

# Processos e ciclos de vida de software

Disciplina Engenharia de Software

Professora Adriana Gomes Alves, Dra

[adriana.alves@univali.br](mailto:adriana.alves@univali.br)

# Processo de software



# Processo de software

É um conjunto (framework) de atividades e resultados associados que geram um produto de software.





# Atividades fundamentais

- ☐ *Especificação do software;*
- ☐ *Desenvolvimento do Software;*
- ☐ *Validação do Software;*
- ☐ *Evolução do Software;*

# Visão geral de um processo



## Definição – “o que”

- Engenharia do sistema
- Planejamento do projeto
- Análise de requisitos

## Desenvolvimento – “como”

- Projeto
- Geração de código
- Teste

## Manutenção

- Análise
- Implementação

# Processo de software

define uma abordagem

# Modelo de processo

descrição simplificada, uma abstração dessa abordagem

# Modelo de processo...

Ou ciclo de vida de software...





O desenvolvimento de um produto inicia com uma ideia e termina com o produto pretendido.



O ciclo de vida de um produto é a definição dos passos que transformam a ideia no produto acabado



# Ciclo de Vida do Software

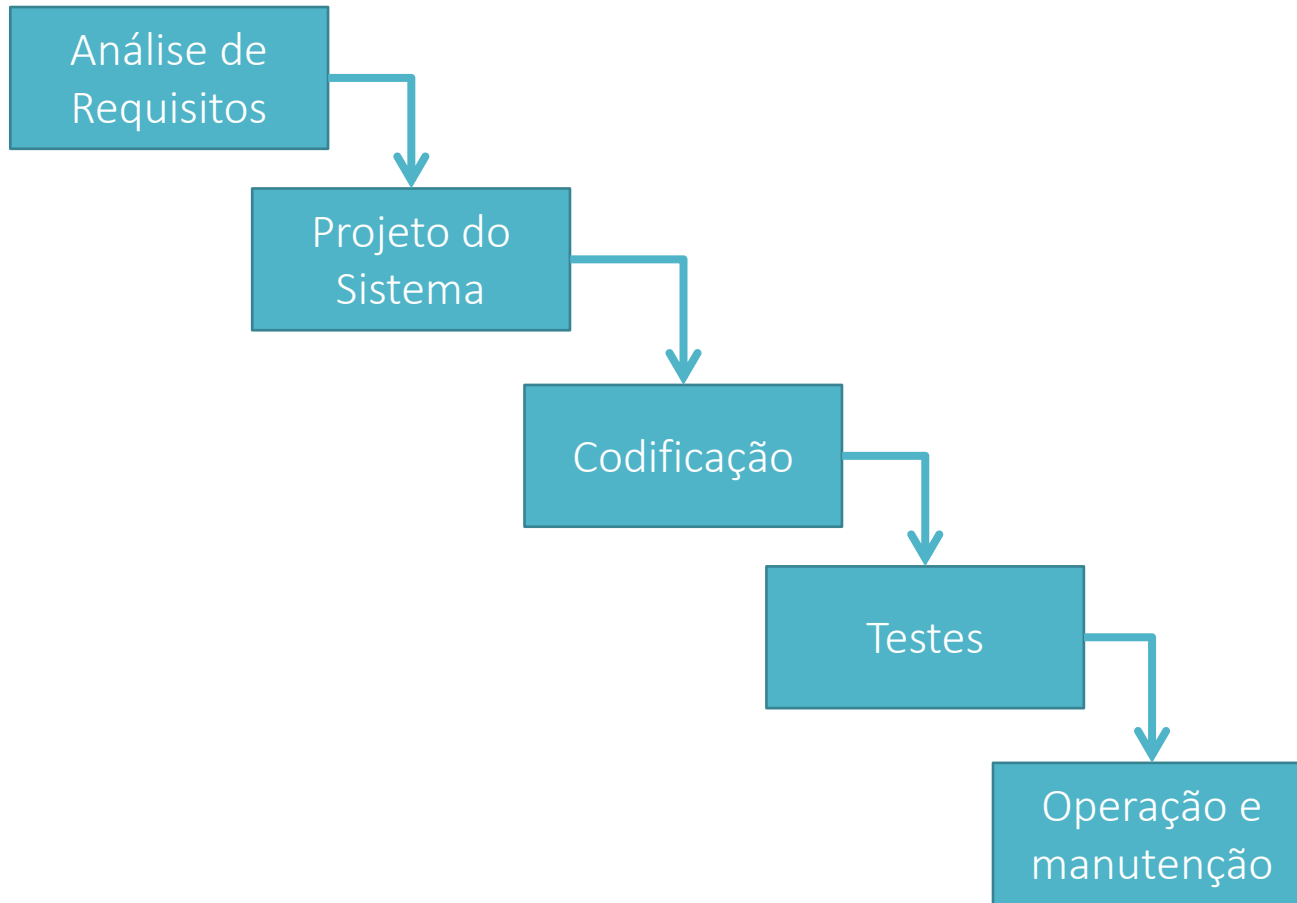


Quais as etapas de um projeto de software?

Etapa 1

Etapa 3

# Modelo Cascata (Royce, 1970)



# Modelo Cascata (Royce, 1970)

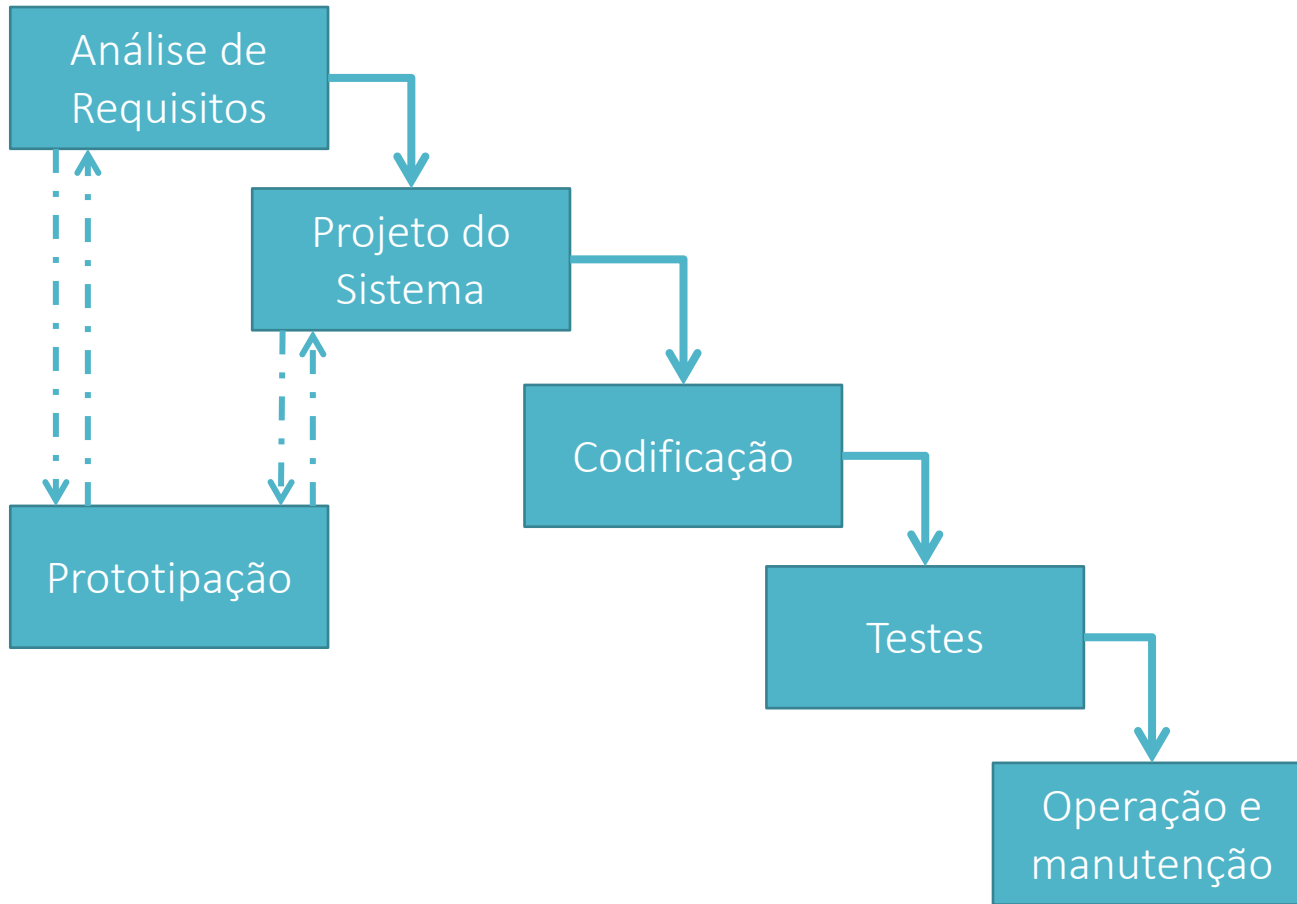
## Vantagens:

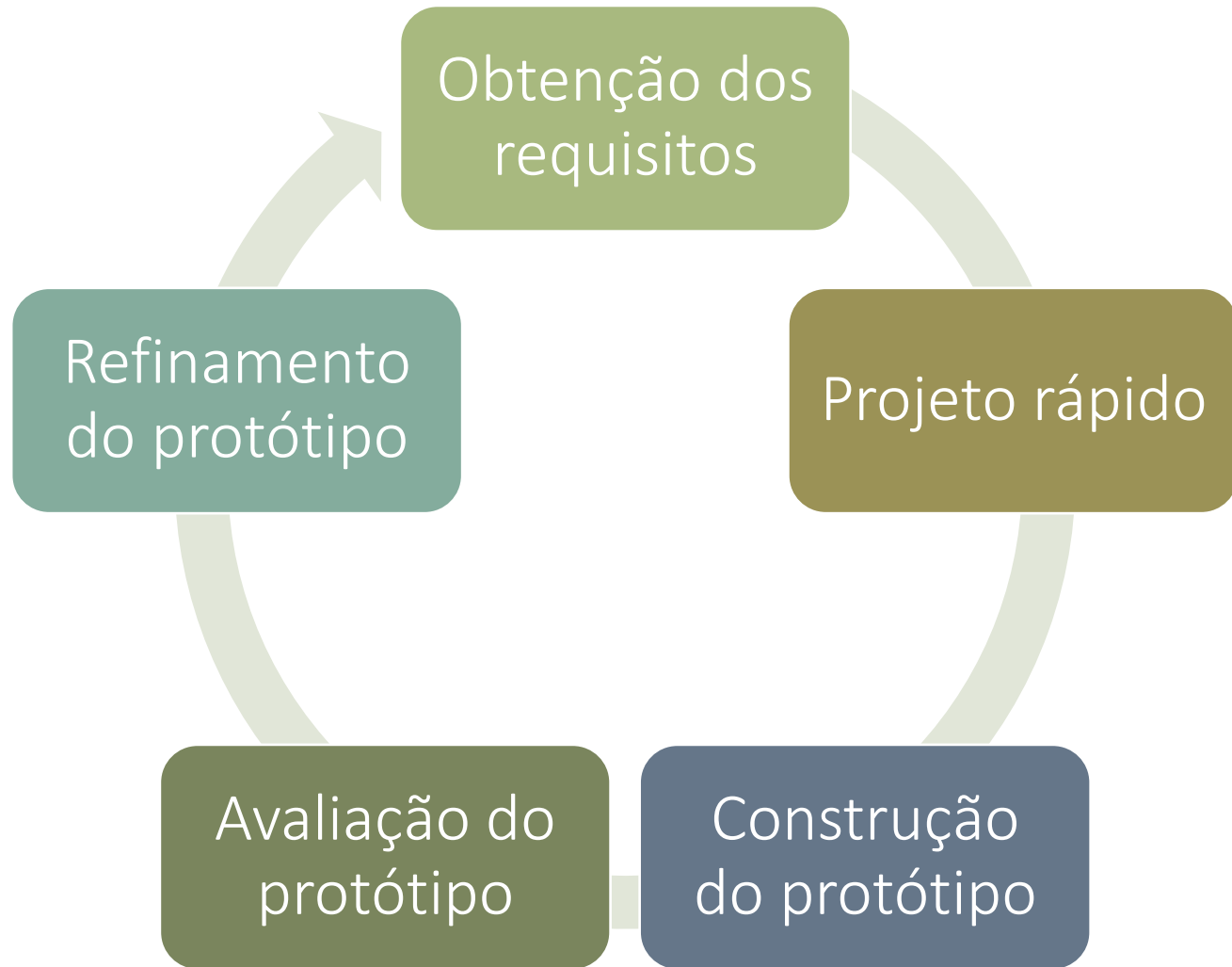
- ✓ Facilidade no gerenciamento;
- ✓ Simples de compreender e de comunicar à equipe e ao cliente;
- ✓ Sequenciamento lógico com etapas muito bem definidas.

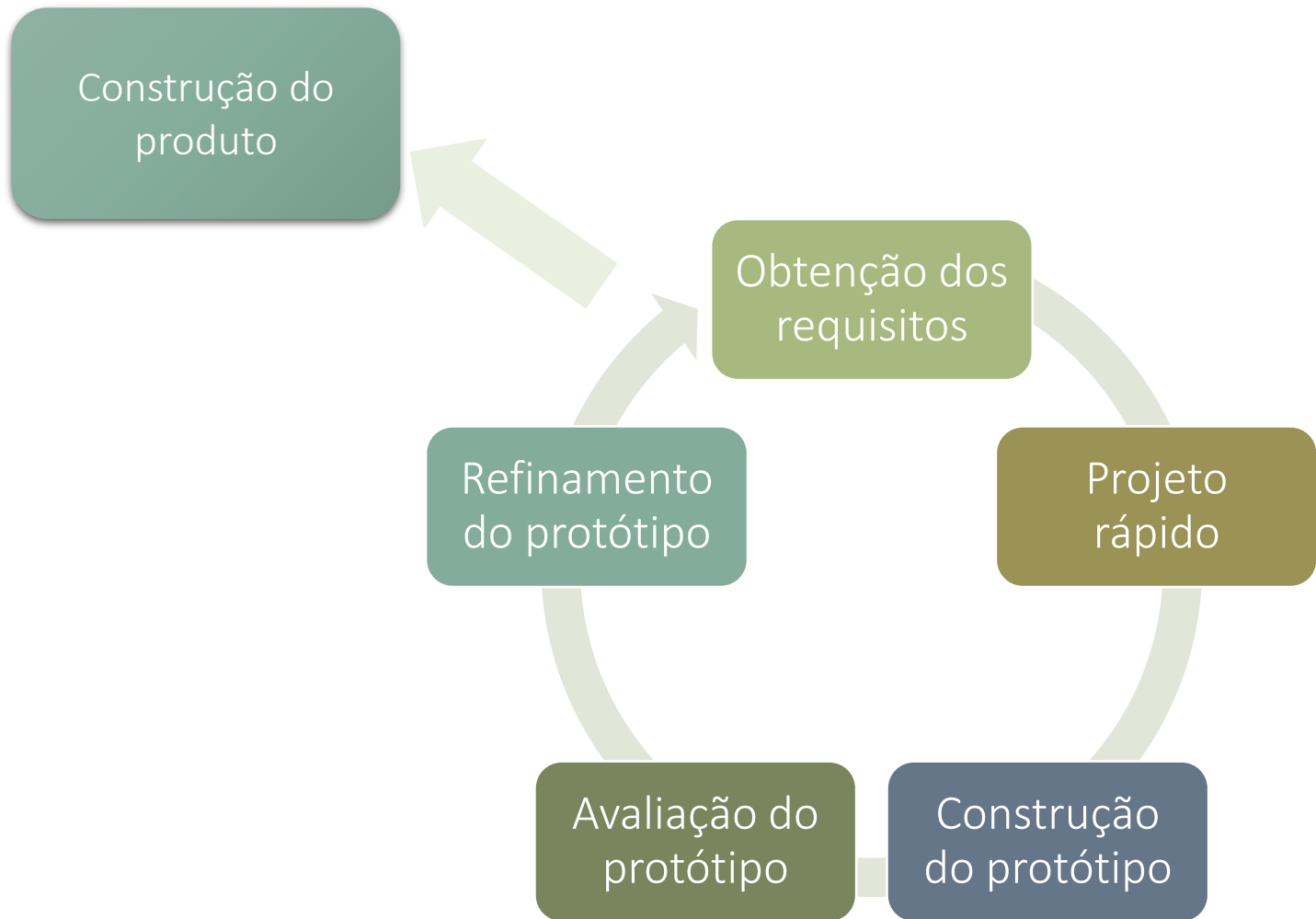
## Desvantagens:

- ✓ Dificuldade na revisão dos requisitos.
- ✓ Cada etapa deve ser totalmente concluída antes de iniciar a etapa seguinte.
- ✓ As entregas do software ocorrem apenas nas últimas etapas.

# Prototipação com modelo em cascata







# Prototipação com modelo em cascata

## Vantagens:

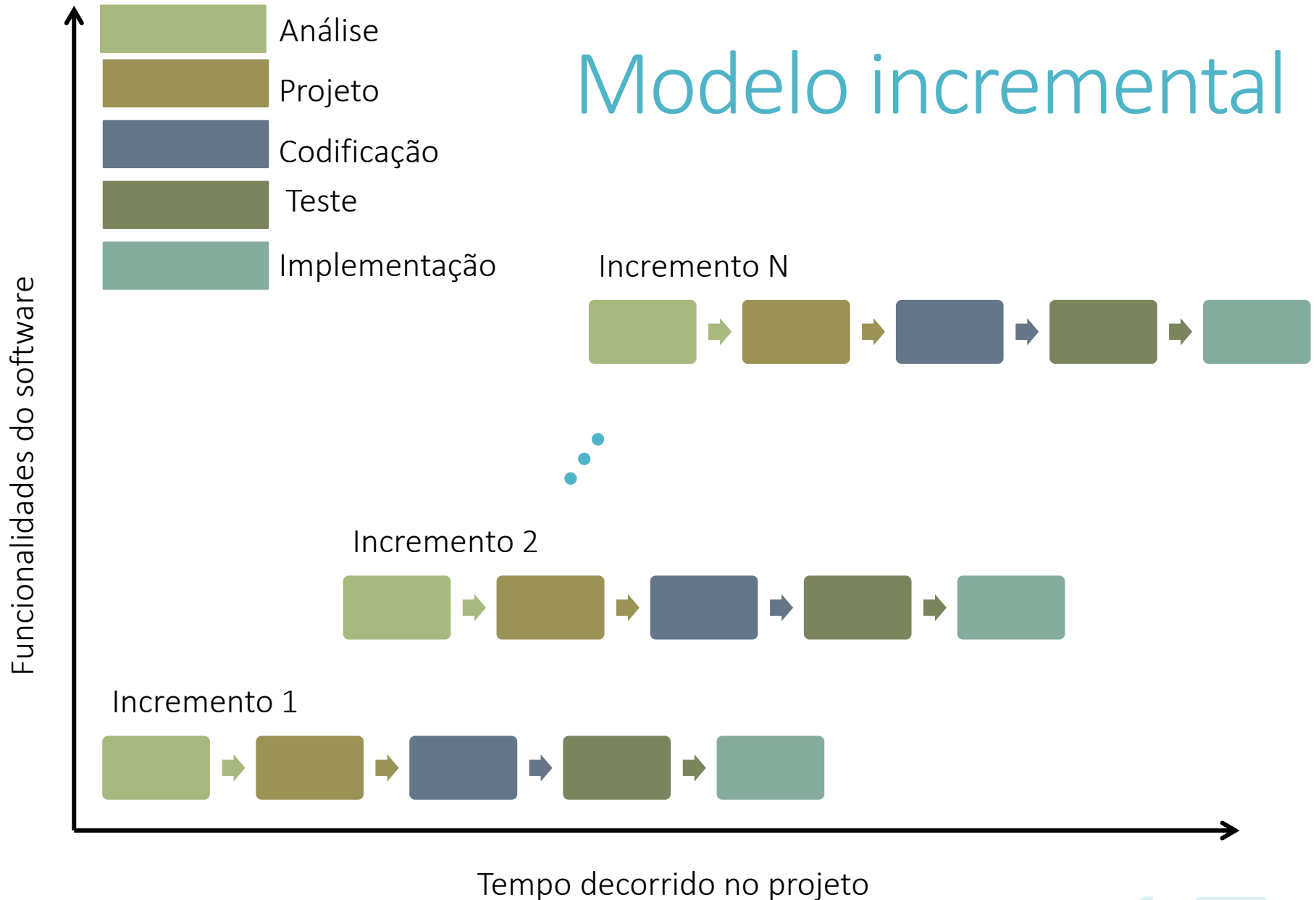
- ✓ O protótipo permite ao cliente visualizar o sistema proposto.
- ✓ As experimentações na construção do protótipo podem reduzir os custos das etapas posteriores.
- ✓ Reduz, momentaneamente, o risco e a incerteza do desenvolvimento.

## Desvantagens:

- ✓ Pressão do cliente para adiantar a entrega final.
- ✓ As decisões não bem planejadas na construção do protótipo podem se tornar parte do sistema final.
- ✓ Custo do protótipo, que normalmente é descartado na construção do sistema final.



# Modelo incremental



# Modelo Incremental

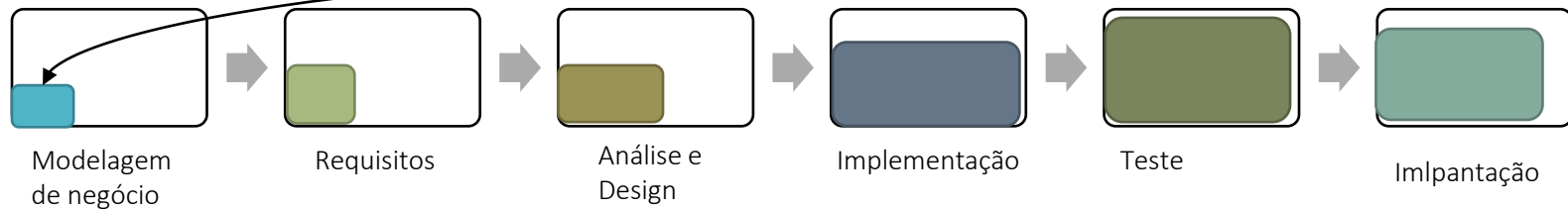
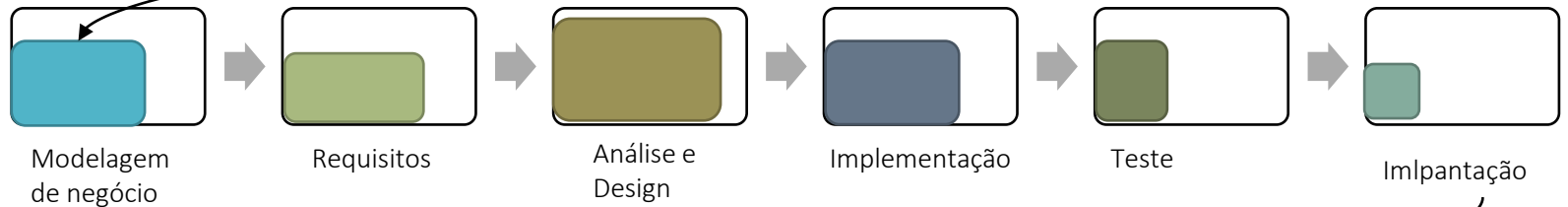
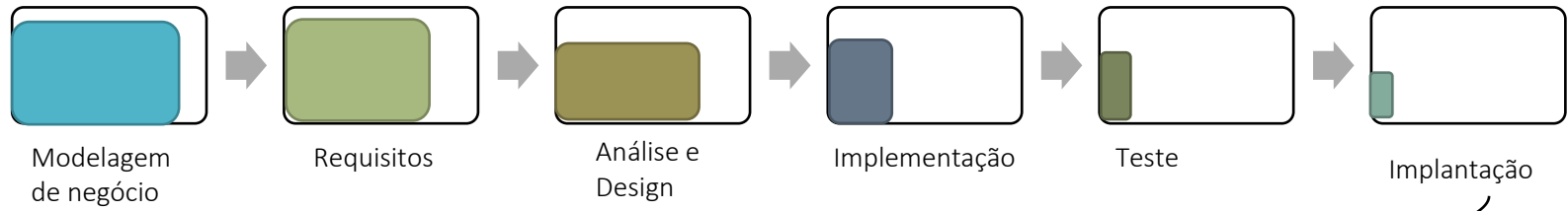
## Vantagens:

- ✓ Facilidade em testar o sistema, pois a cada incremento testa-se o sistema final.
- ✓ A utilização do sistema, mesmo incompleto permite ao cliente validar as entregas e sugerir com maior segurança alterações para as etapas seguintes.

## Desvantagens:

- ✓ Maior complexidade na elaboração do contrato, necessidade pré-determinar o custo das mudanças.
- ✓ Mesmo nos primeiros incrementos o sistema deve apresentar uma estrutura robusta para suportar o incremento de funções com pouco retrabalho.

# Modelo Iterativo



# Modelo Iterativo

## Vantagens:

- ✓ Envolvimento do cliente no processo, permitindo que as alterações nos requisitos sejam rapidamente incorporadas.
- ✓ As entregas baseadas em protótipos permitem aos usuários e a equipe confiança de estar no caminho certo.

## Desvantagens:

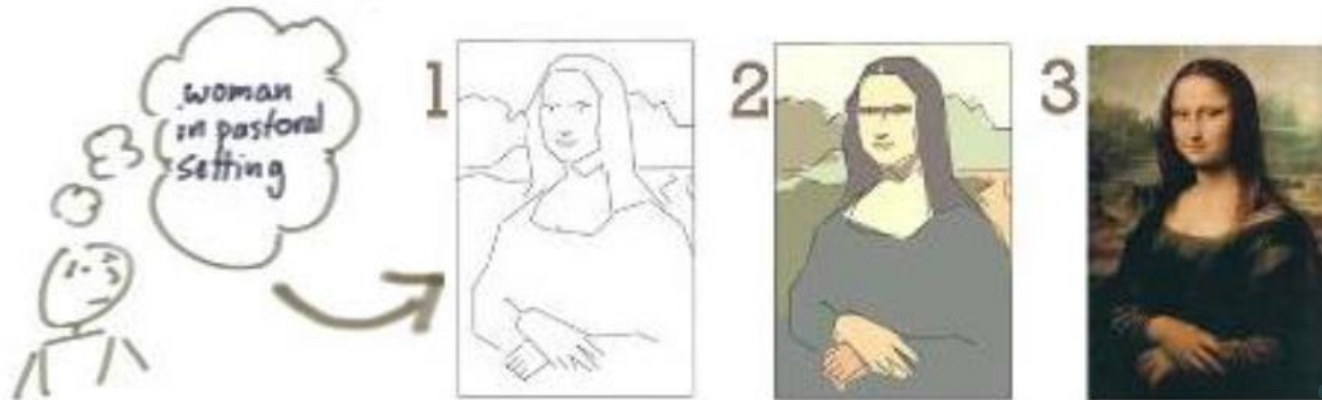
- ✓ Continuamente podem surgir novos requisitos dificultando ou impedindo o término do projeto.
- ✓ Pode-se gerar um alto custo com a construção/evolução dos protótipos.

# Iterativo x Incremental

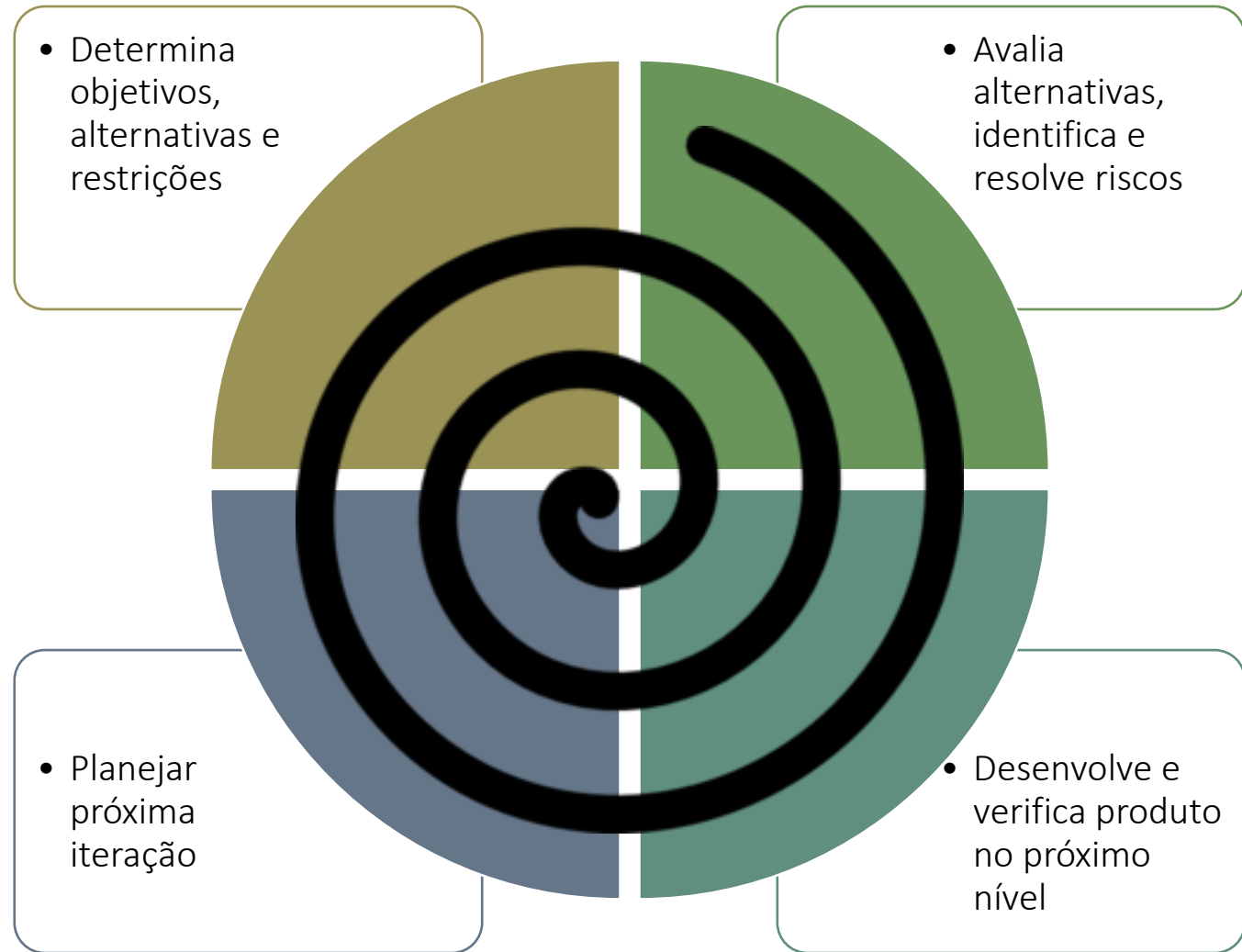
## Incremental



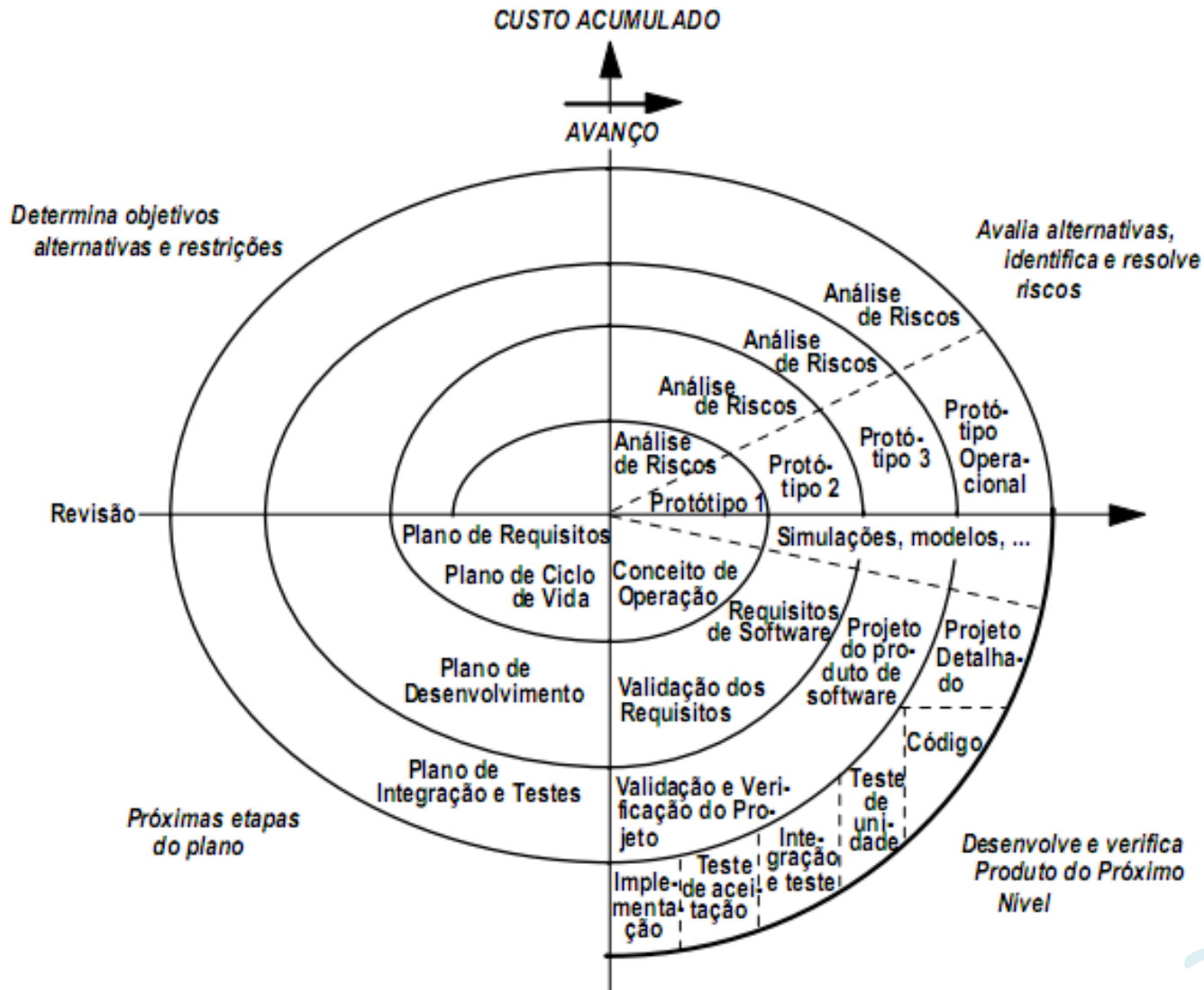
## Iterative



# Modelo espiral



# Modelo Espiral



# Modelo Espiral

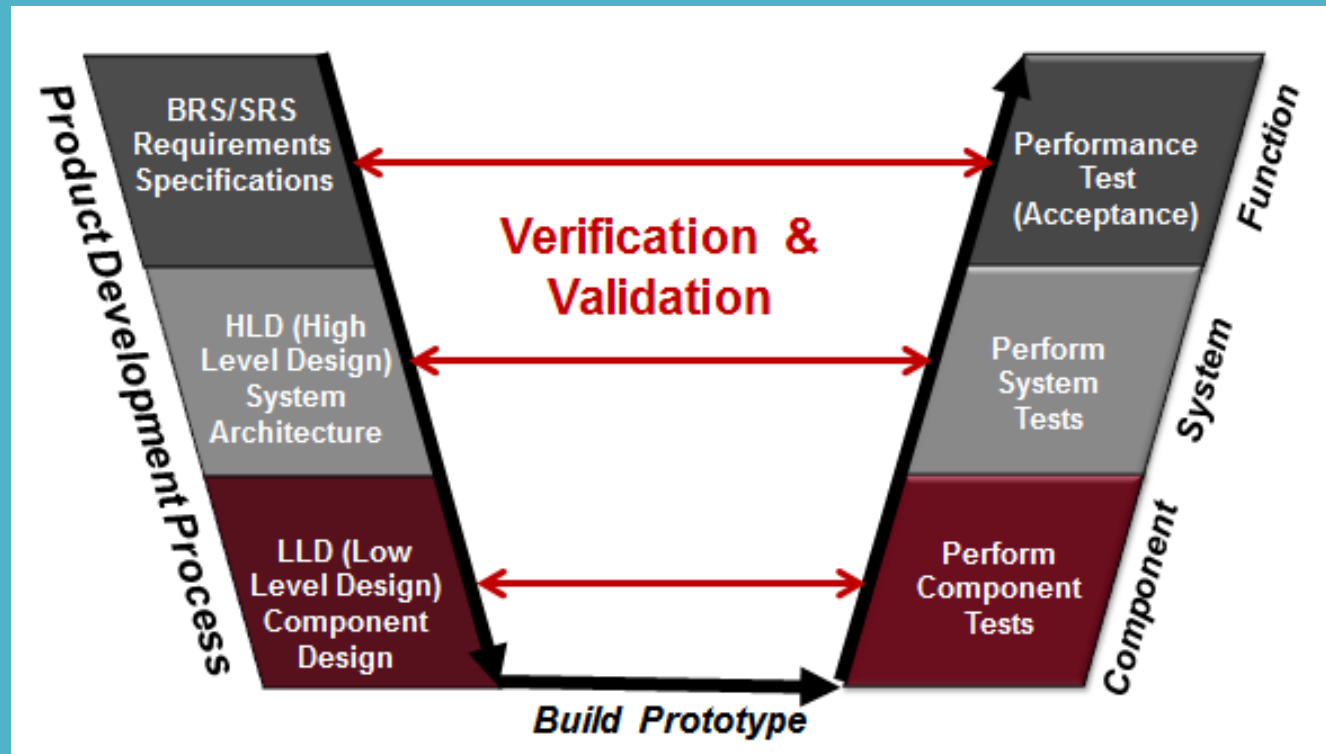
## Vantagens:

- ✓ O foco na avaliação de riscos pode ser considerado um fator diferencial em grandes projetos.
- ✓ Abordagem evolucionária, a medida que o projeto avança a equipe e o cliente entendem melhor o projeto e os riscos envolvidos.

## Desvantagens:

- ✓ Exige experiência na avaliação dos riscos, por se tratar de uma etapa fundamental no sucesso do projeto.
- ✓ Pode ser difícil convencer o cliente que a abordagem evolutiva é controlável.





# Processo de Software – Modelo em V

# Processo de Software – Modelo em V

## Vantagens de V-modelo:

Simple e fácil de usar.

Testando atividades como planejamento, teste de concepção, simulações bem antes da construção. Isso economiza muito tempo. Daí maior chance de sucesso sobre o modelo em cascata (*waterfall model*).

Rastreamento de defeitos pró-ativa, isto é os defeitos são encontrados em fase inicial.

Evita o fluxo descendente dos defeitos.

Funciona bem para pequenos projetos onde os requisitos são facilmente compreendidos.

## Desvantagens de V-modelo:

Se alguma mudança acontecer no meio do caminho, em seguida, os documentos de testes e análises, juntamente com os documentos de requisitos deverão ser atualizados.

# Escolha do modelo de ciclo de vida

- Qual o nível de compreensão do usuário e desenvolvedores em relação aos requisitos no início do projeto? Considera-se provável alguma mudança significativa durante o desenvolvimento do projeto?
- Qual o nível de compreensão dos desenvolvedores em relação a arquitetura do sistema? A arquitetura do sistema pode ser revista durante o projeto?
- Qual nível de confiança é necessário?
- O quanto é necessário planejar e projetar durante o projeto prevendo mudanças em versões futuras?
- Qual o nível de riscos implícitos no projeto?
- Pode ser limitado em um cronograma?
- É necessário habilidade para realizar correções no meio do projeto?
- É necessário mostrar ao cliente o progresso durante o projeto?
- É necessário demonstrar ao usuário aspectos gerenciais durante o projeto?

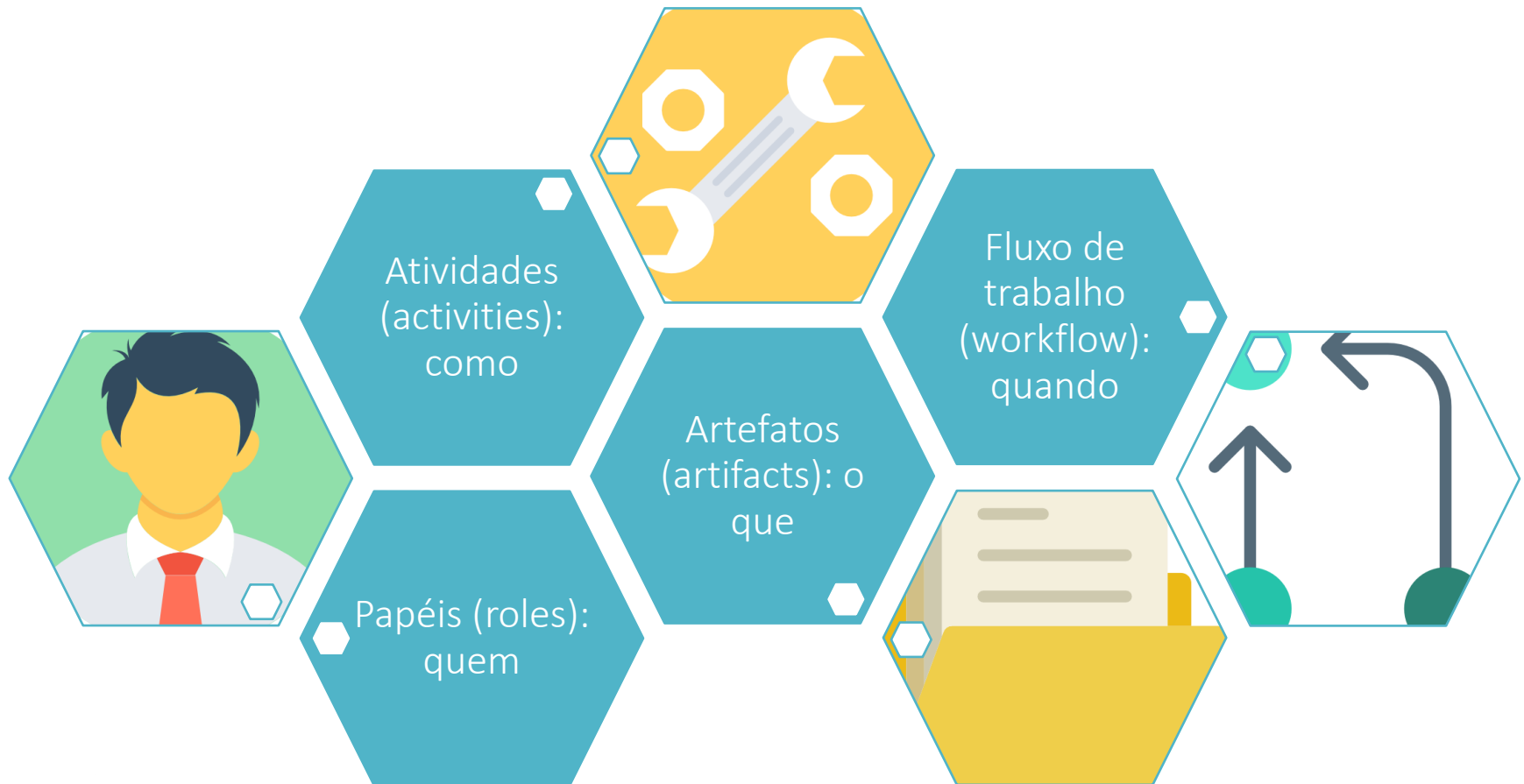
# PROCESSO

Conjunto de atividades coordenadas com o objetivo de transformar um conjunto necessário de entradas em saídas desejadas.



Com base em um ciclo de vida, define em detalhes, quem faz o que, quando e como, para atingir um determinado objetivo.

# Elementos do processo de software



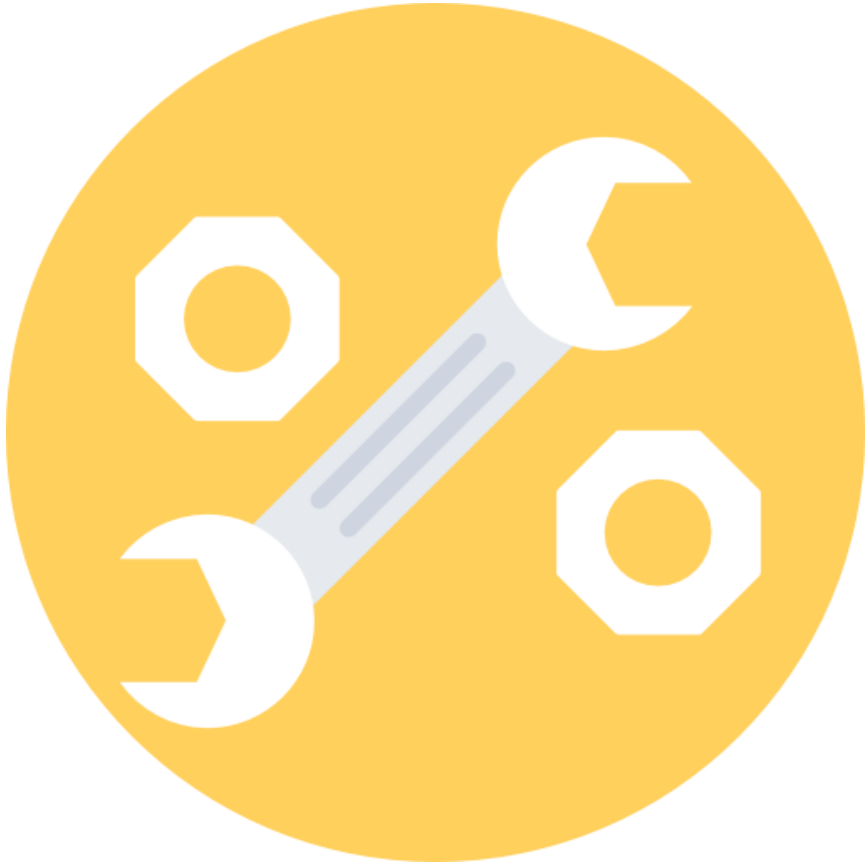


## Papéis

Definição abstrata de um conjunto de responsabilidades (atividades) e dos respectivos artefatos (entregas) atribuídos a um membro da equipe.

Um membro da equipe pode desempenhar vários papéis em um mesmo projeto.

Exemplo: Analista de sistema;  
Programador; Testador;



## Atividades

Unidade de trabalho que um papel deve executar durante o projeto que resulte em um ou mais artefatos;

Exemplo: Identificar requisitos; Executar um conjunto de testes; Prototipar telas do sistema.

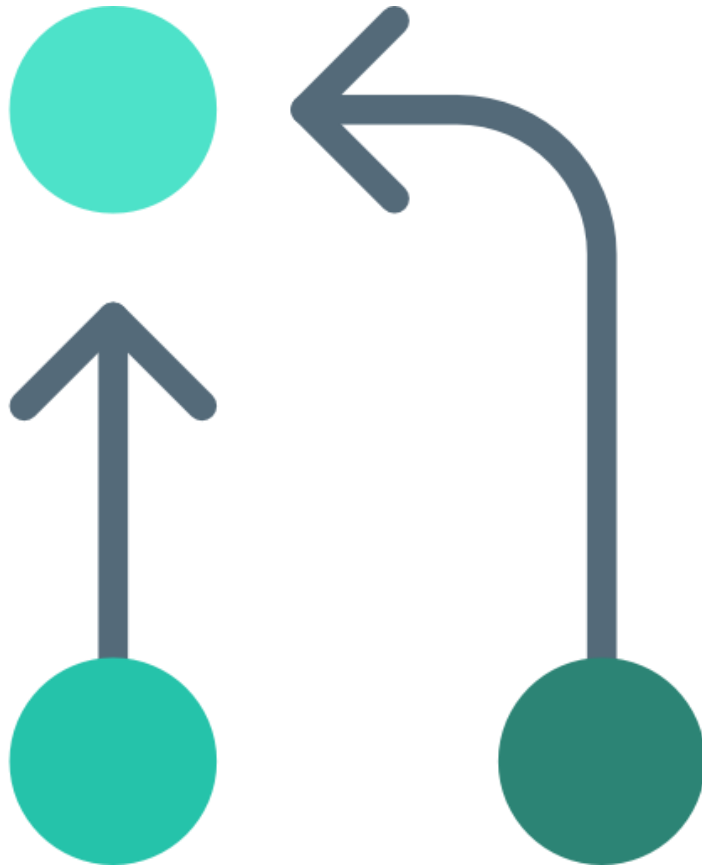
## Artefatos

Informação ou conteúdo utilizado ou produzido por uma atividade no processo de software.

Exemplo: Documento de requisitos; Modelo de casos de uso; Código de um classe; Componente do sistema;







## Fluxo de trabalho

Sequência de atividades no processo de software que devem ser executadas em certa ordem para atingir um determinado objetivo.

# Processos de Software

RUP – Rational Unified Process

ICONIX

XP - Extreme Programming

Scrum

# Processos de Software

RUP – Rational Unified Process

ICONIX

XP - Extreme Programming

Scrum

Tradicional/Formal

Ágil

## Metodologias Ágeis

- Indivíduos e interação entre eles mais que processos e ferramentas;
- Software em funcionamento mais que documentação abrangente;
- Colaboração com o cliente mais que negociação de contratos;
- Responder a mudanças mais que seguir um plano.
- Exemplo:
  - Scrum

## Processos Tradicionais/Formais

- Processos detalhados
- Análise de risco
- Adequado a grandes equipes
- Sistemas críticos
- Exemplo:
  - Rational Unified Process (RUP)

# Abordagem do Processo de Software

# Referências

PRESSMAN, Roger S. Engenharia de software: uma abordagem profissional. 7. ed. São Paulo, SP: McGraw-Hill Interamericana do Brasil, 2011. xxviii, 780 p. ISBN 0073375977.

SOMMERVILLE, Ian. Engenharia de software. 9. ed. São Paulo, SP: Pearson Education do Brasil, c2011. xiii, 529 p. ISBN 9788579361081.