

Modelos de Processo de Software



Seiji Isotani, Rafaela V. Rocha

sisotani@icmc.usp.br
rafaela.vilela@gmail.com

PAE: Armando M. Toda

armando.toda@gmail.com

(material produzido e atualizado pelos professores do grupo de pesquisa em Engenharia de Software do ICMC-USP)

Engenharia de Software

Uma Tecnologia em Camadas



Gerenciamento da Qualidade Total e filosofias similares produzem uma mudança cultural que permite o desenvolvimento crescente de abordagens mais maduras para a Engenharia de Software

ENGENHARIA DE SOFTWARE

pode ser vista como uma abordagem de desenvolvimento de software elaborada com disciplina e métodos bem definidos.



.....“a construção por múltiplas pessoas de um software de múltiplas versões”

[Parnas 1987]

Processo de software

Conjunto de atividades relacionadas que levam à produção de um produto de software

(Sommerville, 2013)

Incluem:

1. Produtos gerados em cada atividade
2. Papéis envolvidos
3. Pré e pós-condições das atividades

Motivação

- **Processos de software são complexos e dependem de vários fatores como:**
 - **Produto a ser desenvolvido**
 - **Equipe**
 - **Recursos disponíveis**
- **Não existe um processo ideal, ele pode ser adaptado de acordo com o contexto, podendo ser mais formal ou mais flexível.**

Motivação

- **Envolvem criatividade e fatores humanos**
- **Decisões precisam ser tomadas ao longo do processo**
- **Decisões erradas podem levar a prejuízos e perda de oportunidades**

Como obter uma representação simplificada de um processo de software?



Modelos de Processo de Software

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

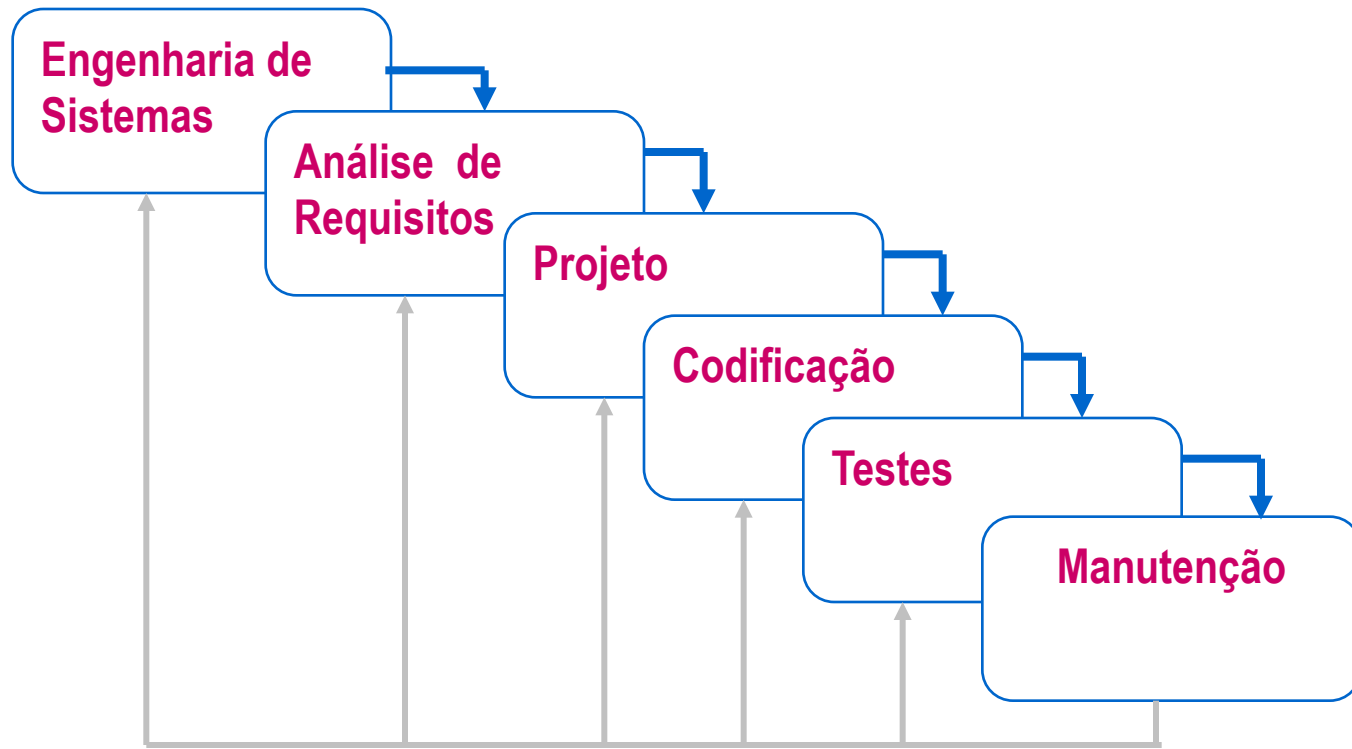
Modelos de Processo de Software

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

O Modelo Cascata

- modelo mais antigo e o mais amplamente usado da engenharia de software
- modelado em função do ciclo da engenharia convencional
- requer uma abordagem sistemática, seqüencial ao desenvolvimento de software
- o resultado de uma fase se constitui na entrada da outra

O Modelo Cascata



O Modelo Cascata

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

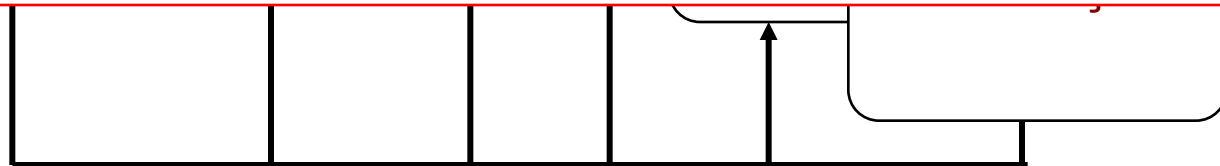
Análise de Requisitos de Software

- 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

Projeto

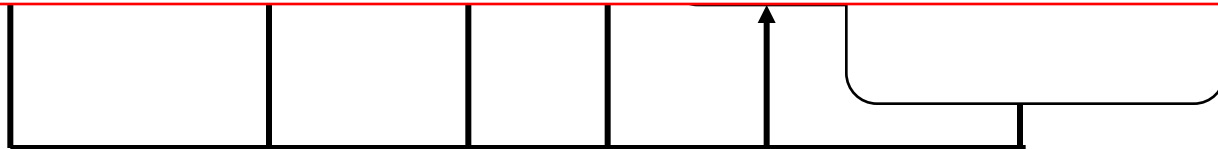
- tradução dos requisitos do software para um conjunto de representações que podem ser avaliadas quanto à qualidade, antes que a codificação se inicie



O Modelo Cascata

Codificação

- tradução das representações do projeto para uma linguagem “artificial” resultando em instruções executáveis pelo computador



O Modelo Cascata

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

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*

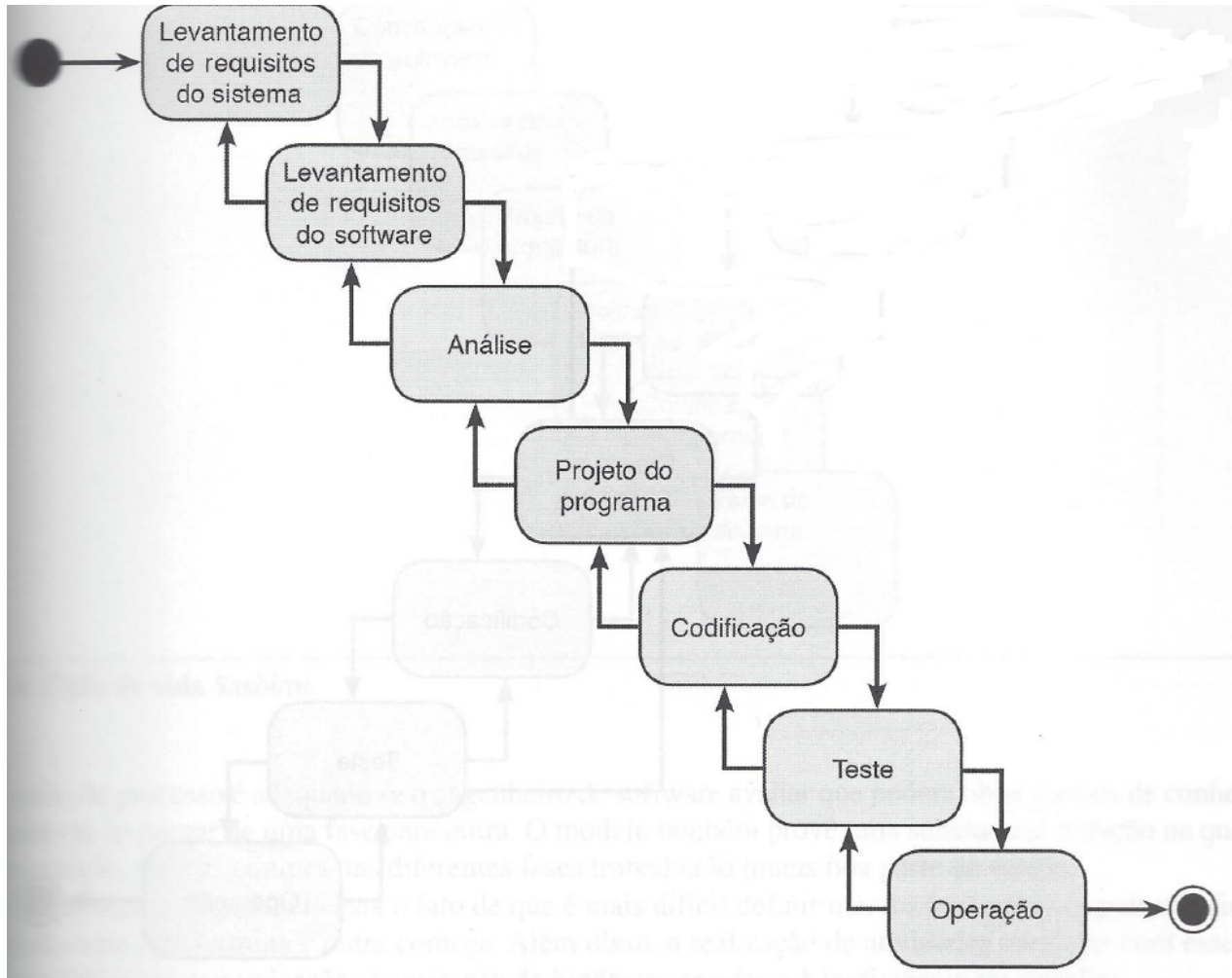
Problemas com o Modelo Cascata

- 💣 Projetos reais raramente seguem o fluxo seqüencial 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

Problemas com o Modelo Cascata

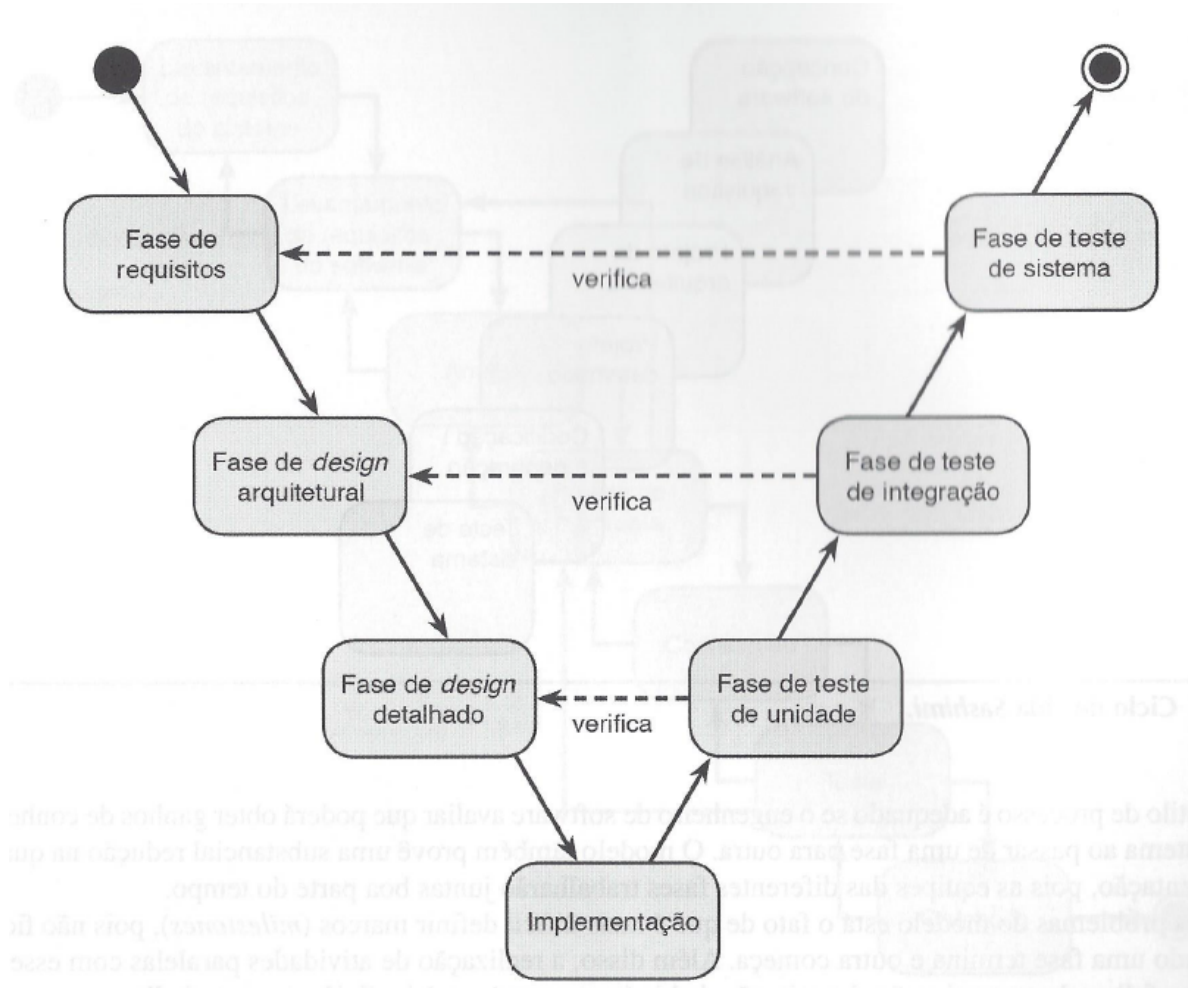
- ✓ *Embora o Modelo Cascata tenha fragilidades, ele é significativamente melhor do que uma abordagem casual ao desenvolvimento de software*

Variações do Modelo Cascata (Wazlawick, 2013)



CASCATA DUPLA (Royce, 1970)

Variações do Modelo Cascata (Wazlawick, 2013)



MODELO V (Spillner, 2002)

Variações do Modelo Cascata (Wazlawick, 2013)

- Outras variações:
 - Sashimi
 - Modelo W
 - Cascata com subprojetos

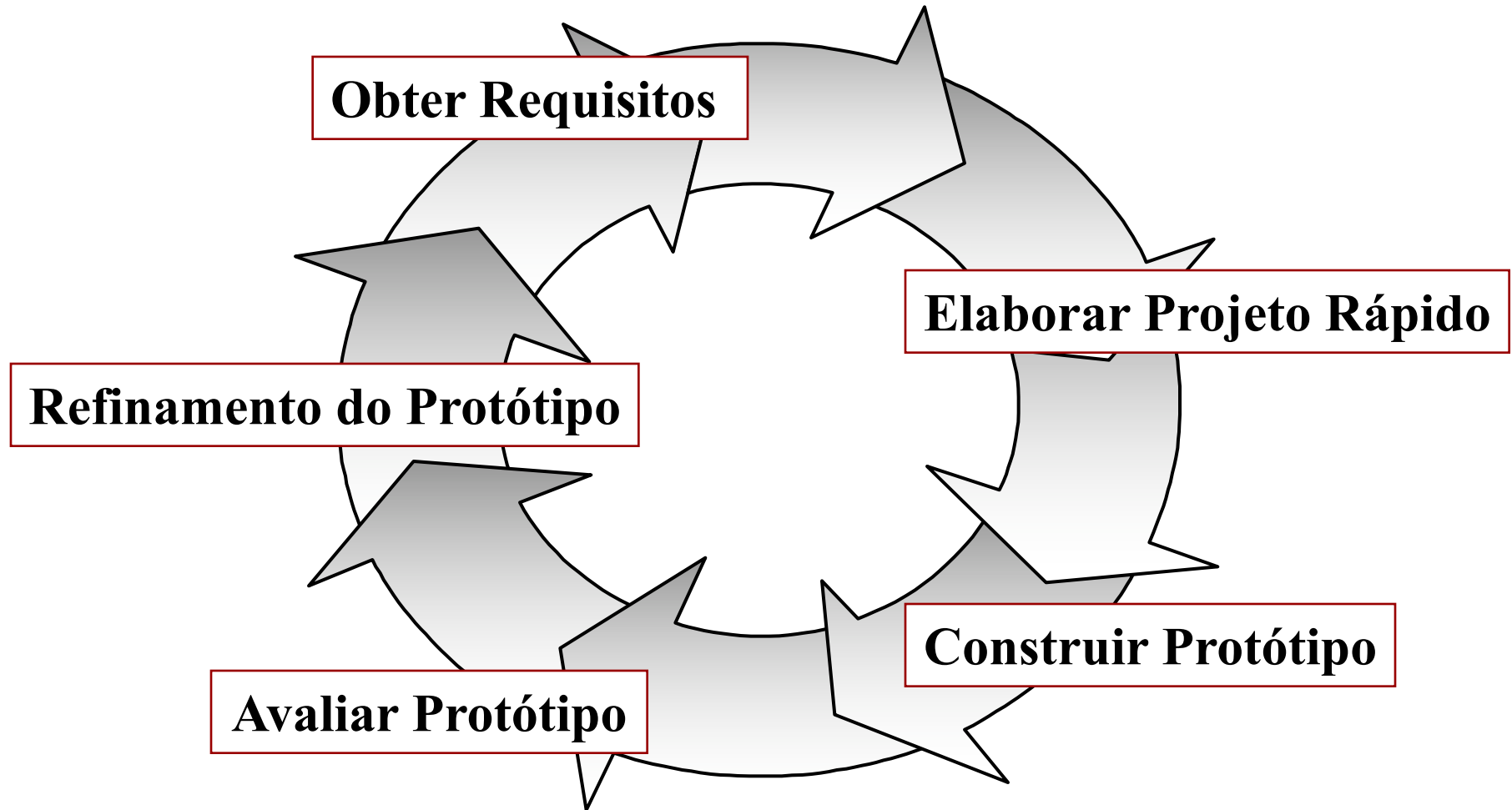
O Modelo Cascata

- 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

O Modelo 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
- apropriado para quando o cliente não definiu detalhadamente os requisitos.

O Paradigma de Prototipação para obtenção dos requisitos



O Paradigma de Prototipação para obtenção dos requisitos

1- 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.

Refinar

o Rápido

Avaliar Protótipo

Construir Protótipo

O Paradigma de Prototipação para obtenção dos requisitos



Obter Requisitos

2- PROJETO RÁPIDO:

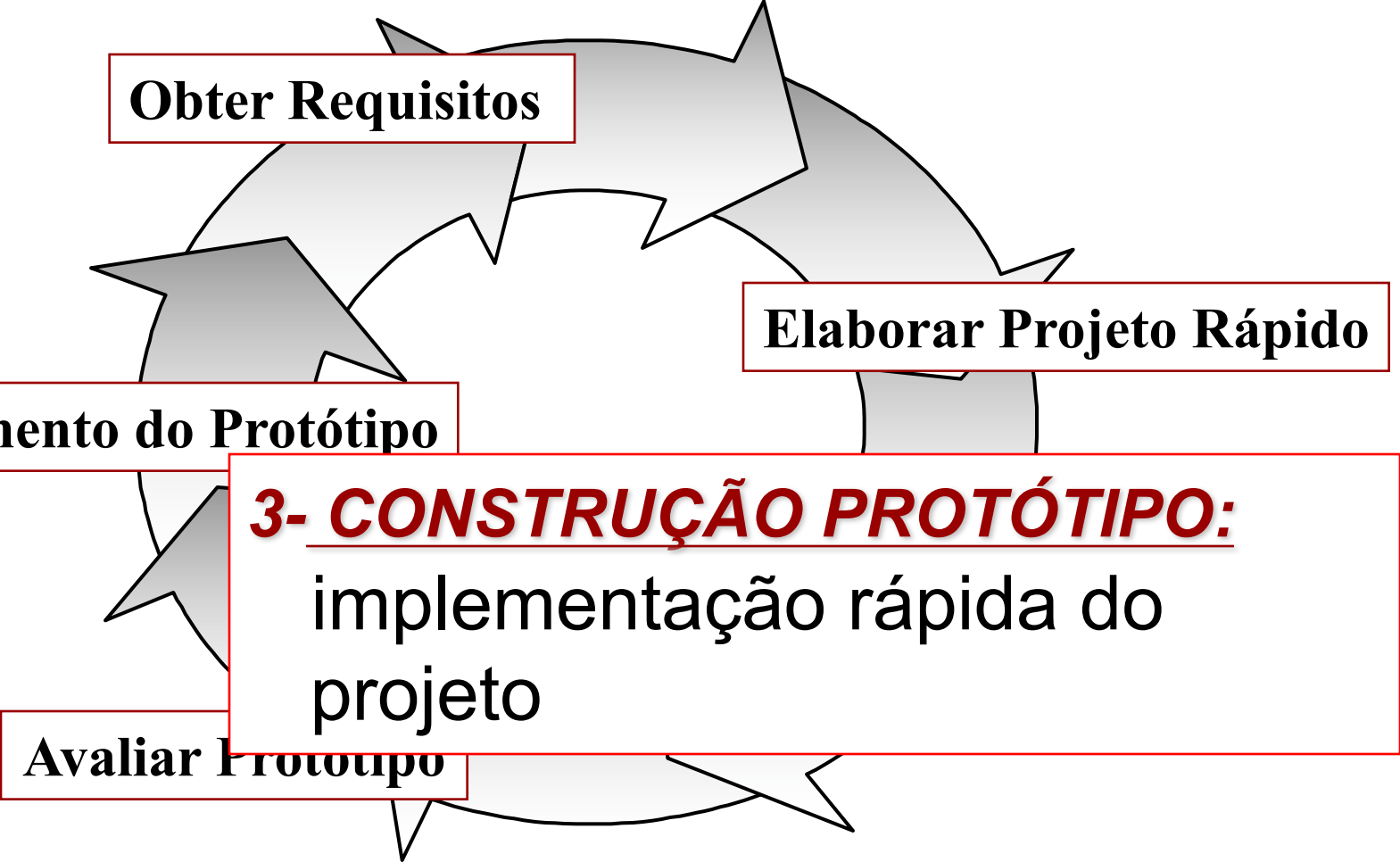
representação dos aspectos do software que são visíveis ao usuário (abordagens de entrada e formatos de saída)

Refinamento do Produto

Construir Protótipo

Avaliar Protótipo

O Paradigma de Prototipação para obtenção dos requisitos



The diagram illustrates the Prototyping Paradigm as a continuous cycle. It features four main steps, each in a rectangular box: 'Obter Requisitos' (top left), 'Elaborar Projeto Rápido' (top right), 'Avaliar Protótipo' (bottom left), and 'Refinamento do Protótipo' (bottom right). These steps are connected by a large, thick, grey arrow that forms a clockwise loop. A central text box, outlined in red, contains the title '3- CONSTRUÇÃO PROTÓTIPO:' followed by the description 'implementação rápida do projeto'.

Obter Requisitos

Elaborar Projeto Rápido

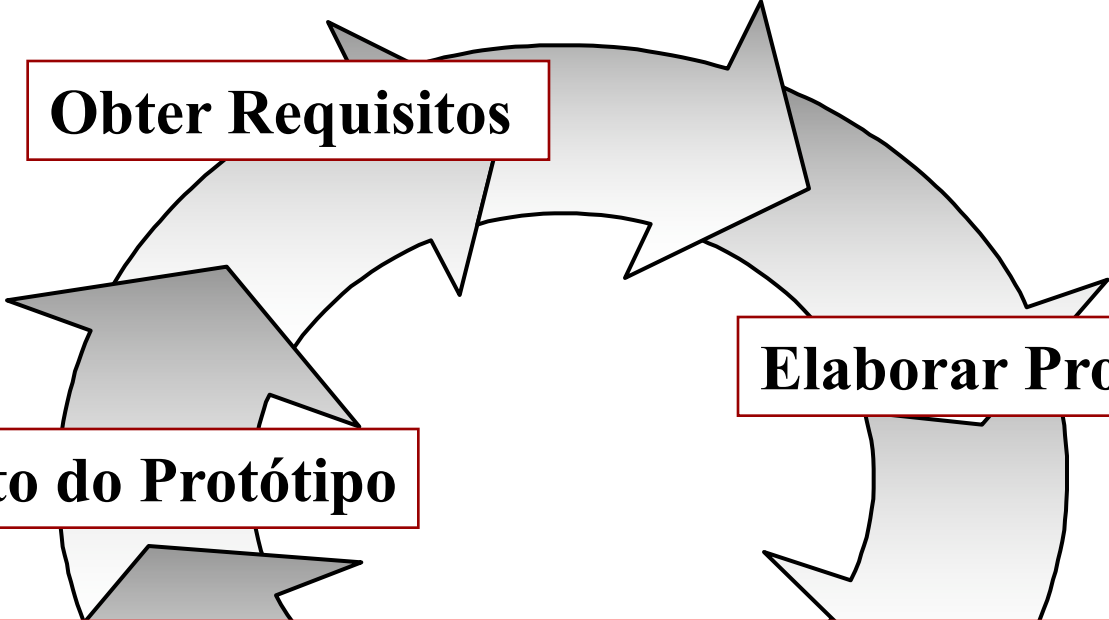
Refinamento do Protótipo

3- CONSTRUÇÃO PROTÓTIPO:

implementação rápida do
projeto

Avaliar Protótipo

O Paradigma de Prototipação para obtenção dos requisitos



The diagram illustrates the Prototyping Paradigm as a continuous cycle. It features three large, light-gray, jagged arrows forming a clockwise loop. The first arrow at the top points right and contains the text 'Obter Requisitos'. The second arrow on the right points down and contains the text 'Elaborar Projeto Rápido'. The third arrow on the left points up and contains the text 'Refinamento do Protótipo'. Below this cycle is a large rectangular box containing the title '4- AVALIAÇÃO DO PROTÓTIPO:' and a description of the evaluation step.

Obter Requisitos

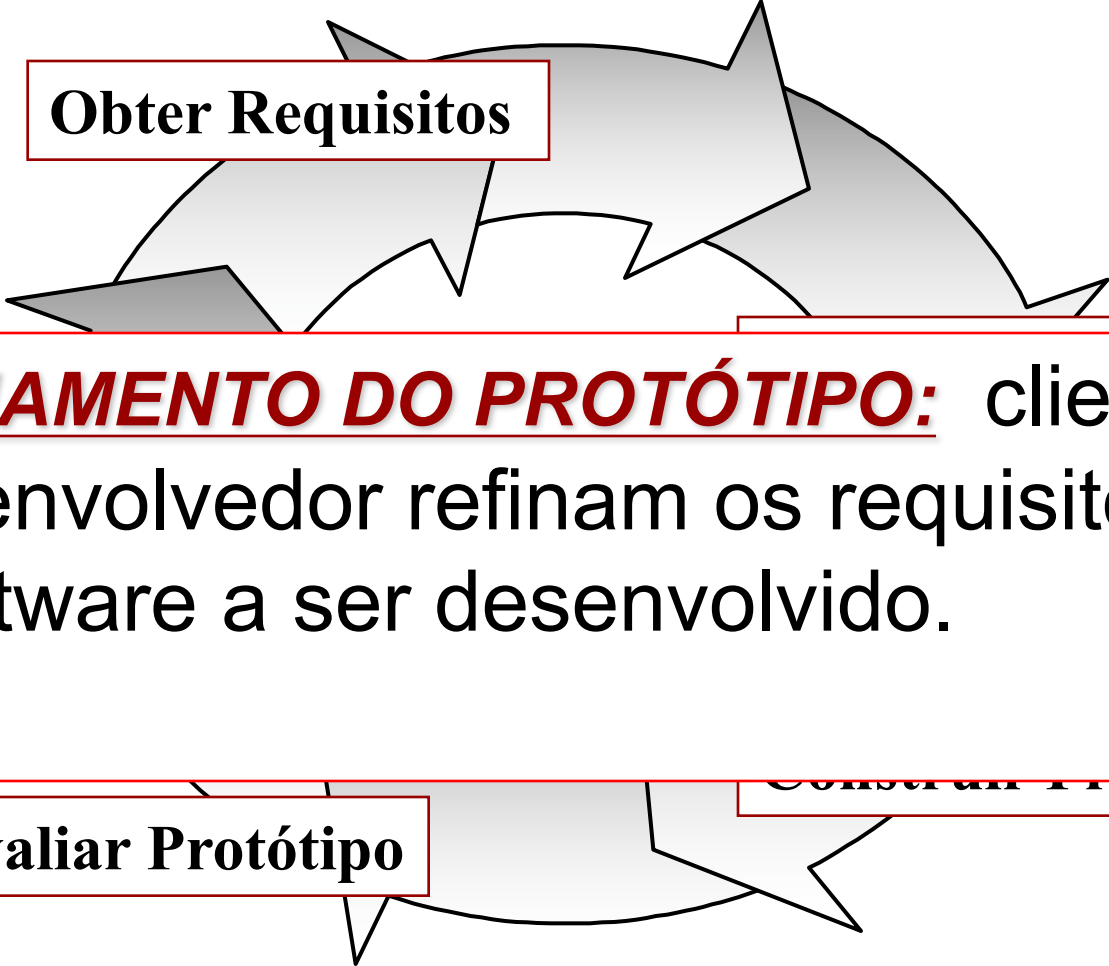
Elaborar Projeto Rápido

Refinamento do Protótipo

4- AVALIAÇÃO DO PROTÓTIPO:

cliente e desenvolvedor avaliam o protótipo

O Paradigma de Prototipação para obtenção dos requisitos



Obter Requisitos

The diagram illustrates the Prototyping Paradigm as a continuous cycle. It features a large, light-gray, jagged shape in the background. Overlaid on this are three rectangular boxes with red borders, each containing a step in the process. The first box, 'Obter Requisitos', is at the top. The second box, '5- REFINAMENTO DO PROTÓTIPO: cliente e desenvolvedor refinam os requisitos do software a ser desenvolvido.', is in the center and is the largest. The third box, 'Avaliar Protótipo', is at the bottom. The boxes are connected by a series of curved lines, suggesting a flow or cycle. The text 'Conservar Protótipo' is partially visible on the right side of the diagram.

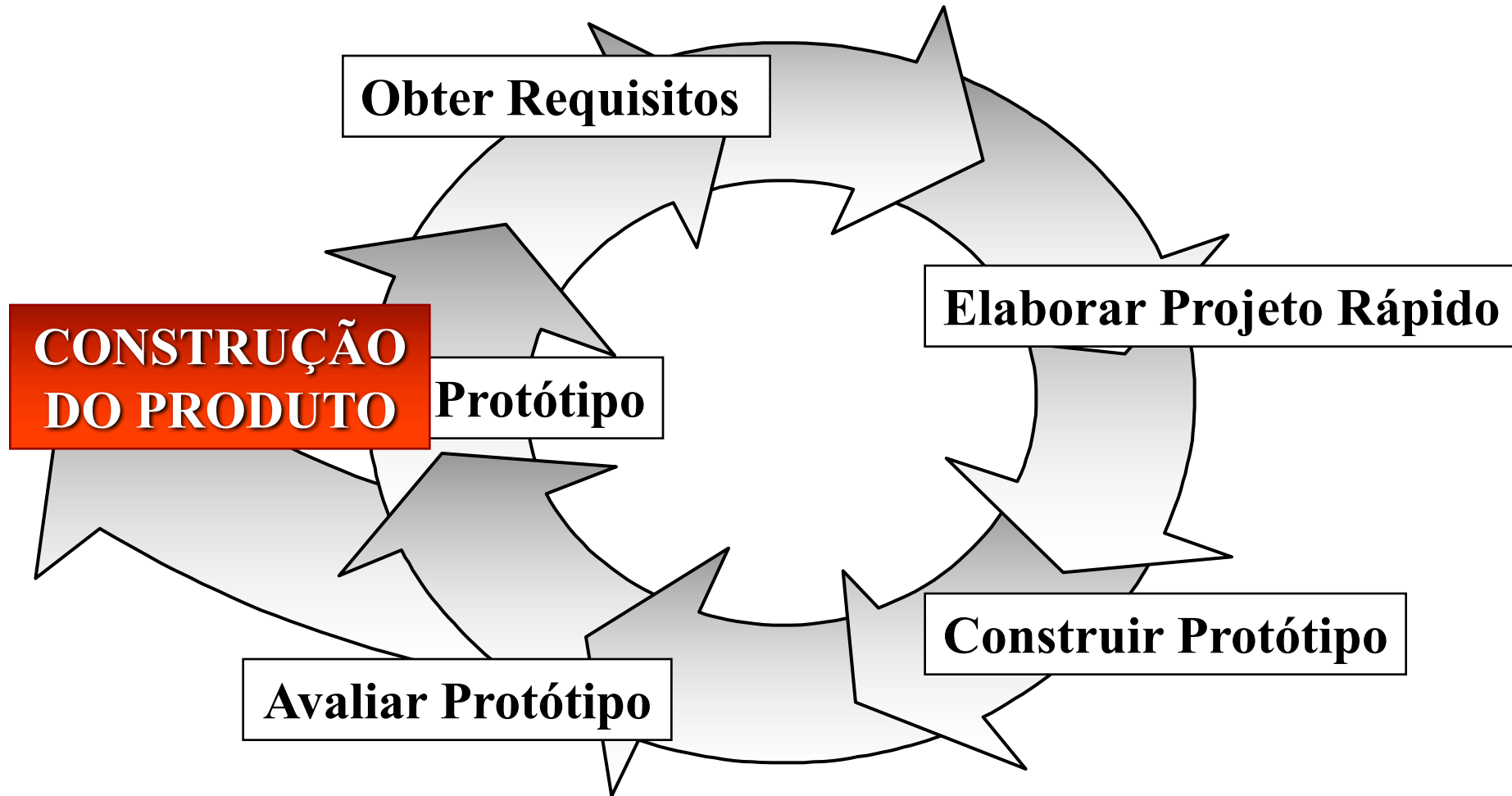
5- REFINAMENTO DO PROTÓTIPO: cliente e desenvolvedor refinam os requisitos do software a ser desenvolvido.

Rápido

Avaliar Protótipo

Conservar Protótipo

O Paradigma de Prototipação para obtenção dos requisitos



O Paradigma de Prototipação para obtenção dos requisitos

Obter Requisitos

6- CONSTRUÇÃO PRODUTO:

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

Realizar Projeto Rápido

Construir Protótipo

Avaliar Protótipo

Problemas com a Prototipação

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

Comentários sobre o Paradigma de Prototipação

- 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

Comentários sobre o Paradigma de Prototipação

- Prototipação descartável X Prototipação evolutiva (ou evolucionária)

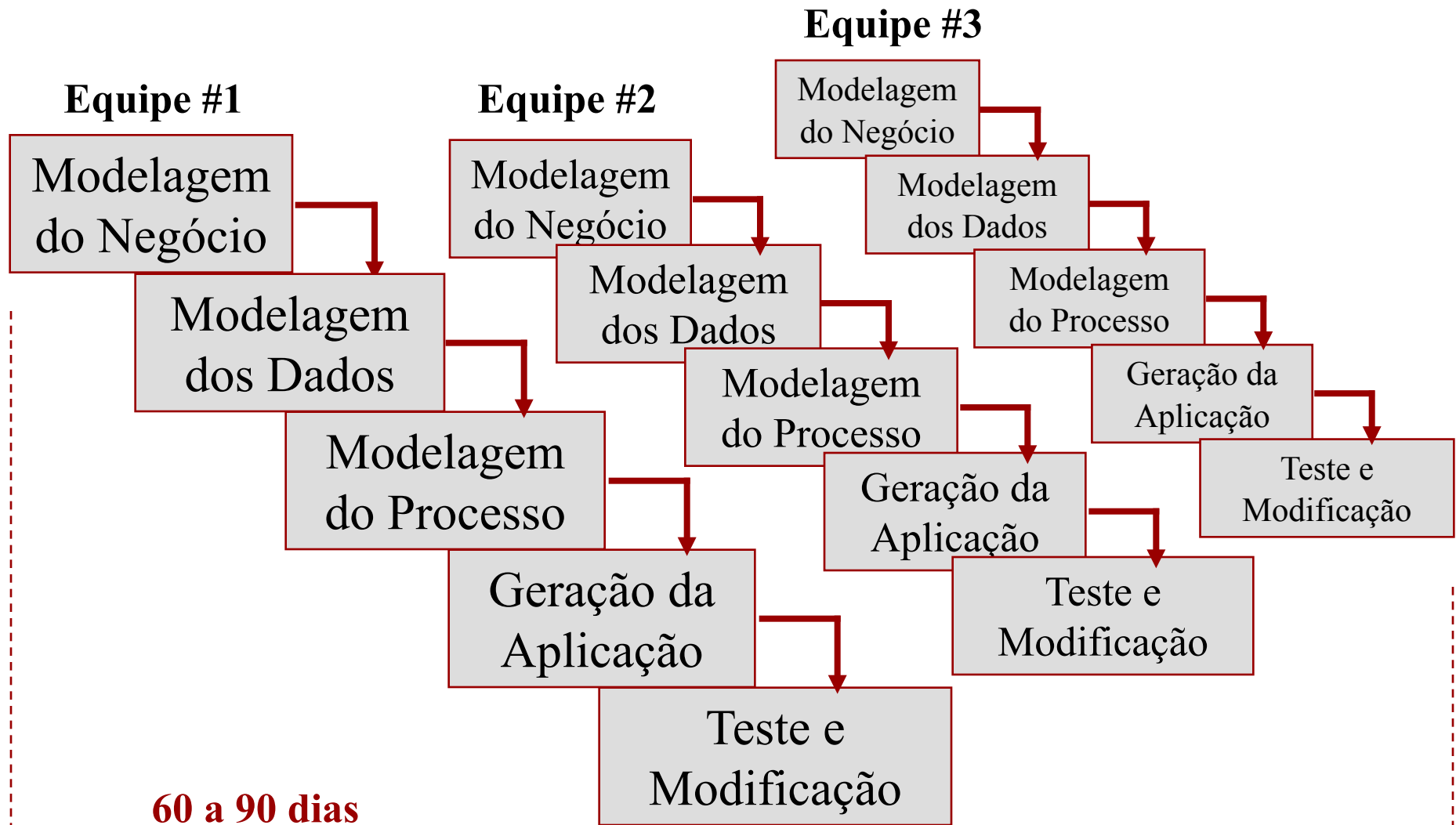
O Modelo RAD

- **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**.

O Modelo RAD

- 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

Modelos de Processo de Software

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

Atividade

Exercício 2.1, 2.3, 2.7, 2.8 adaptados.

Ler capítulo 2 do Pressman

Exercícios sobre o capítulo no início da próxima aula.

