



AULA 8:

MODELOS DE PROCESSO DE
SOFTWARE: CASCATA,
PROTOTIPAÇÃO E RAD

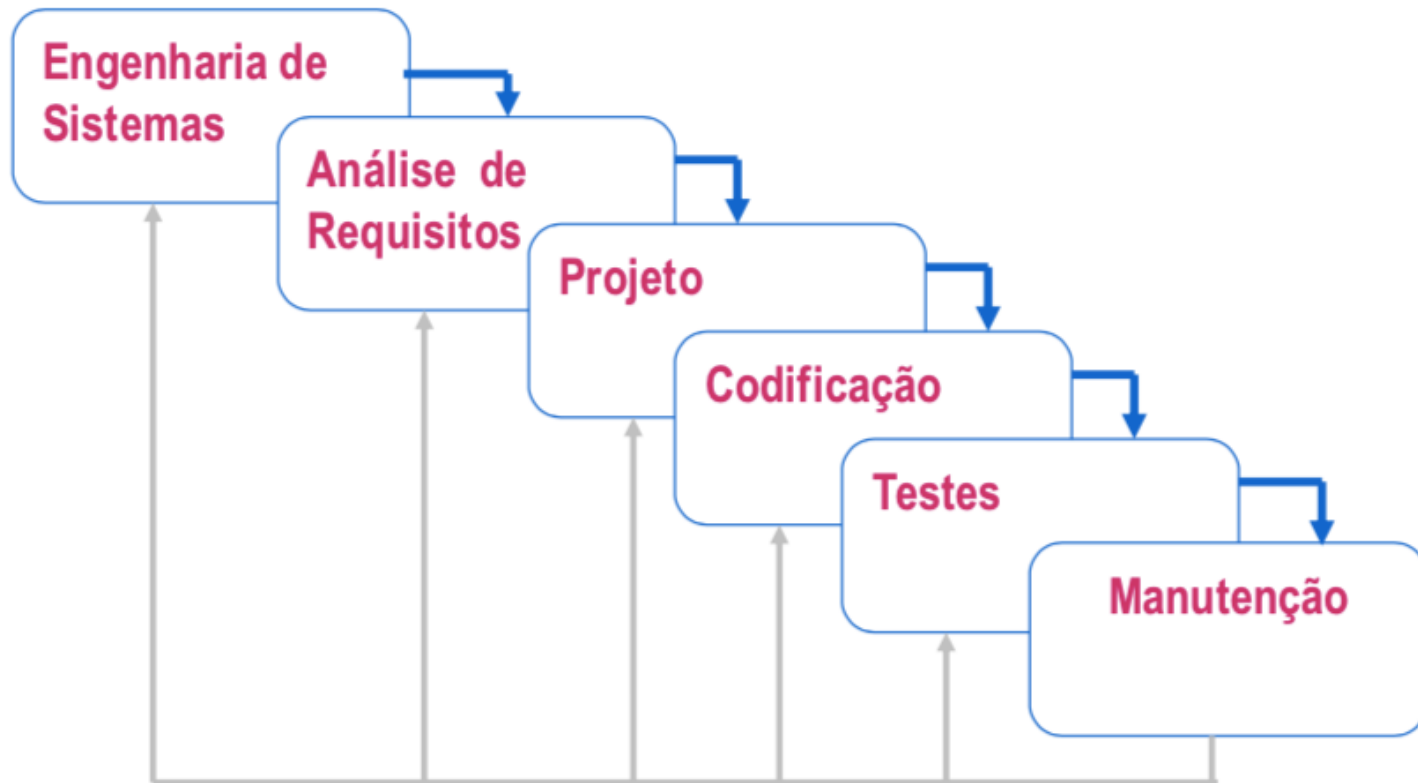
Modelos de Processo de Software

- ❖ **Modelo Sequencial Linear**
 - ❖ Também chamado Ciclo de Vida Clássico ou **Modelo Cascata**.
- ❖ O Paradigma de Prototipação
- ❖ Técnicas de Quarta Geração
- ❖ Modelo RAD (Rapid Application Development)
- ❖ Modelos de Métodos Formais
- ❖ Modelos Evolutivos de Processo de Software
 - ❖ Modelo Incremental.
 - ❖ Modelo Espiral.
 - ❖ Modelo de Montagem de Componentes.
 - ❖ Modelo de Desenvolvimento Concorrente.

O Modelo Cascata

- ❖ A aplicação de uma abordagem sistemática, disciplinada e possível de ser medida para o desenvolvimento, operação e manutenção do software (*IEEE*).
- ❖ Modelo mais antigo e o mais amplamente usado da engenharia de software.
- ❖ Modelado em função do ciclo da engenharia convencional.
- ❖ Requer uma abordagem sistemática, sequencial ao desenvolvimento de software.
- ❖ O resultado de uma fase se constitui na entrada da outra.

O Modelo Cascata



O Modelo Cascata

1. Engenharia de Sistemas / Informação e Modelagem

- ❖ Envolve a coleta de requisitos em nível do sistema, com uma pequena quantidade de projeto e análise de alto nível.
- ❖ Esta visão é essencial quando o software deve fazer interface com outros elementos (hardware, pessoas e banco de dados).

O Modelo Cascata

2. Análise de Requisitos de Software

- ❖ O processo de coleta dos requisitos é intensificado e concentrado especificamente no software.
- ❖ Deve-se compreender o domínio da informação, a função, desempenho e interfaces exigidos.
- ❖ Os requisitos (para o sistema e para o software) são documentados e revistos com o cliente

O Modelo Cascata

3. Projeto

- ❖ Tradução dos requisitos do software para um conjunto de representações que podem ser avaliadas quanto à qualidade, antes que a codificação se inicie.

4. Codificação

- ❖ Tradução das representações do projeto para uma linguagem "artificial", resultando em instruções executáveis pelo computador.

O Modelo Cascata

5. Testes

- ❖ Concentra-se:
 - ❖ Nos aspectos lógicos internos do software, garantindo que todas as instruções tenham sido testadas.
 - ❖ Nos aspectos funcionais externos, para descobrir erros e garantir que a entrada definida produza resultados que concordem com os esperados.

O Modelo Cascata

6. Manutenção

- ❖ Provavelmente o software deverá sofrer mudanças depois que for entregue ao cliente.
- ❖ Causas das mudanças: erros, adaptação do software para acomodar mudanças em seu ambiente externo e exigência do cliente para acréscimos funcionais e de desempenho.

O Modelo Cascata

Problemas

- ❖ Projetos reais raramente seguem o fluxo sequencial que o modelo propõe.
- ❖ Logo no início é difícil estabelecer explicitamente todos os requisitos. No começo dos projetos sempre existe uma incerteza natural.
- ❖ O cliente deve ter paciência. Uma versão executável do software só fica disponível numa etapa avançada do desenvolvimento.
- ❖ Embora o modelo Cascata tenha fragilidades, ele é significativamente melhor do que uma abordagem casual ao desenvolvimento de software.

O Modelo Cascata

Contribuições

- ❖ O modelo Cascata trouxe contribuições importantes para o processo de desenvolvimento de software:
 - ❖ Imposição de disciplina, planejamento e gerenciamento.
 - ❖ A implementação do produto deve ser postergada até que os objetivos tenham sido completamente entendidos.

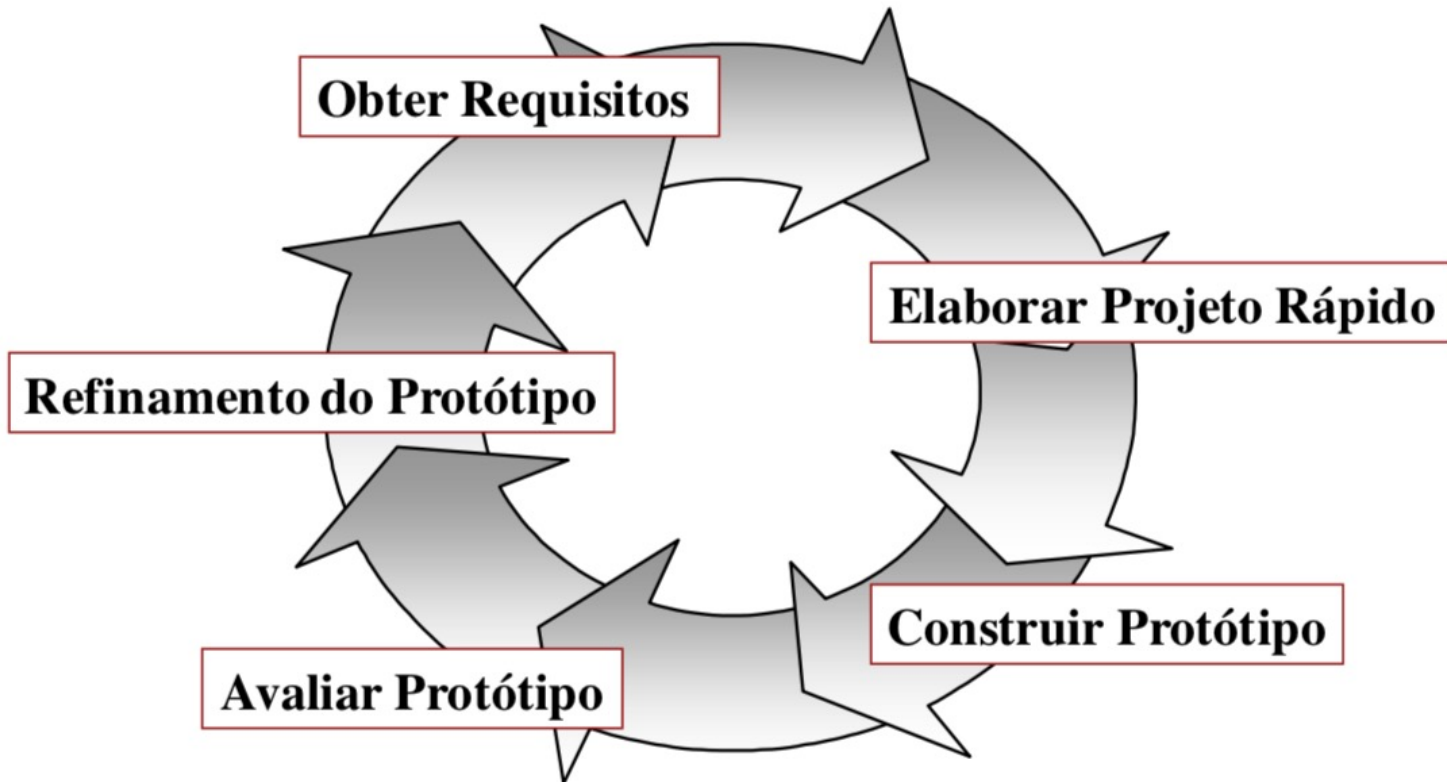
Modelos de Processo de Software

- ❖ Modelo Sequencial Linear
 - ❖ Também chamado Ciclo de Vida Clássico ou **Modelo Cascata**.
- ❖ O Paradigma de Prototipação
- ❖ Técnicas de Quarta Geração
- ❖ Modelo RAD (Rapid Application Development)
- ❖ Modelos de Métodos Formais
- ❖ Modelos Evolutivos de Processo de Software
 - ❖ Modelo Incremental.
 - ❖ Modelo Espiral.
 - ❖ Modelo de Montagem de Componentes.
 - ❖ Modelo de Desenvolvimento Concorrente.

O Paradigma de Prototipação

- ❖ O objetivo é entender os requisitos do usuário e, assim, obter uma melhor definição dos requisitos do sistema.
- ❖ Possibilita que o desenvolvedor crie um modelo (protótipo) do software que deve ser construído.
- ❖ Adequado para quando o cliente não definiu detalhadamente os requisitos.

O Paradigma de Prototipação



O Paradigma de Prototipação

Obtenção dos Requisitos

- ❖ Desenvolvedor e cliente definem os objetivos gerais do software, identificam quais requisitos são conhecidos e as áreas que necessitam de definições adicionais.

Projeto Rápido

- ❖ Representação dos aspectos do software que são visíveis ao usuário (abordagens de entrada e formatos de saída).

O Paradigma de Prototipação

Construção do Protótipo

- ❖ Implementação rápida do projeto.

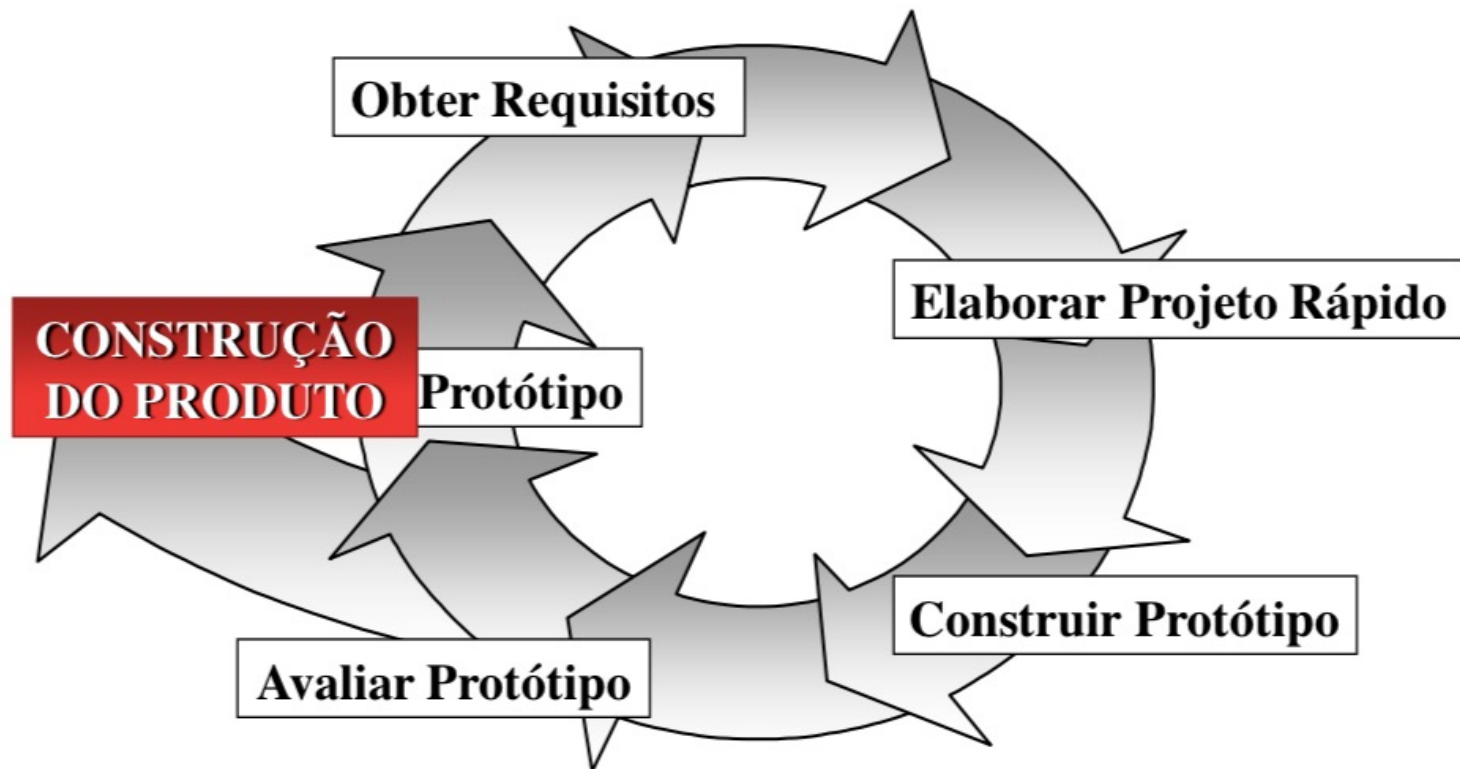
Avaliação do Protótipo

- ❖ Cliente e desenvolvedor avaliam o protótipo.

Refinamento do Protótipo

- ❖ Cliente e desenvolvedor refinam os requisitos de software a ser desenvolvido.

O Paradigma de Prototipação



O Paradigma de Prototipação

Construção do Produto

- ❖ Identificados os requisitos, o protótipo deve ser descartado e a versão de produção deve ser construída considerando os critérios de qualidade.

Problemas

- ❖ Cliente não sabe que o software que ele vê não considerou, durante o desenvolvimento, a qualidade global e a manutenibilidade a longo prazo.
- ❖ Desenvolvedor frequentemente faz uma implementação comprometida (utilizando o que está disponível) com o objetivo de produzir rapidamente um protótipo.

O Paradigma de Prototipação

Comentários

- ❖ Ainda que possam ocorrer problemas, a prototipação é um ciclo de vida eficiente.
- ❖ A chave é definir-se as regras do jogo logo no começo.
- ❖ O cliente e o desenvolvedor devem ambos concordar que o protótipo seja construído para servir como um mecanismo a fim de definir os requisitos

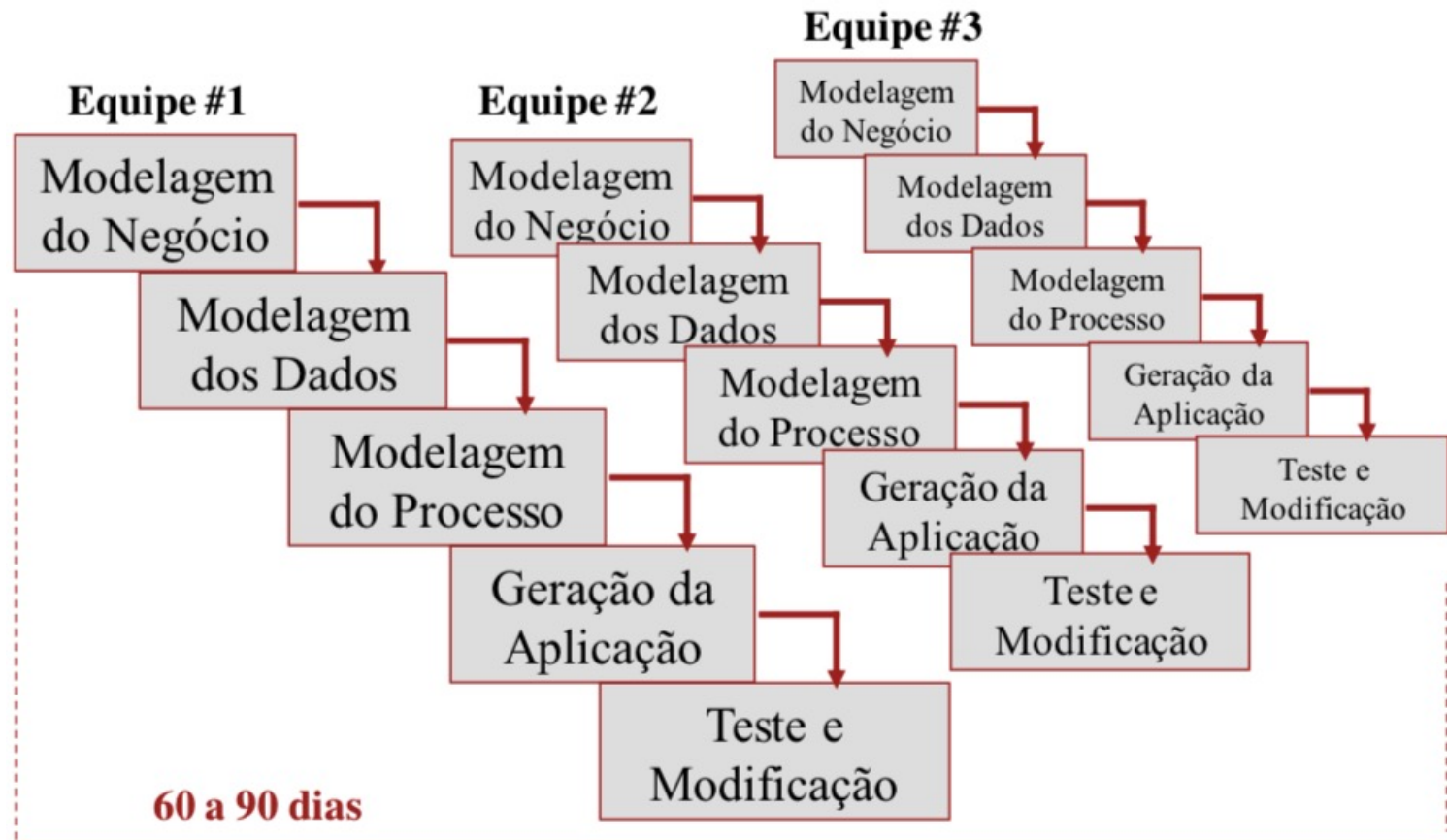
Modelos de Processo de Software

- ❖ Modelo Sequencial Linear
 - ❖ Também chamado Ciclo de Vida Clássico ou Modelo Cascata.
- ❖ O Paradigma de Prototipação
- ❖ Técnicas de Quarta Geração
- ❖ **Modelo RAD (Rapid Application Development)**
- ❖ Modelos de Métodos Formais
- ❖ Modelos Evolutivos de Processo de Software
 - ❖ Modelo Incremental.
 - ❖ Modelo Espiral.
 - ❖ Modelo de Montagem de Componentes.
 - ❖ Modelo de Desenvolvimento Concorrente.

O Modelo RAD

- ❖ RAD (*Rapid Application Development*) é um modelo sequencial linear que enfatiza um ciclo de desenvolvimento extremamente **curto**.
- ❖ O desenvolvimento rápido é obtido usando uma **abordagem** de construção baseada em **componentes**.
- ❖ Os requisitos devem ser bem entendidos e o alcance do projeto restrito.
- ❖ O modelo RAD é usado principalmente para aplicações de sistema de informação.
- ❖ Cada função principal pode ser direcionada para uma equipe RAD separada e então integrada para formar o todo.

O Modelo RAD



O Modelo RAD

Desvantagens

- ❖ Exige recursos humanos suficientes para todas as equipes.
- ❖ Exige que desenvolvedores e clientes estejam comprometidos com as atividades de “fogo-rápido” a fim de terminar o projeto num prazo curto.

O Modelo RAD

- ❖ Nem todos os tipos de aplicação são apropriadas para o RAD:
 - ❖ Deve ser possível a modularização efetiva da aplicação.
 - ❖ Se alto desempenho é uma característica e o desempenho é obtido sintonizando as interfaces dos componentes do sistema, a abordagem RAD pode não funcionar.