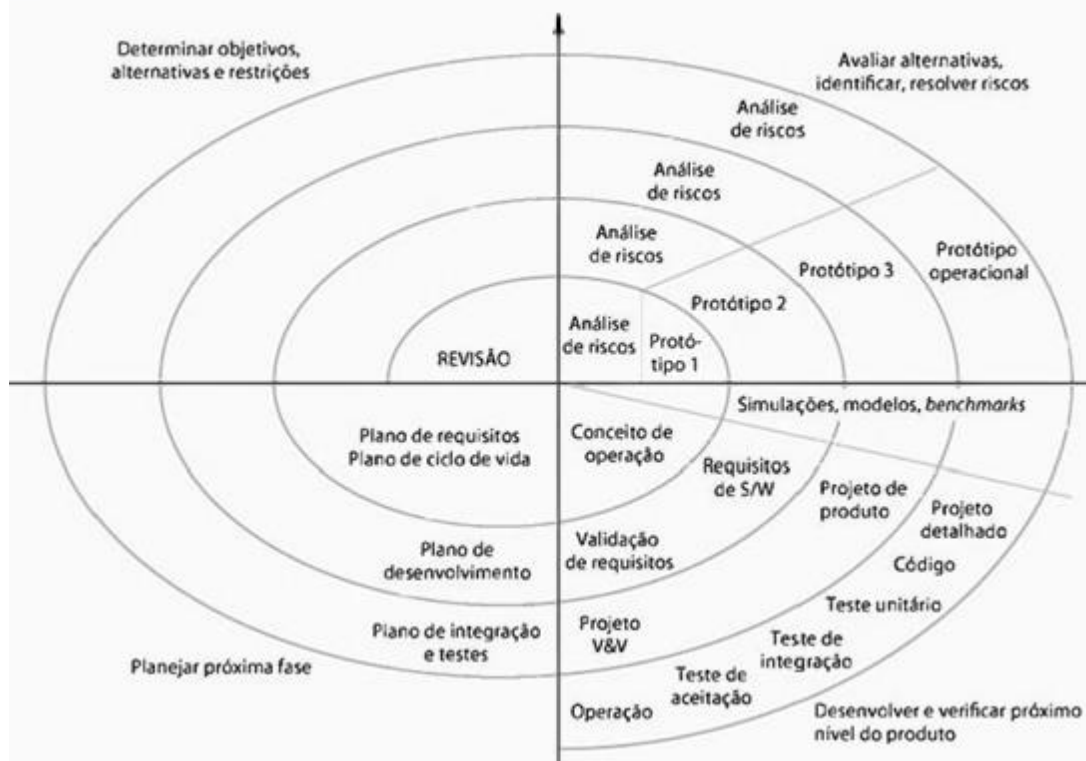
---

- **Espiral (cont...)**
- Garantir o comprometimento dos interessados.
- Usa uma abordagem que capacita o desenvolvedor e o cliente a entender e reagir aos riscos em cada etapa evolutiva.
- Pode ser difícil convencer os clientes que uma abordagem “evolutiva” é controlável.
- Exige considerável experiência na determinação de riscos e depende dessa experiência para ter sucesso.

# Ciclo de Vida de Desenvolvimento

- Espiral

11 Modelo em espiral de processo de software de Boehm (©IEEE 1988)



# Ciclo de Vida de Desenvolvimento

---

## RUP (Rational Unified Process)

- Processo de desenvolvimento de SW criado e mantido pela Rational/IBM.
- Baseado no ciclo de vida em espiral (refinamentos sucessivos).
- Baseada em disciplinas:
  - Tarefas.
  - Responsabilidades.
- Guia para a utilização de UML:
  - Desenvolvimento dirigido por casos de uso.

# Ciclo de Vida de Desenvolvimento

---

## RUP (Rational Unified Process)

- Processo configurável (customizável).
- Avaliação contínua dos riscos do projeto.
- Geração de produtos interdependentes em todas as interações.
- Desenvolvimento iterativo-incremental:
  - Iteração: atividades de modelagem de negócios, requisitos, análise e projeto, implementação, teste e implantação, em várias proporções, dependendo do local em que ela está localizada no ciclo de desenvolvimento.

# Ciclo de Vida de Desenvolvimento

---

## RUP (Rational Unified Process)

- Disciplina:

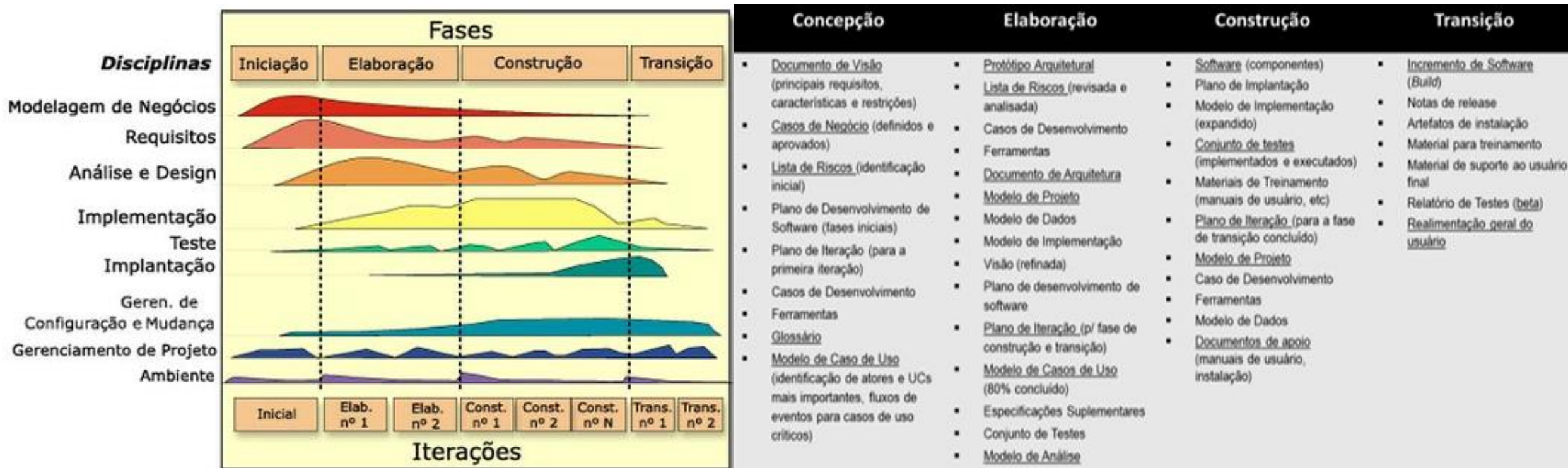
- Conjunto de atividades relacionadas a uma 'área de interesse' importante em todo o projeto.

- Papéis:

- O comportamento e as responsabilidades de um indivíduo ou de um conjunto de indivíduos.

# Ciclo de Vida de Desenvolvimento

## RUP (Rational Unified Process)





# Atividade em Sala

---

- Os grupos do projeto se reúnem para discutir, avaliar e elaborar a solução da questão abaixo:

Justificando sua resposta com base no tipo de sistema a ser desenvolvido, sugira o modelo genérico de processo de software mais adequado para ser usado como base para a gerência do desenvolvimento dos sistemas a seguir:

Um sistema para controlar o antibloqueio de frenagem de um carro.

Um sistema de realidade virtual para dar apoio à manutenção de software.

Um sistema de contabilidade para uma universidade, que substitua um sistema já existente.

Um sistema interativo de planejamento de viagens que ajude os usuários a planejar viagens com menor impacto ambiental.

# Aula 2

---

## Metodologias Ágeis