

# Modelagem de Software



---

Prof. Dr. Ronaldo Castro de Oliveira

[ronaldo.co@ufu.br](mailto:ronaldo.co@ufu.br)

[www.facom.ufu.br/~ronaldooliveira](http://www.facom.ufu.br/~ronaldooliveira)

FACOM - 2018



---

# Paradigmas e Processo de Software

# Engenharia de Software: Abrangência

---

- Engenharia de Software possui 3 elementos fundamentais:
  - métodos: “como fazer”
  - ferramentas: apoio automatizado aos métodos.
  - Procedimentos: elo de ligação entre os métodos e os procedimentos
- Existem diversos *Paradigmas de Engenharia de Software*:
  - abordagens que envolvem estes métodos, ferramentas e procedimentos

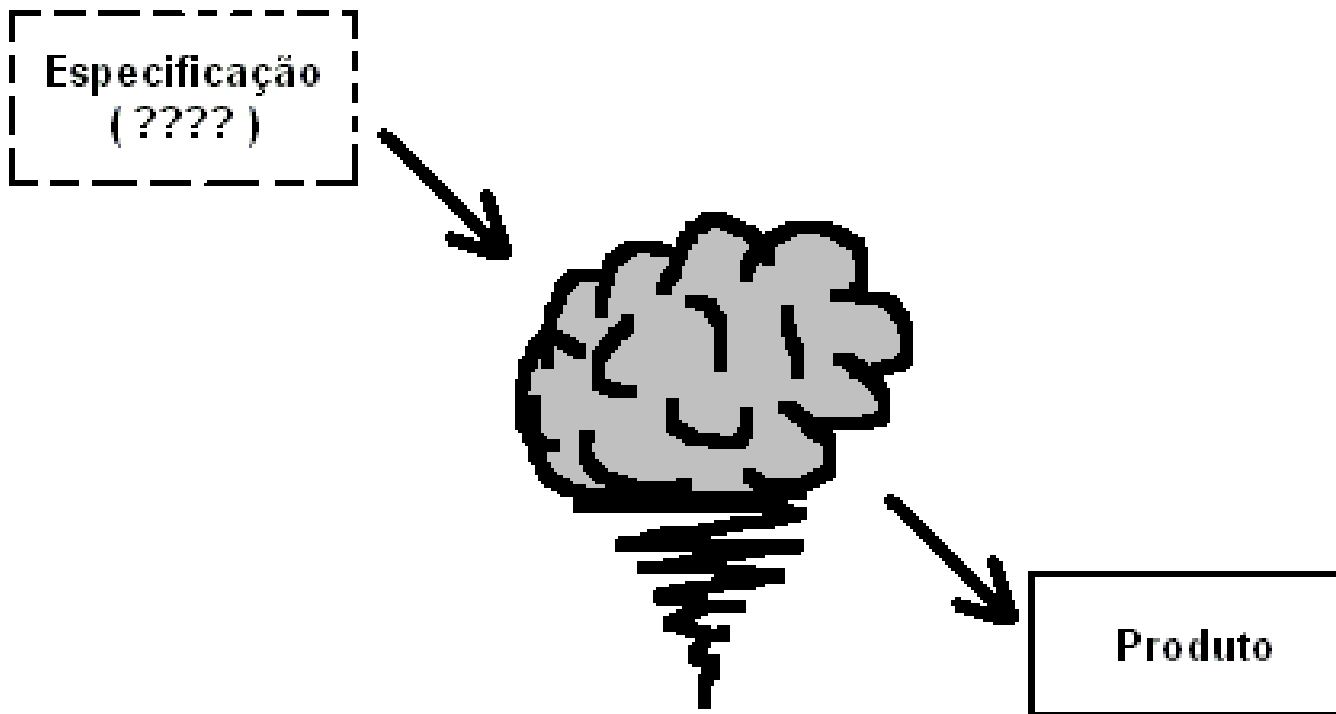
# Paradigmas da Engenharia de Software



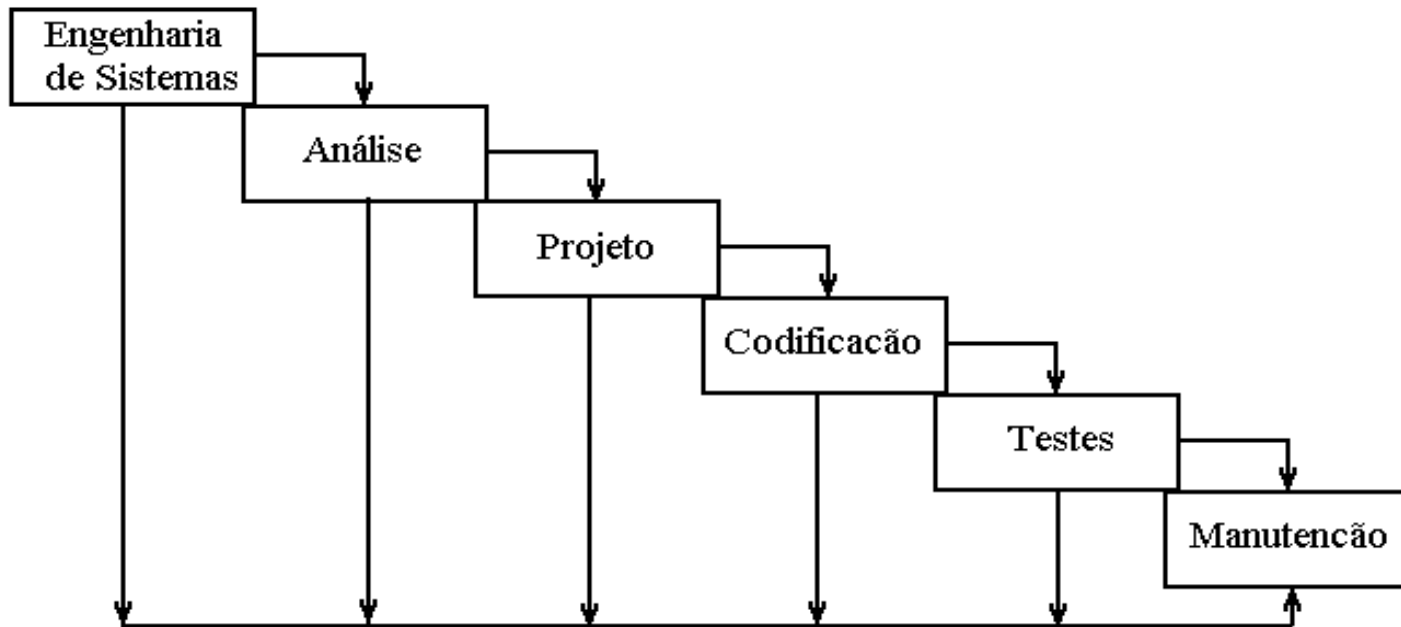
---

- Existem diversos paradigmas de software, dentre eles:
  - Desenvolvimento Caótico
  - Modelo Clássico
  - Modelo de Prototipação
  - Modelo Espiral
  - Técnicas de 4a Geração
  - Modelo de Entrega Evolutiva

# Ciclo de Vida Caótico



# Ciclo de Vida Clássico (I)





# Ciclo de Vida Clássico (II)

---

- Problemas:
  - projetos reais não seguem um fluxo seqüencial: dificuldade de acomodar mudanças depois de iniciado.
  - Dificuldade de declaração de todas as exigências pelo cliente.
  - Paciência!

# Modelo de Prototipação (I)





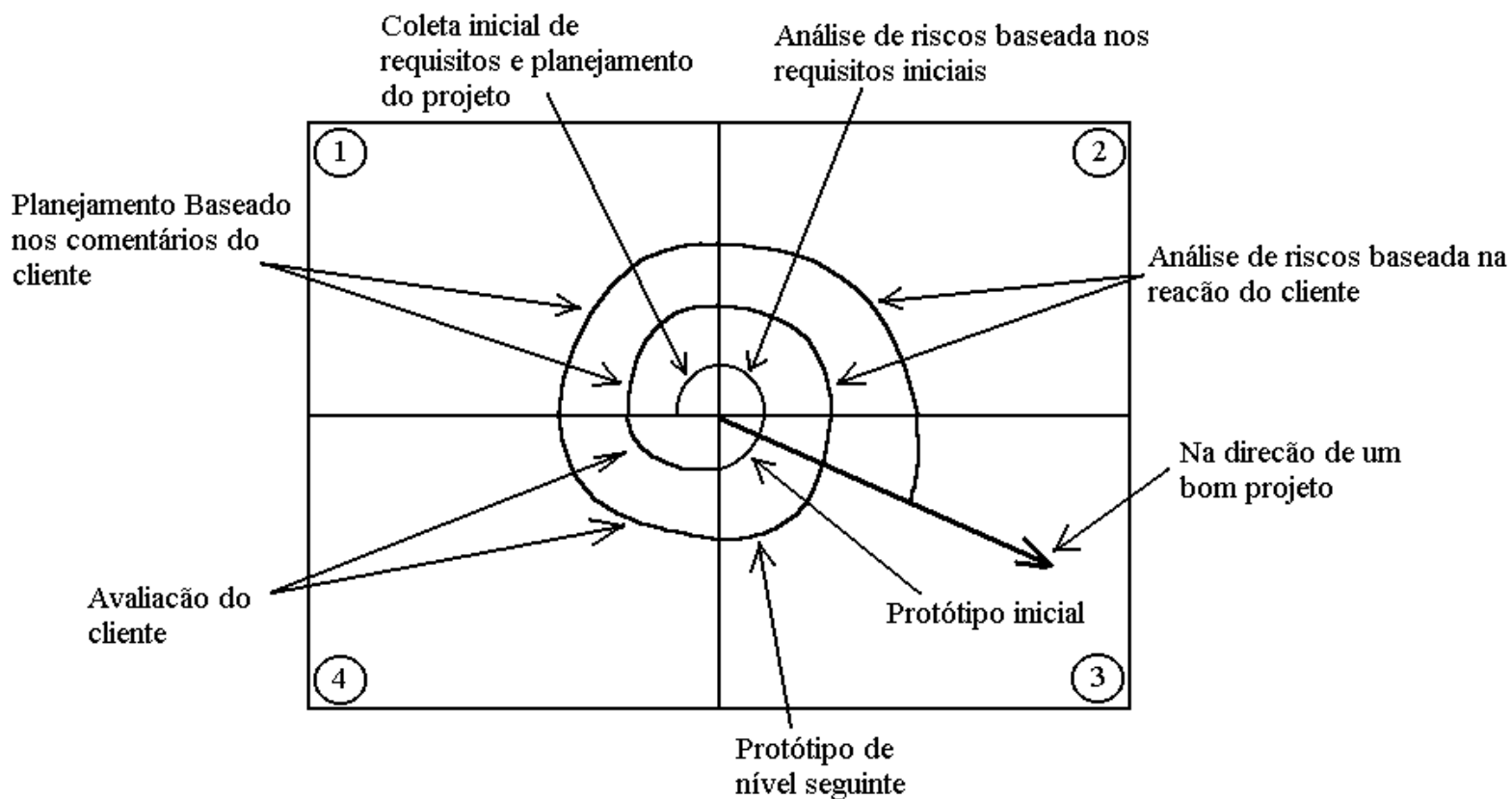


# Modelo de Prototipação (II)

---

- Problemas:
  - o cliente pega em mãos um produto inacabado e sem qualidade e exige ótimo funcionamento
  - devido a rapidez com que o protótipo é implementado o programador pode fazer concessões que futuramente podem ser esquecidas, piorando assim a qualidade do produto

# Modelo Espiral (I)



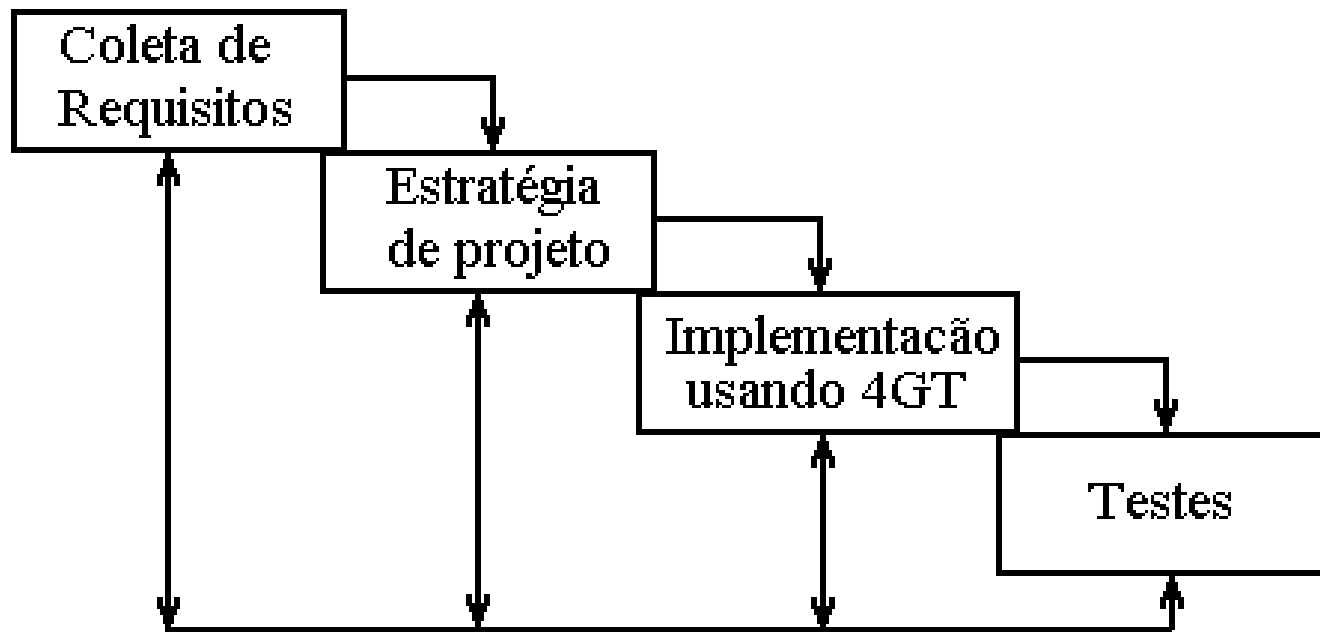


# Modelo Espiral (II)

---

- 1. Planejamento → determinação dos objetivos, alternativas e restrições;
- 2. Análise de Riscos → análise de alternativas e identificação e resolução dos riscos;
- 3. Engenharia → desenvolvimento do produto no “nível seguinte”;
- 4. Avaliação feita pelo cliente → avaliação dos resultados da engenharia do produto

# Técnicas de 4a Geração (I)





# Técnicas de 4a Geração (II)

---

- Abrange um amplo conjunto de ferramentas de software que têm uma coisa em comum: cada uma delas possibilita que o desenvolvedor de software especifique algumas características num software num nível elevado. A ferramenta gera então, automaticamente, o código fonte, tendo como base a especificação do desenvolvedor.



# Técnicas de 4a Geração (III)

---

- Inclui as seguintes ferramentas:
  - linguagens não procedimentais para consultas de bancos de dados;
  - geração de relatórios;
  - manipulação de dados;
  - interação e definição de todos;
  - geração de código;
  - capacidade gráfica de alto nível;
  - capacidade de planilhas eletrônicas.

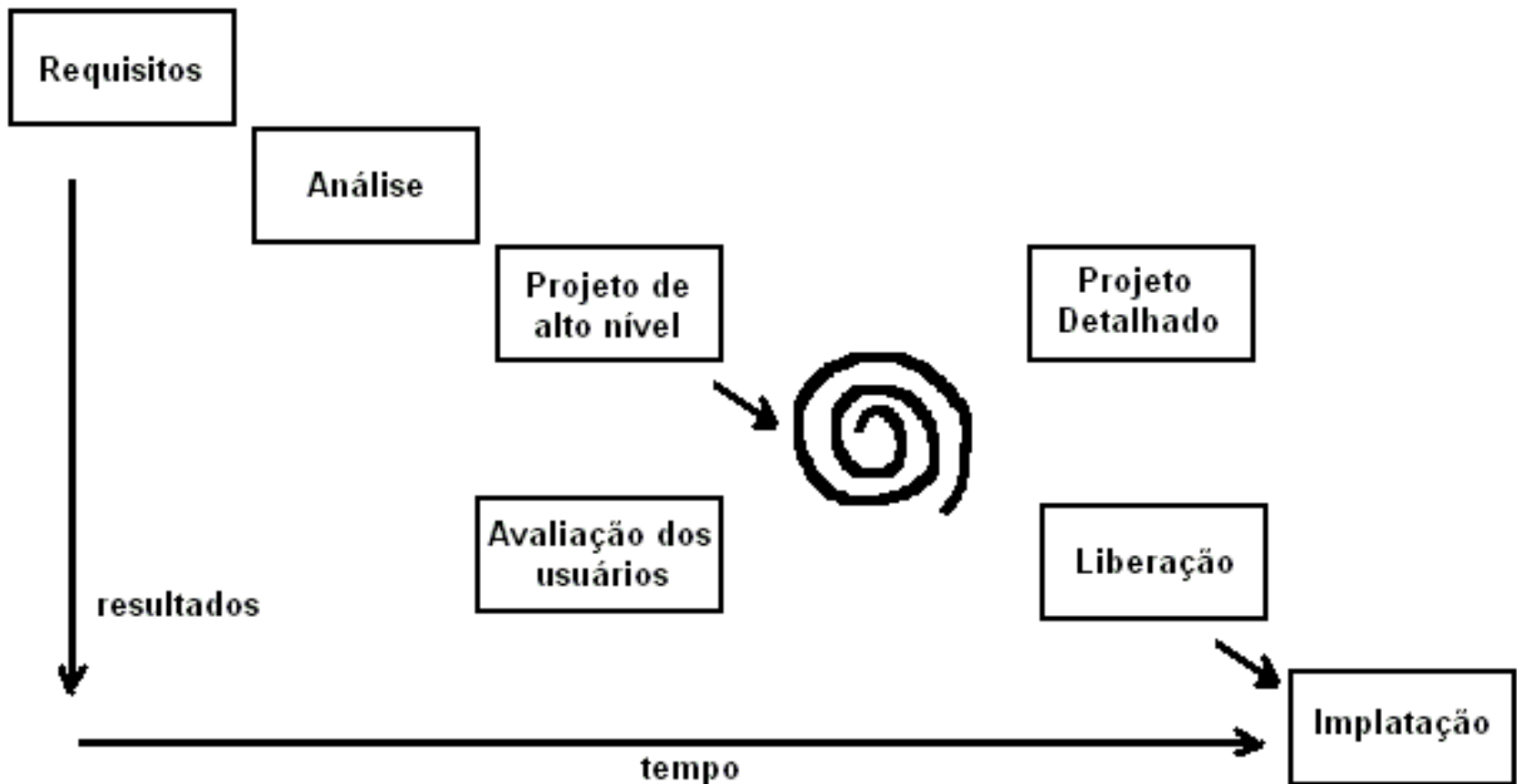


# Técnicas de 4a Geração (IV)

---

- Resumo da abordagem 4GT
  - Com raras exceções as 4GT limitam-se a aplicações comerciais, especificamente, análise de informações e relatórios ligados a grandes bancos de dados.
  - Os dados coletados com o uso de 4GT parecem indicar que o tempo exigido para se produzir software é reduzido para aplicações pequenas e médias.
  - O uso das 4GT para grandes sistemas exige tanto ou mais análise, planejamento e testes para conseguir as significativas economias de tempo que podem ser obtidas por meio da eliminação da atividade de codificação.

# Entrega Evolutiva (I)







# Entrega Evolutiva (II)

---

- Combinação dos modelos de Cascata e Prototipagem em Espiral.
- Permite, em pontos bem definidos, que os usuários possam avaliar partes do produto e fornecer realimentação quanto às decisões tomadas.
- Facilita o acompanhamento do processo de cada projeto, tanto pelos gerentes como dos clientes.



---

# Conceitos de Gestão em Software



# Gestão em Software

---

- O que está envolvido quando recebemos a incumbência de desenvolver um software?
  - Produto
  - Processo
  - Construção
  - Pessoal



# Gestão em Software

---

- O produto
  - Qual é o problema do cliente?
  - O que o cliente deseja?
    - O software está alinhado com o negócio do cliente?
    - Quais são as **funcionalidades** e as **restrições** do software?



# Gestão em Software

---

- O produto
  - O que mais precisamos descobrir a respeito do produto?
    - Qual é o prazo de entrega?
    - Qual é o custo?
    - Qual é a diferença entre custo e preço?



# Gestão em Software

---

- O processo
  - Quais são as **etapas** para a produção do produto e como estas se **relacionam**?
  - Que **tarefas** estão relacionadas a cada etapa?
  - O processo **está adequado** às necessidades do produto/cliente?



# Gestão em Software

---

- A construção
  - Quais são os **riscos** do produto?
    - Afeta prazos e custo.
  - A equipe tem a **aptidão** necessária?
  - A equipe está **motivada**?
  - Qual será a intensidade dos **testes** e **inspeções**?

# Gestão em Software

- O pessoal: Que pessoas estão envolvidas no desenvolvimento de um software?

## ■ Cliente

- Usuários - Pessoal Operativo;
- Gerentes;
- Diretoria;

## ■ Fornecedor

- Analista de Negócios
- Gerente de Projetos
- Analistas de Sistemas;
- Projetistas de Sistemas;
- Programadores;
- Testadores;
- Pessoal de implantação e treinamento
- Pessoal de Suporte Manutenção
- Auditores, pessoal de controle e qualidade





# Gestão em Software

---

- Pessoal
  - Quais devem ser as preocupações com o pessoal?
    - Comunicação
    - Confiança
    - Distribuição de aptidões
    - Coesão
  - “Trabalhar com pessoas é difícil, mas não impossível.” (Peter Drucker)



# Gestão em Software

---

- Pessoal
  - Como se organiza uma equipe?
    - **Paradigma Fechado:** Hierarquia tradicional de autoridade.
    - **Paradigma Aleatório:** Iniciativa individual de seus membros (caos).
    - **Paradigma Aberto:** Mescla (Fechado + Aleatório); Decisões em consenso.
    - **Paradigma Síncrono:** Organizada em torno da decomposição do problema. Pouca interação.



# Gestão em Software

---

- Considerações Finais
  - Não perca o foco:
    - Por que o software está sendo desenvolvido?
    - O que vai ser feito?
    - Quando vai ser feito?
    - Quem vai fazer?
    - Onde eles estão?
    - Como o trabalho será conduzido?
    - Quanto recurso será dispendido?



---

# Processo de Software



# Processo de Software

---

- O que se espera da ESOF?
  - Processo de construção de software
    - Modelos, ferramentas, padrões, qualidade, prazos e custos estimados.
- O que é um processo de software?
  - Sequência de atividades (e resultados associados) que levam à produção de um software de qualidade.



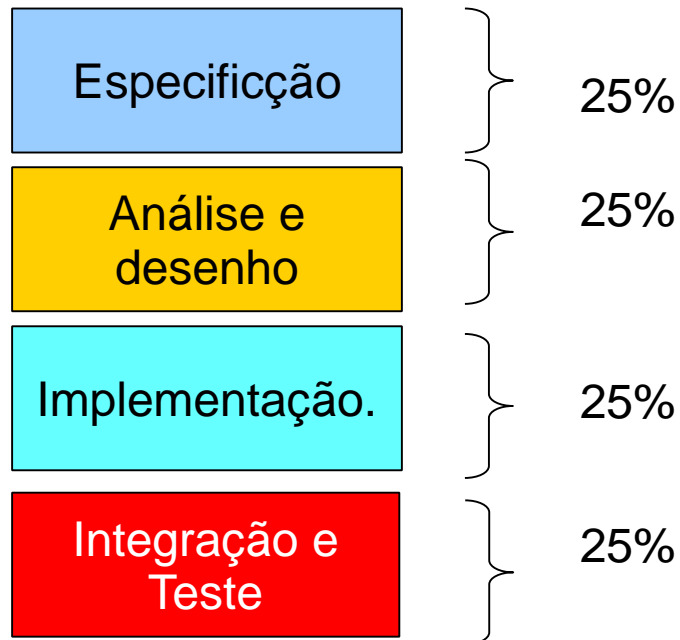
# Processo de Software

---

- Atributos de qualidade de um software
  - Manutenibilidade
  - Robustez
    - confiabilidade
    - proteção
    - segurança
  - Eficiência
  - Usabilidade

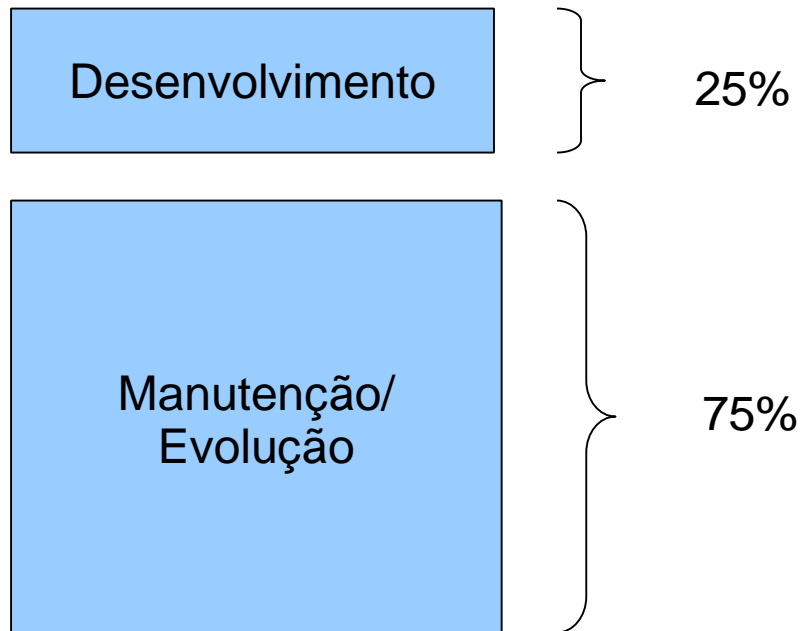
# Processo de Software

- Custos associados ao software
  - Desenvolvimento



# Processo de Software

- Custos associados ao software
  - Evolução







# Processo de Software

---

- Alguns desafios para a ESOF
  - Sistemas Legados
  - Heterogenidade dos ambientes
  - Entrega
    - no prazo especificado
    - dentro do custo estimado
    - sem erro

# Processo de Software

---

- Ética na ESOF
  - Aspectos relacionados a
    - confidencialidade
    - competência
    - propriedade intelectual
    - mal uso do computador



---

# Cronograma de Projeto a ser Desenvolvido



# Cronograma de Projeto

---

- O curso de Modelagem de Software desenvolve um projeto completo de um sistema usando os métodos apresentados em aula:
  - Grupos de no máximo 5 alunos
  - O sistema a ser desenvolvido será escolhido pelos alunos.
  - É necessário um cliente real com necessidades reais.



# Cronograma de Projeto

---

- Cronograma:
  - **Etapa 1 (22/03)** - 5 pontos - Definição das equipes e do sistema a ser desenvolvido
  - **Etapa 2 (05/04)** - 5 pontos - Identificação das necessidades, coleta inicial de dados com estudo de viabilidade do sistema
    - **Template 1 - Documento de Modelagem de Negócio**
  - **Etapa 3 (22/04)** - 5 pontos - Análise Estruturada de Sistemas (Lista de eventos, diagrama de contexto, diagrama de fluxo de dados completo e modelo de entidade e relacionamento notação Chen)
    - **Template 2 - Documento de Análise Estruturada**
  - **Etapa 4 (20/05)** - 5 pontos - Análise, especificação e validação dos requisitos (diagrama de caso de uso com especificação dos completa dos cenários)
    - **Template 3 - Documento de Requisitos de Projeto**
  - **Etapa 5 (17/16)** - 5 pontos - Análise de Sistemas (modelo conceitual de objetos, diagrama de estado de objetos e diagramas de sequência, diagrama de entidade e relacionamentos – DER, protótipos das interfaces, diagrama de implantação)
    - **Template 4 - Documento de Análise e Projeto do Sistema**
  - **Etapa 6 (08/07)** - 5 pontos - Avaliação final de todos os templates