Modelos de Processo de Software

Prof. André Takeshi Endo

UTFPR
UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
CAMPUS CORNÉLIO PROCÓPIO

Modelo de Processo

- Software é importante
  - Consequências boas e ruins
- Não é possível desenvolver software de maneira ad-hoc
- Definir uma série de passos previsíveis
- Roteiro que ajude a criar um
  - **Resultado de alta qualidade**
  - **Dentro do prazo estabelecido**

Modelo de Processo

- Processo de software
  - Uma metodologia para as atividades, ações e tarefas necessárias para desenvolver software de alta qualidade.
  - Define a abordagem adotada para a aplicação da engenharia de software
- Propicia
  - Estabilidade, controle, e organização

Modelo de Processo

- Processo de software
  - **Quem** faz **o que**
  - **Quando**
  - **Como**

Modelo de Processo

- O que acontece quando não temos um processo de software?
  - Ilustração da Internet

Modelo de Processo

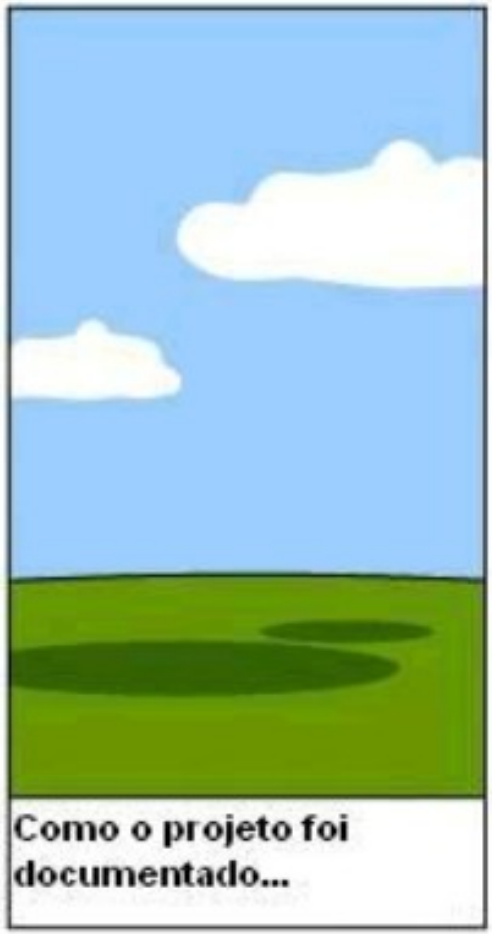
Modelo de Processo

Modelo de Processo

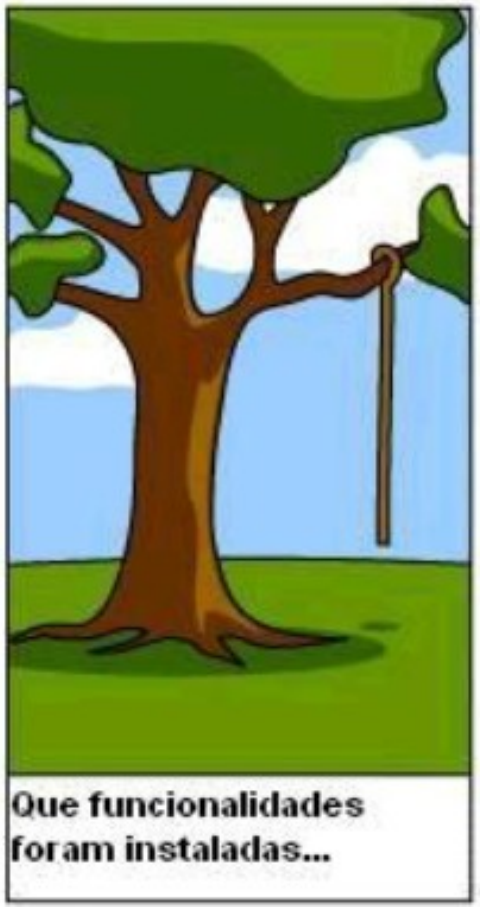
Modelo de Processo

Modelo de Processo

Modelo de Processo

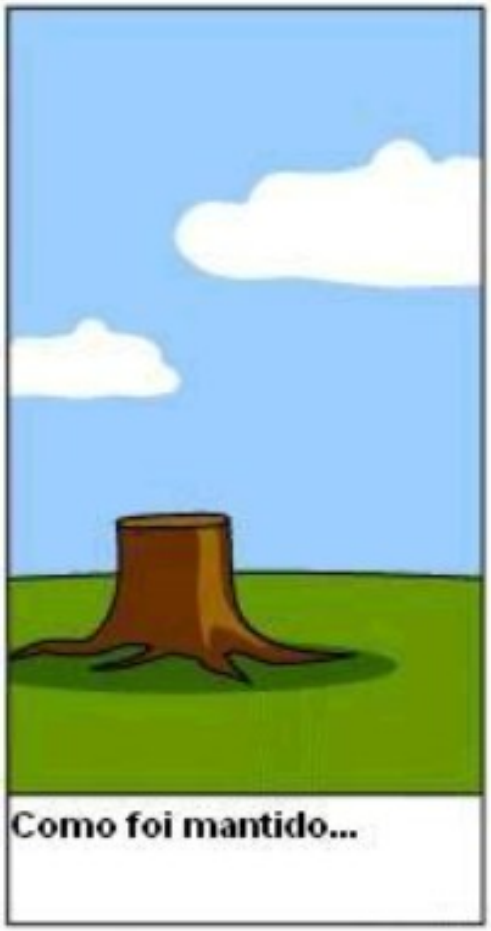


Modelo de Processo



Modelo de Processo

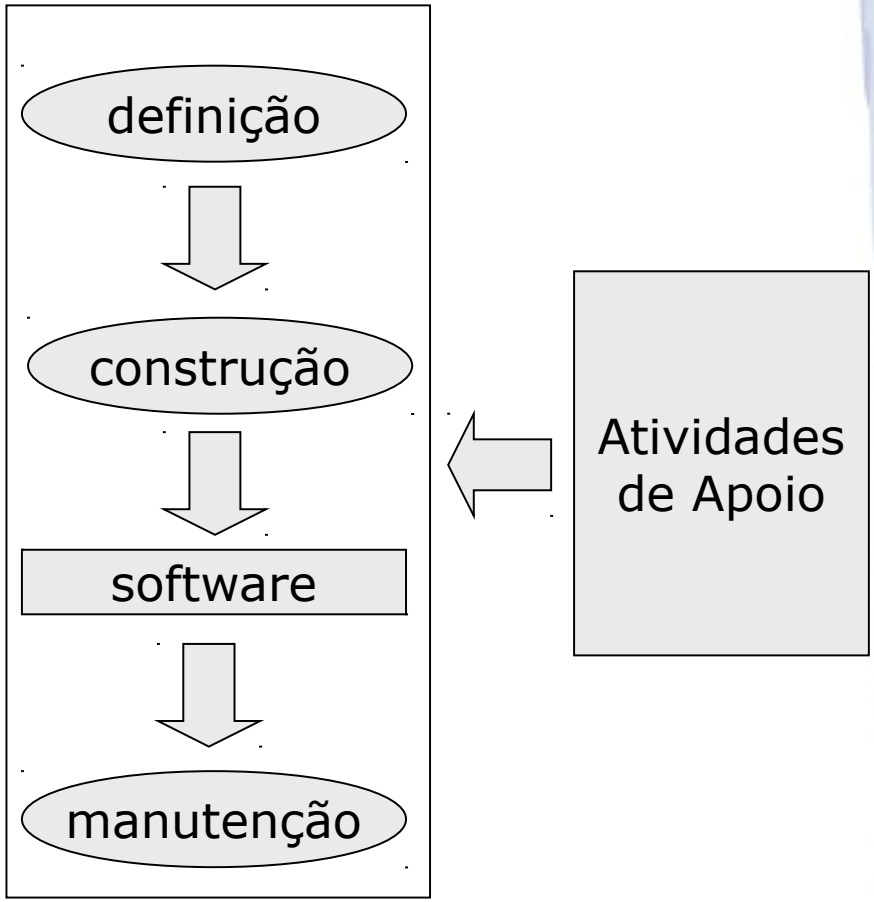
Modelo de Processo



Modelo de Processo

Modelo de Processo

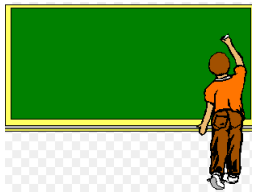
- Fases genéricas
  - Definição (o que?)
  - Construção (como?)
    - Software
  - Manutenção
- Atividades de apoio



Modelo de Processo

- Um modelo de processo representa uma tentativa de colocar ordem em uma atividade inerentemente caótica
- Por que modelo?
  - Não é um processo específico
- Modelos prescritivos (tradicionais)
  - Atividades, ações de ES, tarefas, mecanismos de controle e gerenciamento
  - Fluxo de processo (fluxo de trabalho)

Modelos de Processo de Software

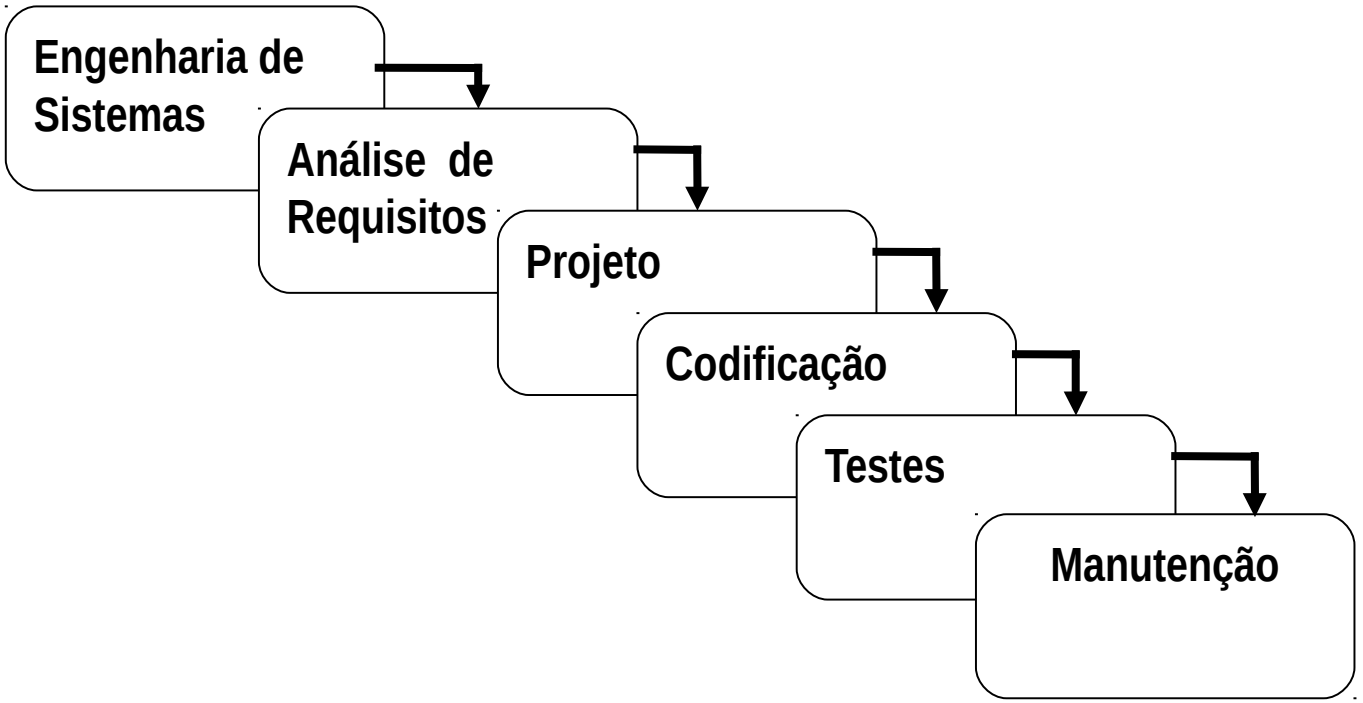


Modelo Cascata

- Modelo mais antigo
- Amplamente conhecido
- Baseado na engenharia tradicional
- Sequencial e linear

Modelo Cascata

- Representação (elementos e fluxo)



Modelo Cascata (-)

- Fluxo linear!
- Requisitos no início (incerteza)
- Cliente paciente

Modelo Cascata (+)

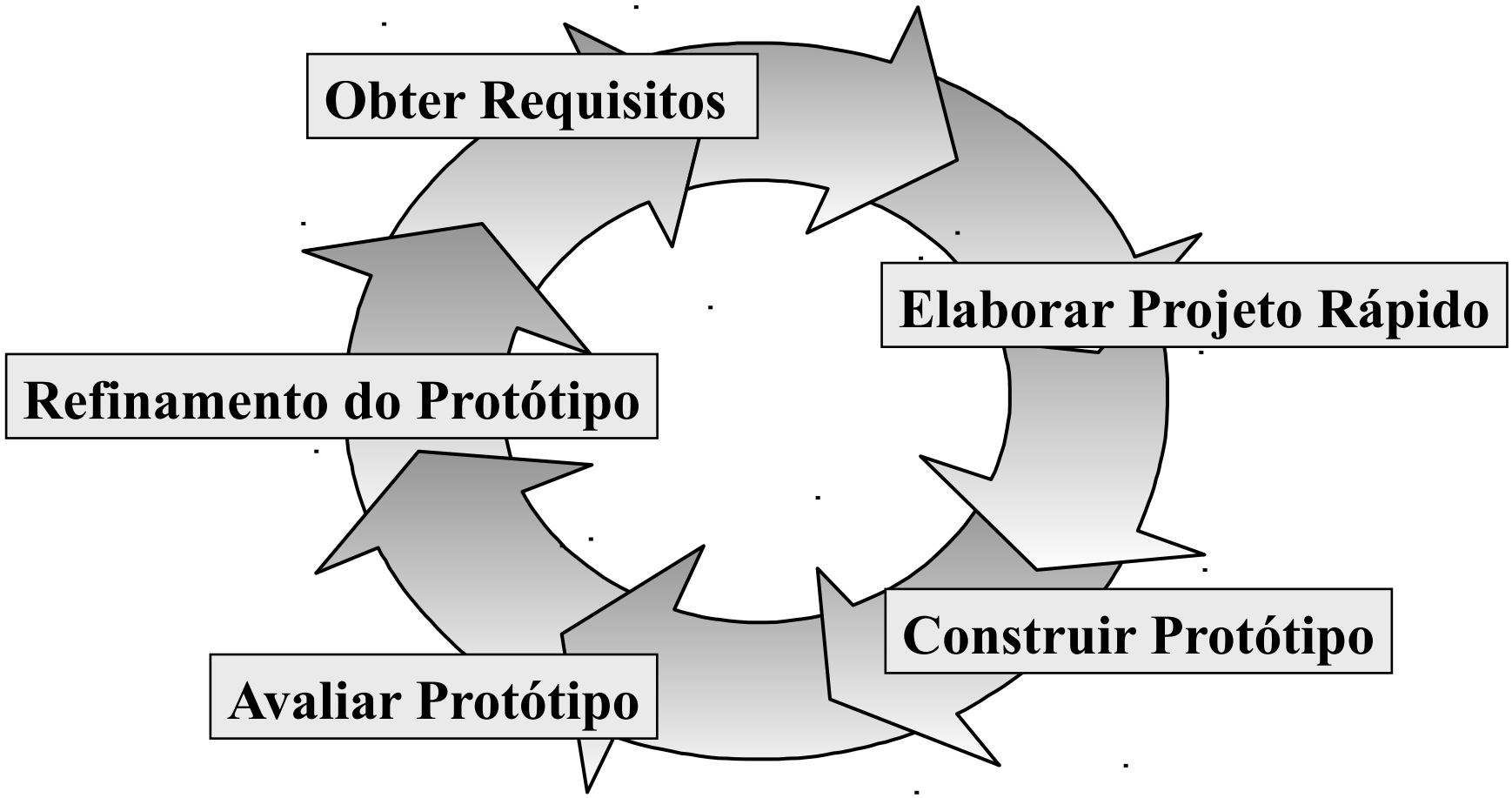
- É melhor do que uma abordagem ad-hoc
- Contribuições históricas
  - Impôs disciplina, planejamento e gerenciamento
  - A implementação do produto é postergada até que todos os requisitos estejam definidos

Prototipação

- Entender melhor os requisitos
- Definir melhor os requisitos
- Trabalho com o conceito de protótipos
- O cliente não sabe o que quer

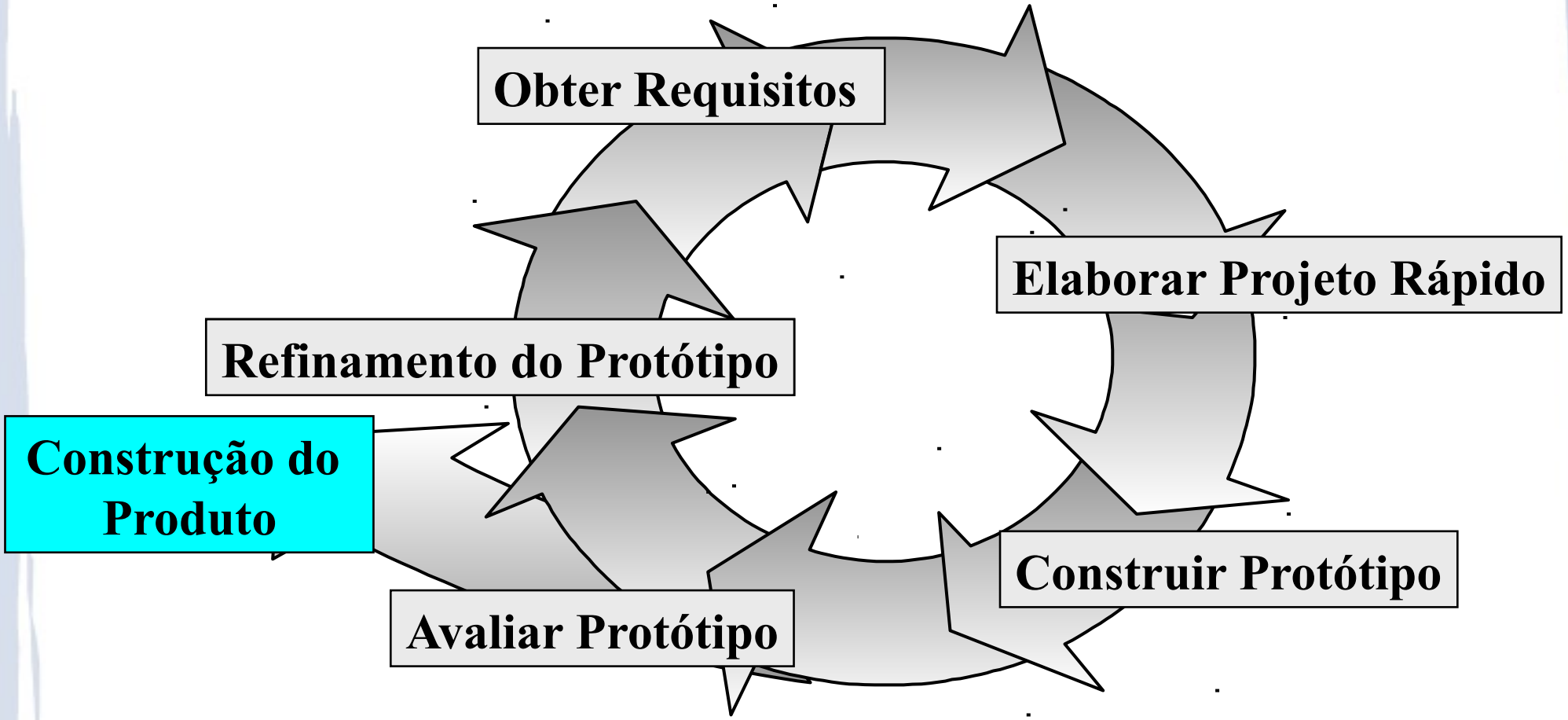
Prototipação

- Representação (elementos e fluxo)



Prototipação

- Representação (elementos e fluxo)



Prototipação (-)

- Cliente não entende o protótipo
- Decisões ruins são tomadas para desenvolver o protótipo rapidamente
  - Incorporado no processo!
- Protótipo → Produto

Prototipação (+)

- Pode ser eficiente
- Definindo as regras do jogo inicialmente
- “Protótipo será usado apenas para definir os requisitos”

Cliente

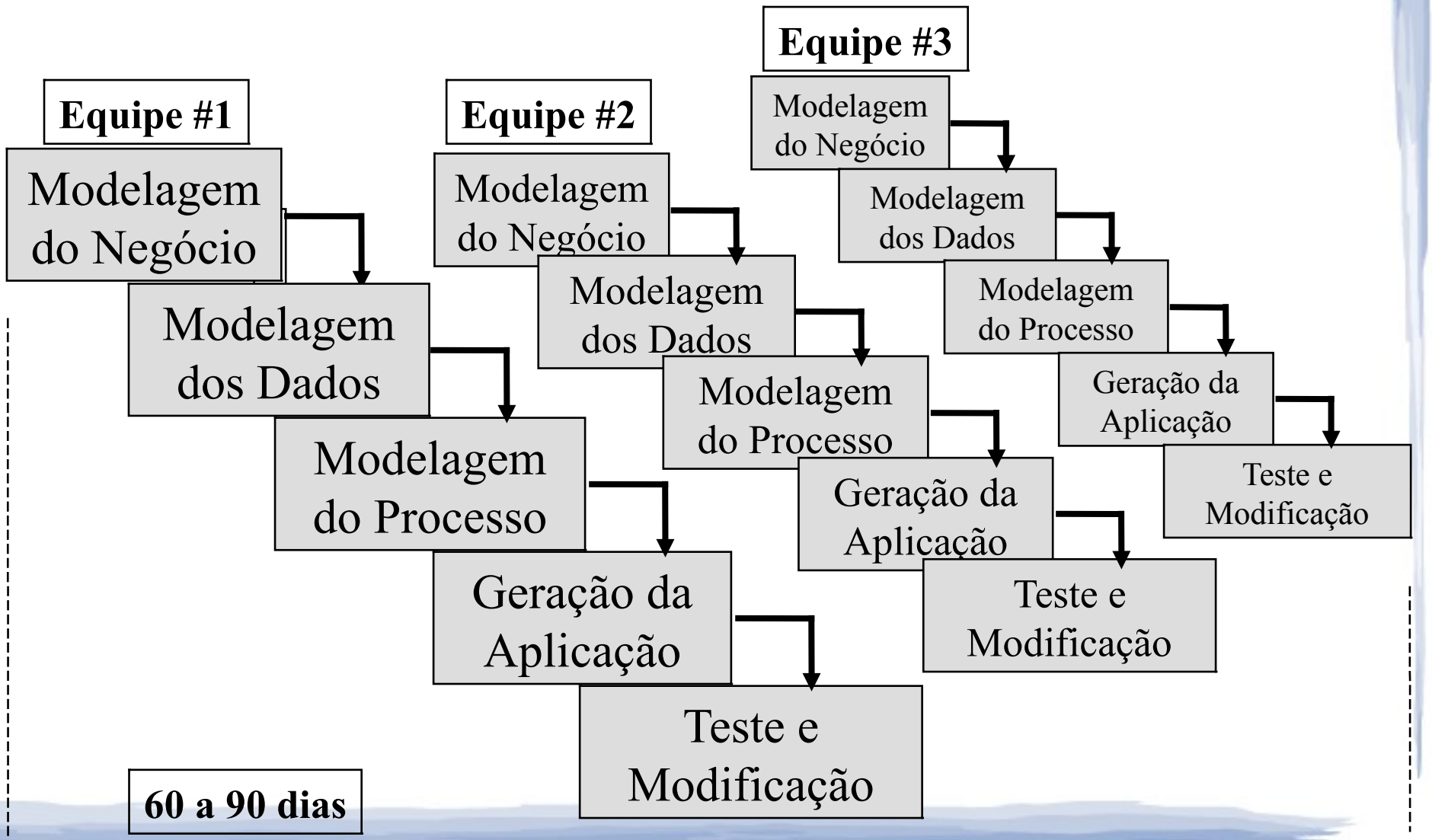
Desenvolvedor

Modelo RAD

- *Rapid Application Development* (RAD)
- Modelo sequencial linear
- Ciclo de desenvolvimento extremamente curto
- Rápido → abordagem de construção baseada em componentes
- Requisitos claros, projeto restrito
- Sistemas de informação
- Divide funções em equipes

Modelo RAD

- Representação (elementos e fluxo)



Modelo RAD (-)

- Recursos humanos
- Clientes e desenvolvedores comprometidos
- “A toque de caixa”
- Pode não ser adequado a determinadas aplicações
  - Modular

Modelo RAD (+)

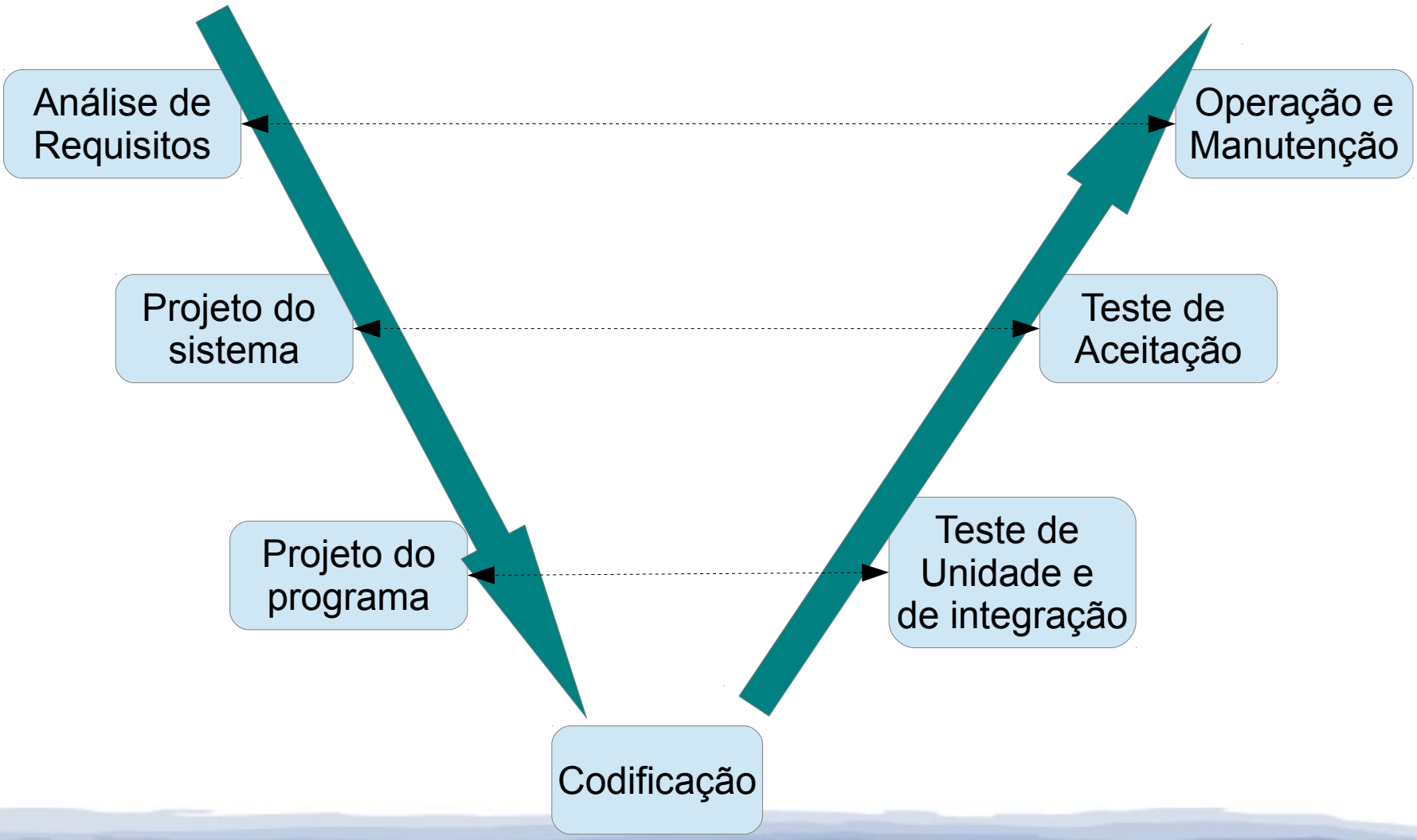
- Efetiva com sistemas modulares
- Equipes não precisam estar fisicamente juntas

Modelo em V

- Uma variação do cascata
- V para “Verificação” e “Validação”
- Preocupação com a garantia de qualidade de software

Modelo em V

- Representação (elementos e fluxo)



Modelo em V (-)

- Falsa sensação de segurança
- Problemas do cascata
- Limita a criatividade dos testes

Modelo em V (+)

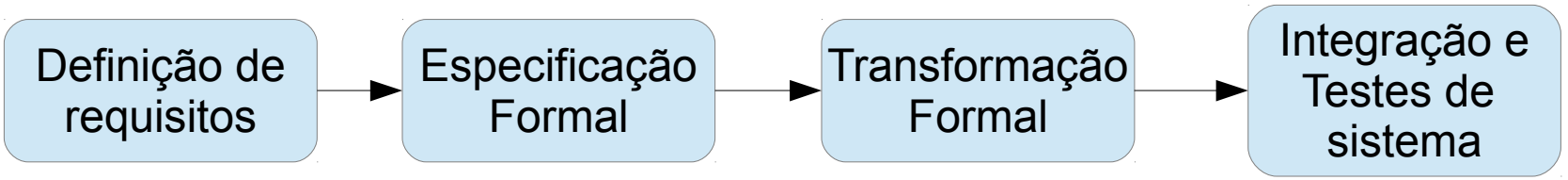
- É de fácil compreensão
- Dá importância a atividades de garantia de qualidade de software

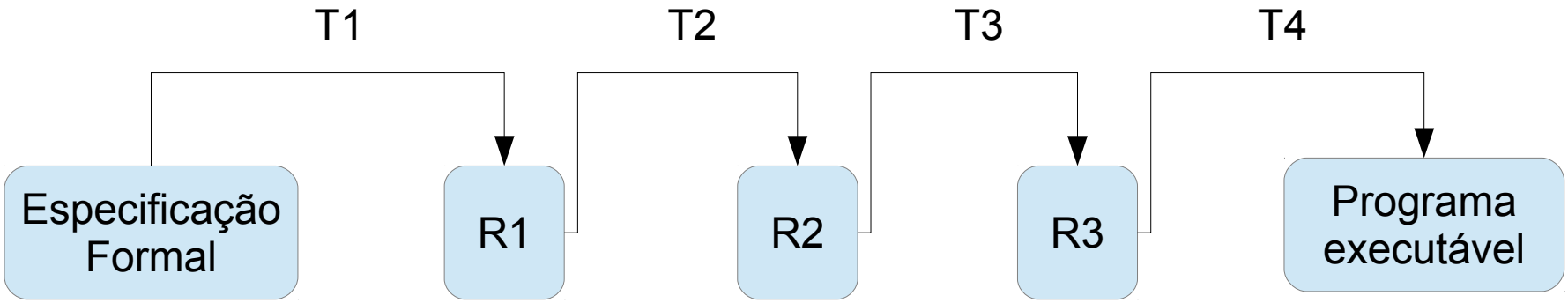
Desenvolvimento Formal

- Modelo de métodos formais
- Especificação matemática formal do software
- Elimina ambigüidade, incompletude e inconsistência (análise matemática)
- Sistemas aviônicos e equipamentos médicos

Desenvolvimento Formal

- Representação (elementos e fluxo)





Desenvolvimento Formal (-)

- Restrito a certos domínios
- Especializado
- Transformações podem se complexas

Example: Banking System

WithdrawMoney

ΔBankAccount
dollarAmount? : $\mathbb{N}$
centAmount? : $\mathbb{N}$

---

dollarAmount? $\leq$ dollars
dollarAmount? = dollars $\Rightarrow$ centAmount? $\leq$ cents
centAmount? > cents
$\Rightarrow$ (dollars' = dollars – dollarAmount? – 1
$\wedge$ cents' = cents – centAmount? + 100)
centAmount? $\leq$ cents
$\Rightarrow$ (dollars' = dollars – dollarAmount?
$\wedge$ cents' = cents – centAmount?)

Desenvolvimento Formal (+)

- Mapeamento formal da especificação até o programa
- Uso com sucesso em sistemas críticos (segurança, confiabilidade)
- Model checking
  - Verificação formal automatizada
- *Cleanroom software engineering*
  - IBM

Influências

- Modelos de processo (até agora)
  - Cascata
  - Prototipação
  - RAD
  - Modelo em V
  - Desenvolvimento formal
- **Modelos de processo evolutivos**
  - *Rational Unified Process* (RUP)
- Métodos ágeis

Bibliografia

- [Pfleeger07] S. L. Pfleeger, “Engenharia de Software: Teoria e Prática”, 2007.
- [Pressman11] R. S. Pressman, “Engenharia de Software: uma abordagem profissional”, 2011.
- [Sommerville03] I. Sommerville, “Engenharia de Software”, 2003.
- [Brooks87] “No Silver Bullet: Essence and Accidents of Software Engineering”, 1987.
  http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1663532
- [IEEE90] “IEEE Standard Glossary of Software Engineering Terminology”, 1990.
  http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=159342

Bibliografia

- [UUU] Materiais didáticos elaborados pelos grupos de engenharia de software do ICMC-USP, DC-UFSCAR e UTFPR-CP.