PECE – POLI – USP
MBA – Tecnologia de Software

Aula 04
Modelos de Processos de Software

Prof. Dr. Rogério Rossi
2021

Roteiro

- Conceitos e Tipos de Processos
- Fluxos de Processos
- Modelos de Processos Descritivos e Prescritivos
- Modelos Prescritivos
  - Cascata
  - Evolucionário
  - Incremental
  - Processo Unificado

Conceitos e Tipos de Processos

Processo é uma sequência de passos realizados para um determinado propósito.

(IEEE Standard STD-601.12, 1990)

Processo é um conjunto de recursos e atividades inter-relacionados que transformam insumos em produtos.

(ISO 8.402)

Conceitos e Tipos de Processos

Processo de software é um conjunto de atividades, métodos, práticas e transformações que as pessoas utilizam para desenvolver e manter software e produtos relacionados.

(SEI/CMMI)

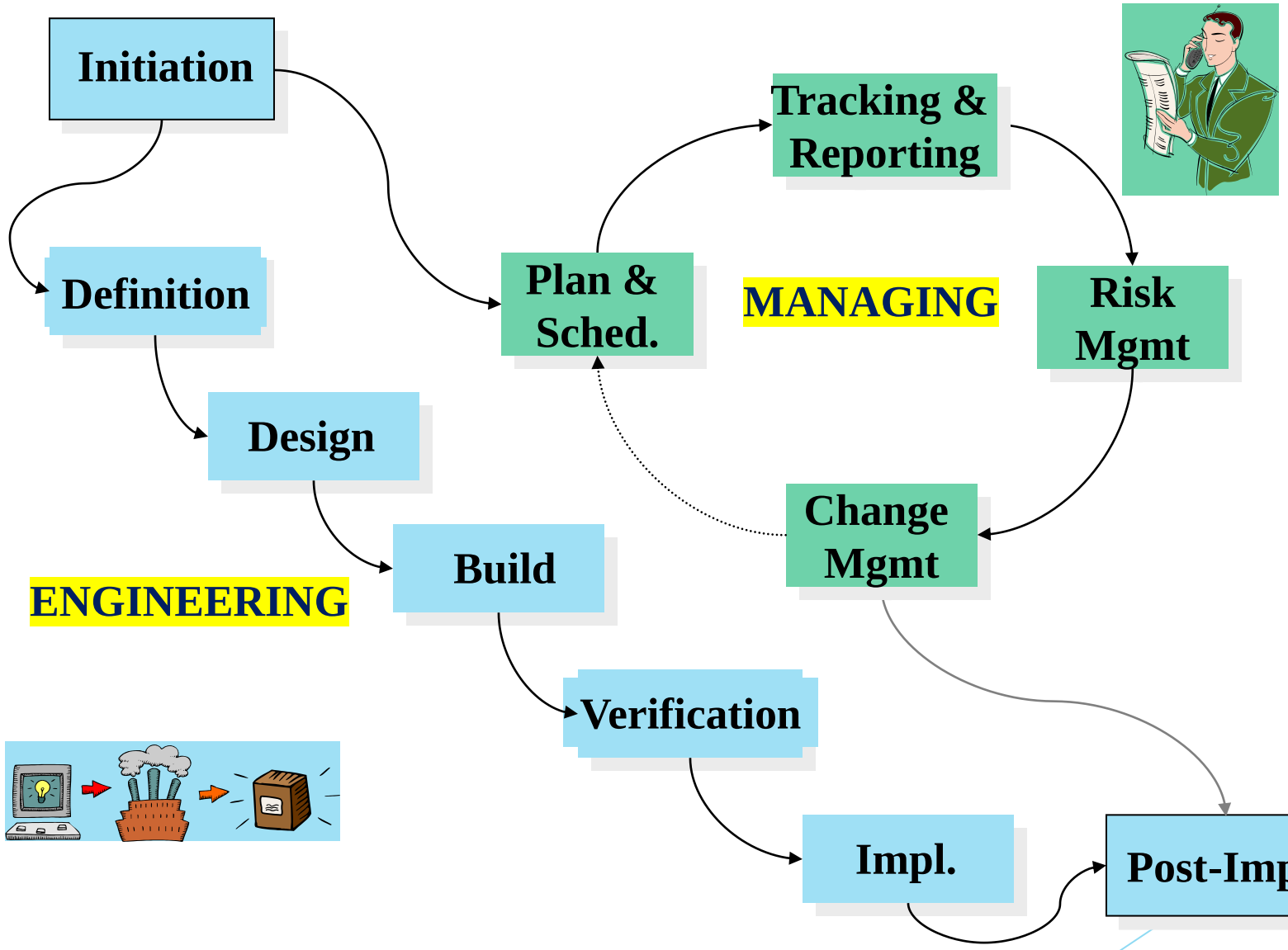
Conceitos e Tipos de Processos

Processos de Software

- **Processos de Engenharia** – construção e manutenção do produto de software
- **Processos de Gerenciamento** – estimativas, planejamento e controle dos recursos necessários para o desenvolvimento do software

(Acuña et al., 2001)

Conceitos e Tipos de Processos



Questão

Como devem ser integrados os processos de gerenciamento e de desenvolvimento do software?

FLUXOS DE PROCESSOS

Fluxos de Processos

Linear

realiza as atividades de forma sequencial considerando a dependência entre elas

Cíclico

executa as atividades de uma forma circular

Paralelo

executa uma ou mais atividades em paralelo com outras atividades

Iterativo

repete uma ou mais atividades antes de seguir para a seguinte

(Pressman , 2011), (Acuña et al., 2001)

MODELOS DESCRITIVOS E PRESCRITIVOS

Modelos de Processos Descritivos e Prescritivos

Modelos de Processos Descritivos (*Descriptive Models*)

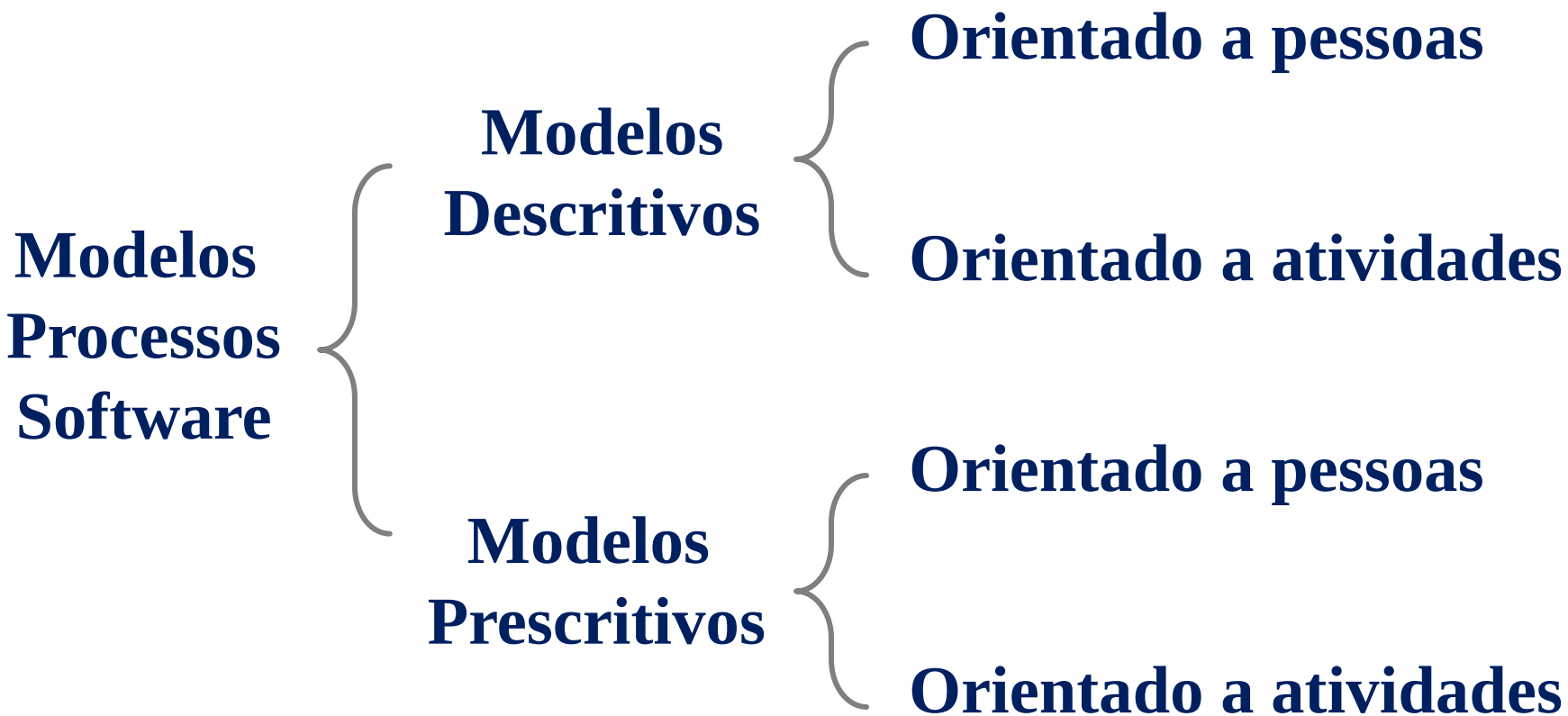
não determinam o que fazer - buscam representar o que está sendo feito

Modelos de Processos Prescritivos (*Prescriptive Models*)

representam o que fazer e em qual ordem - e quais resultados deverão produzir

(Acuña et al., 2001); (Acuña et al., 2005)

Modelos de Processos Descritivos e Prescritivos



(Acuña et al., 2001); (Acuña et al., 2005)

Modelos de Processos Descritivos e Prescritivos

Orientação dos Modelos Descritivos e Prescritivos

Modelos orientados a atividades (*Activity-Oriented Models*)

Concentre-se em definir as funções, atividades e artefatos do gerenciamento, desenvolvimento e suporte de processos de software.

Modelos orientados a pessoas (*People-Oriented Models*)

Concentre-se em definir as pessoas envolvidas no processo de software e seus relacionamentos.

(Acuña et al., 2001); (Acuña et al., 2005)

Modelos de Processos Descritivos e Prescritivos

Exemplos de Modelos de Processos Descritivos

- Agile Methodologies
- SCRUM (*People-Oriented*)
- DSDM - Dynamic System Development Method (*People Oriented*)

(Acuña et al., 2001); (Acuña et al., 2005)

Modelos de Processos Descritivos e Prescritivos

Exemplos de Modelos de Processos Prescritivos:

Manuais

padrões, metodologias e métodos centrados no gerenciamento e produção de software

Automatizados

assistivo, suporte, *computer-assisted support techniques*.

(Pressman , 2011), (Acuña et al., 2001)

Modelos de Processos Prescritivos

- Denominam-se **prescritivos** porque **prescrevem um conjunto de elementos do processo** – atividades, métodos, ações de engenharia, tarefas, produtos de trabalho (artefatos), e mecanismos de controle de mudanças para cada projeto.

- Acomodam as **atividades metodológias** por meio de fluxos de processos (também denominados fluxos de trabalho).

(Pressman , 2011)

Modelos de Processos Prescritivos

Modelos de Processos Prescritivos (*Activity-Oriented*):

- **Cascata (*Waterfall*)** . . . . . estrutura linear
- **Evolucionário.** . . . . . .estrutura combinada (cíclica e iterativa)
- **Incremental** . . . . . .estrutura combinada (linear e iterativa)

(Pressman , 2011), (Acuña et al., 2001)

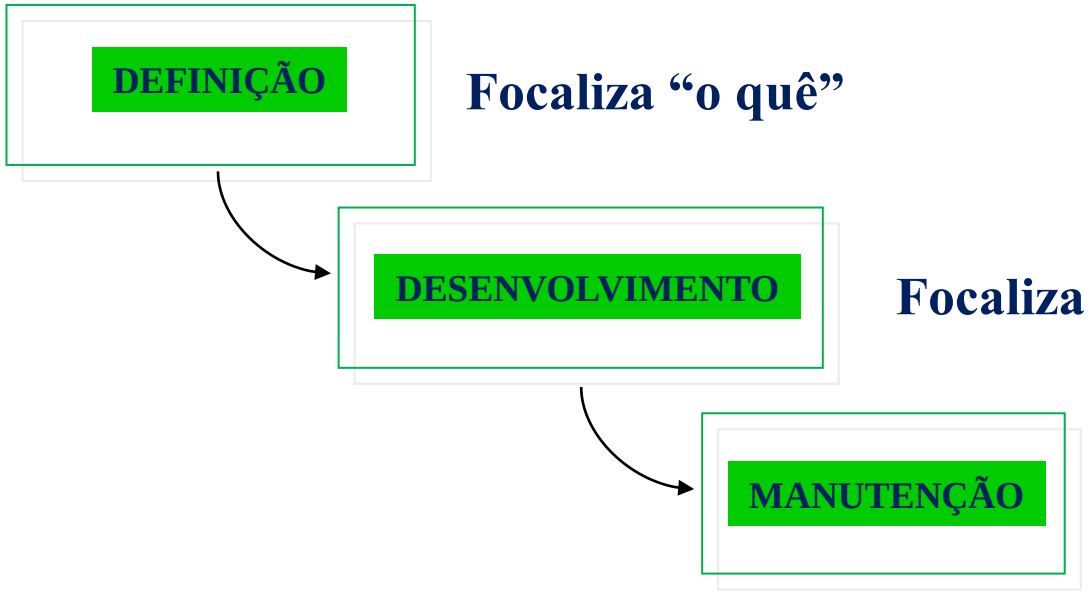
Modelos de Processos Prescritivos

Modelo Cascata (*Waterfall*)

- Requisitos razoavelmente estáveis
- Abordagem sequencial e sistemática para o desenvolvimento de software
- Sequencia rígida de atividades
- Usuário/cliente envolvidos somente no início e no fim do processo

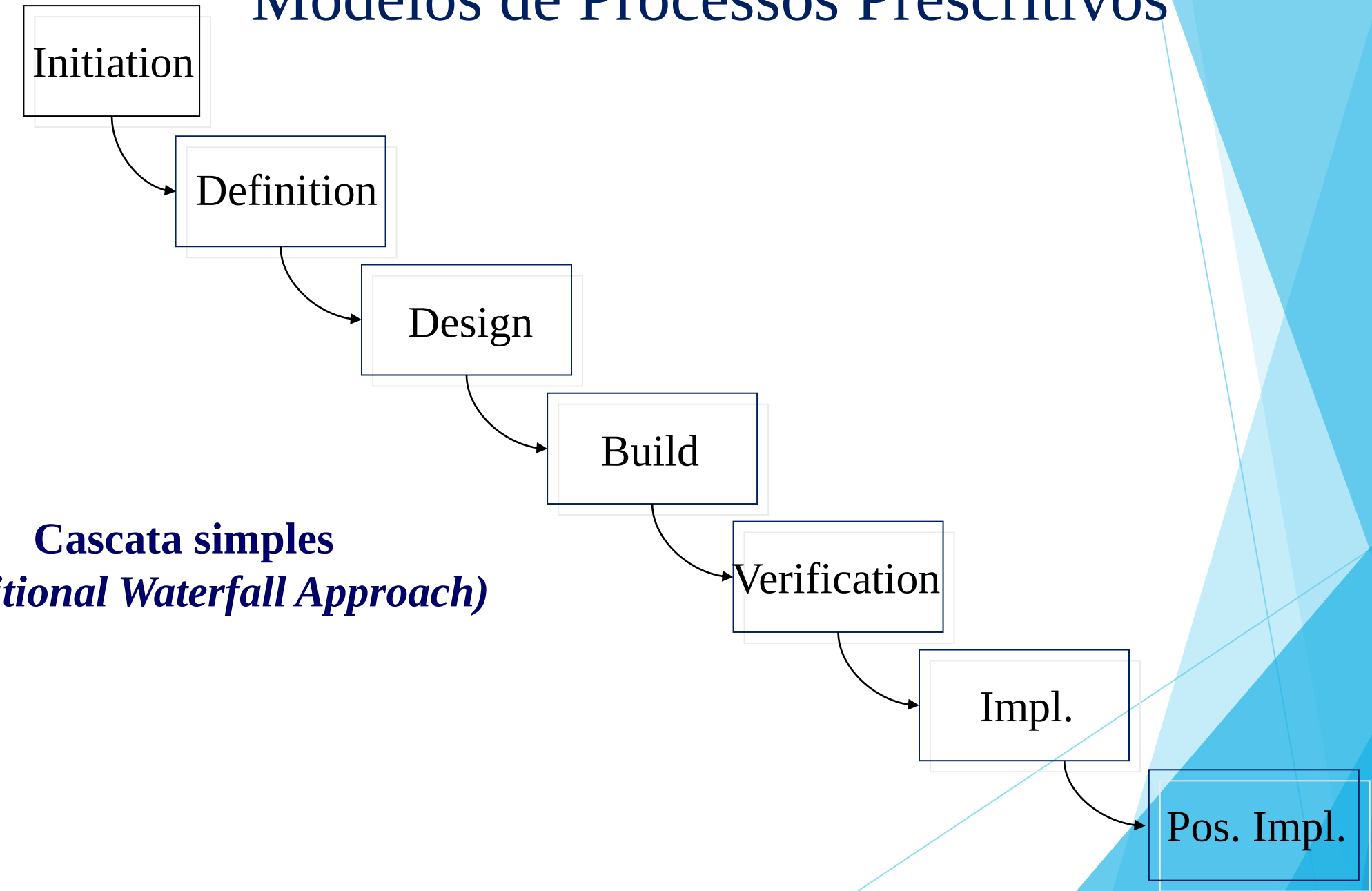
Modelos de Processos Prescritivos

Modelo de Processo de Software Genérico (baseado no Cascata)



(Pressman , 2011)

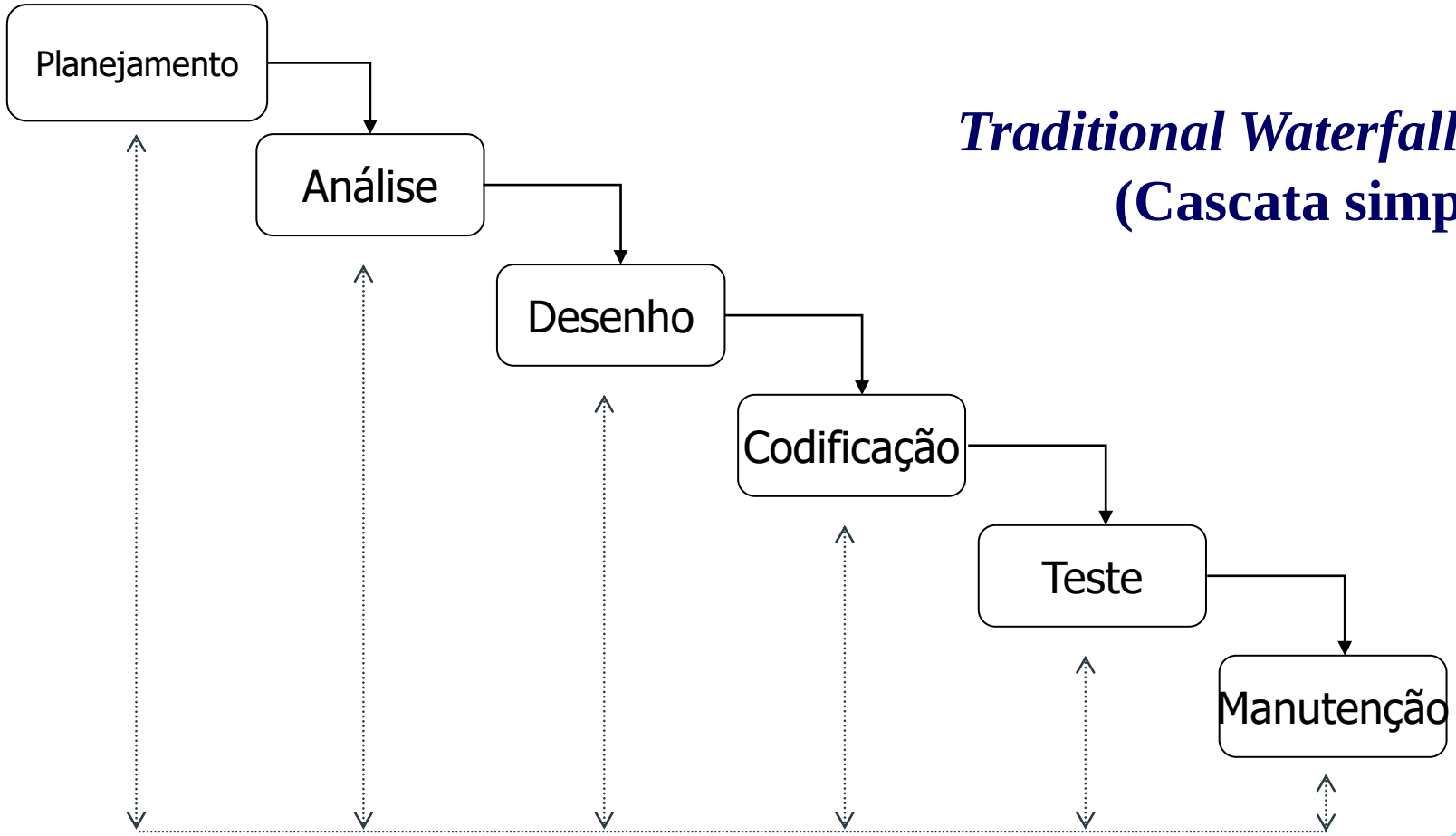
Modelos de Processos Prescritivos



Cascata simples

(Traditional Waterfall Approach)

Modelos de Processos Prescritivos



Traditional Waterfall Approach
(Cascata simples)

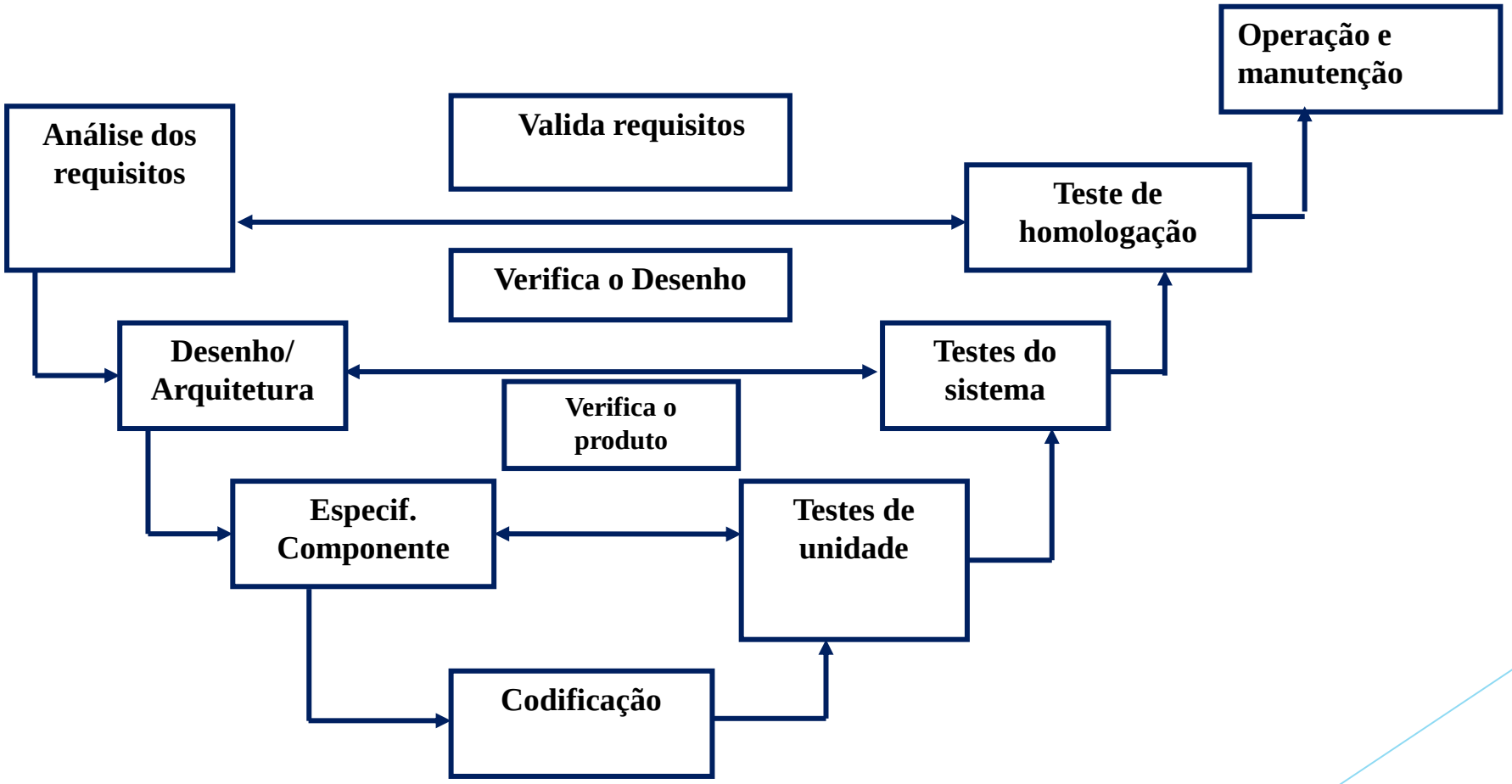
Modelos de Processos Prescritivos

Modelo Cascata (*Waterfall*) (Cascata “V”)

- Ações conjuntas de desenvolvimento e de testes
- Propicia a garantia da qualidade
- O lado esquerdo favorece ações de análises dos requisitos, modelagem, culminando na codificação; enquanto o lado direito favorece o gerenciamento de testes
- A realização do lado esquerdo favorece as posteriores ações do lado direito

Modelos de Processos Prescritivos

Cascata “V” - *Waterfall Approach*



Modelos de Processos Prescritivos

Questão

Os modelos com fluxos de processos exclusivamente linear, como o Cascata, possuem espaço nas organizações de desenvolvimento de software nos dias de hoje ?

Modelos de Processos Prescritivos

Modelo Evolucionário

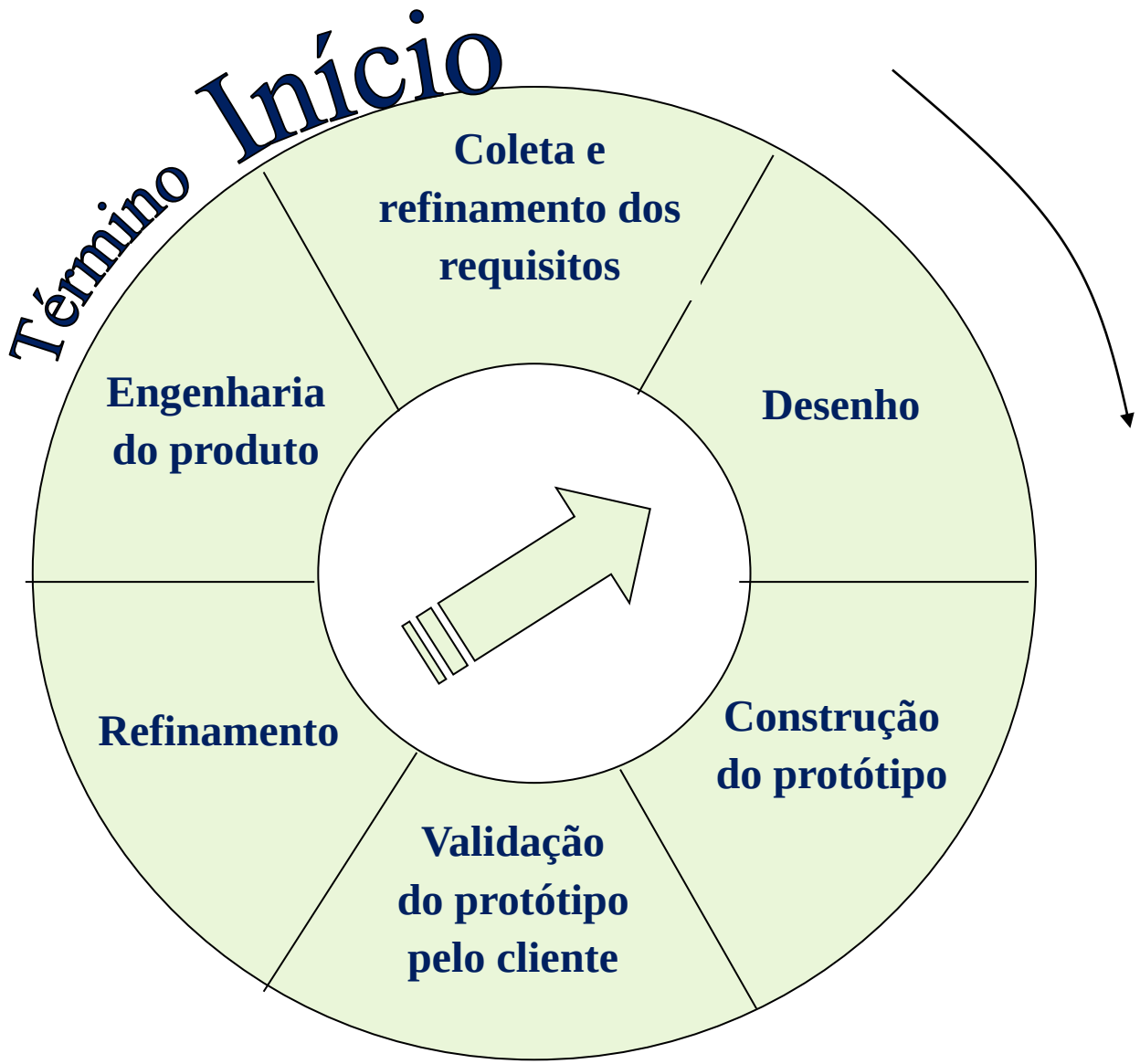
- Modelos também iterativos – possibilitam versões cada vez mais completas do produto de software
- Permite versões parciais
- Contam com requisitos que mudam constantemente
- Exemplos de Modelos Evolucionários correspondem ao Prototipação e Espiral

Modelos de Processos Prescritivos

Modelo Evolucionário - Prototipação

- Pode ser considerado como um modelo de processo isolado (*stand-alone process*)
- Em geral é um modelo utilizado com outros modelos prescritivos
- Favorece ao “projeto rápido” mesmo quando não há segurança quanto a eficiência de um algoritmo, adaptabilidade do sistema ou quanto à forma que deve ocorrer a interação homem-máquina

Modelos de Processos Prescritivos



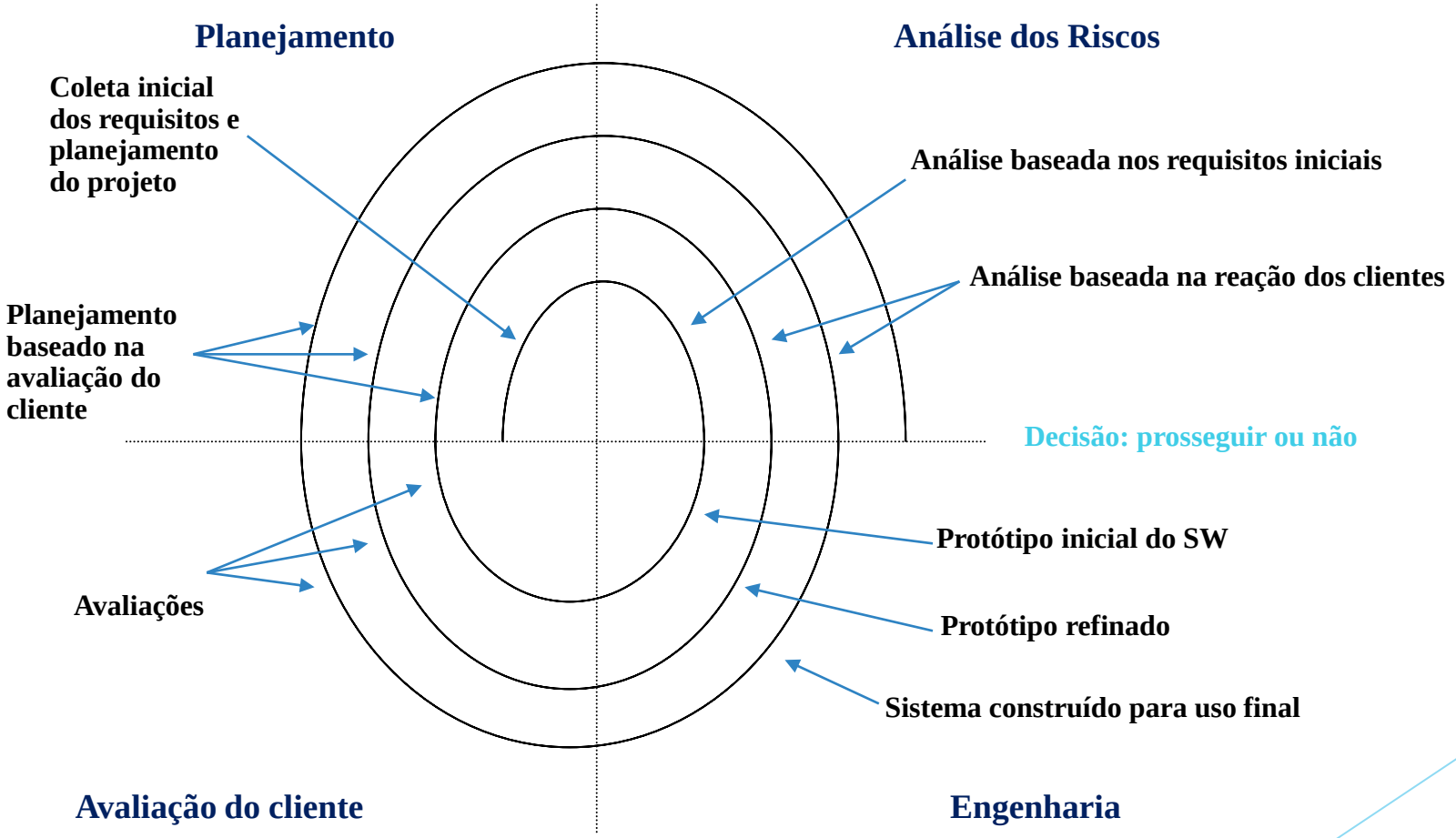
Evolucionário
Paradigma da
Prototipação

Modelos de Processos Prescritivos

Modelo Evolucionário - Espiral

- Acopla as características iterativas próprias do Prototipação juntamente com aspectos sistemáticos e controlados do Cascata
- Permite desenvolvimento de versões cada vez mais completas do software
- As primeiras versões podem corresponder a um protótipo
- Riscos são revisados a cada iteração
- É um ciclo que pode ser considerado ao longo de toda a vida do software

Modelos de Processos Prescritivos



Evolucionário
Espiral de Boehm

Modelos de Processos Prescritivos

Modelo Incremental

- Requisitos iniciais razoavelmente bem definidos, porém não favorece o uso de um modelo puramente linear
- Desenvolvimento de forma incremental – fornecimento de um conjunto funcional aos usuários para posterior refinamento
- O Processo Unificado (UP) (*Unified Process*) é um exemplo de aplicação do Modelo Incremental sendo dirigido a casos de uso e centrado na arquitetura.

Processo Unificado

Processo Unificado (*Unified Process*)

Unified Process . . . . . .estrutura combinada (**Linear e iterativo**)
(Jacobson *et al.*, 1999)

O UP (*Unified Process*) é um processo proprietário de Engenharia de Software criado pela *Rational Software Corporation*, posteriormente adquirida pela IBM, a partir da evolução e integração do trabalho de três autores: Ivar Jacobson, Grady Booch e James Rumbaugh.

Segue a abordagem da orientação a objetos em sua concepção e é projetado utilizando a notação UML (*Unified Modeling Language*) para ilustrar os processos em ação.

O UP é um modelo híbrido de processo que reúne as perspectivas dinâmicas (as fases) e estáticas (os fluxos de trabalho (*workflows*)) em um único processo.

Processo Unificado

Principais Características do Processo Unificado

- Desenvolvimento utiliza o Modelo de Processos Incremental, portanto com fluxo misto, Iterativo e Linear
- Conduzido por Casos de Uso
  - A identificação de casos de uso e cenários típicos de utilização é a atividade que conduz todo o processo de desenvolvimento, desde a análise de requisitos até ao teste final do sistema
- Centrado numa Arquitetura
  - Promove a definição inicial de uma arquitetura de software robusta, que facilita a paralelização do desenvolvimento, a reutilização e a manutenção

Scott (2003)

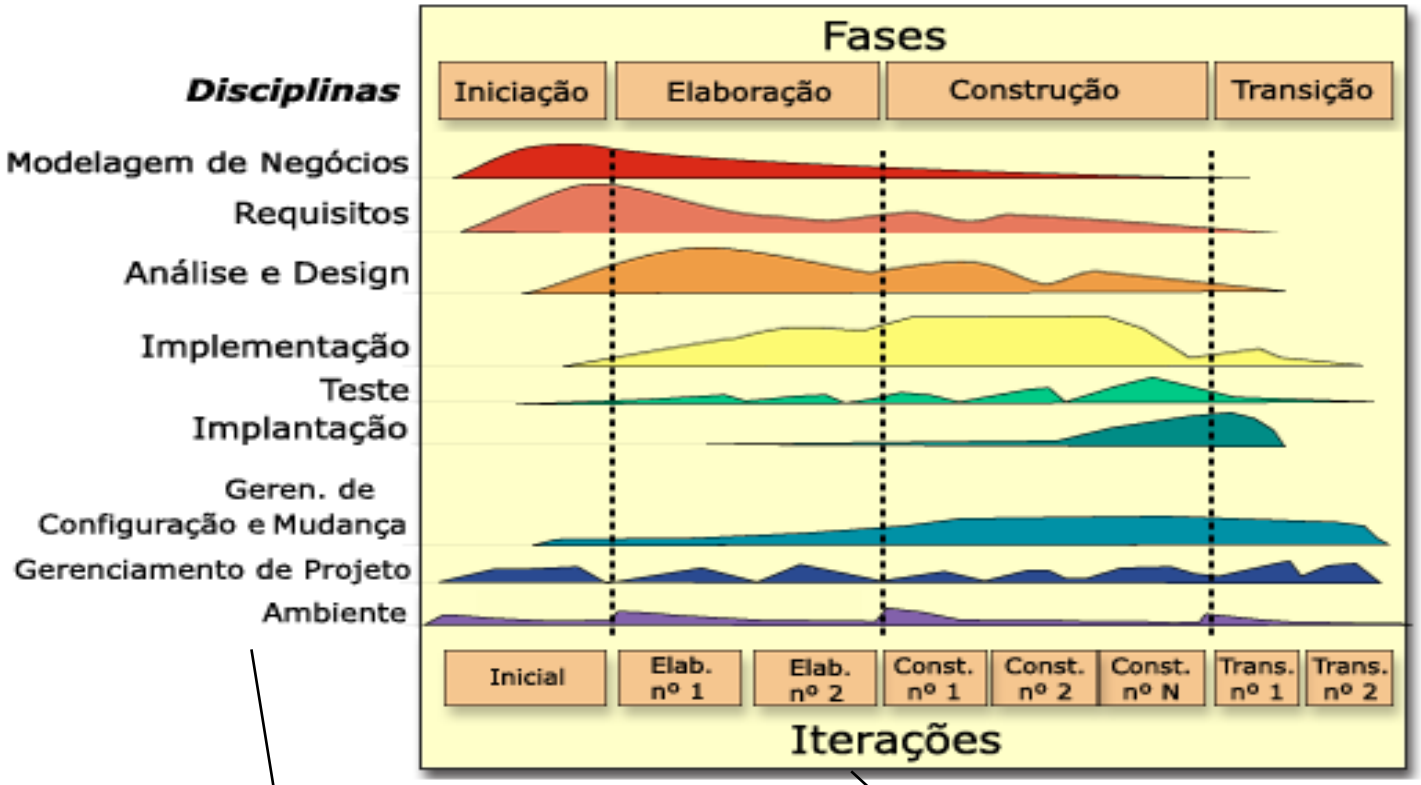
Processo Unificado

Estrutura do Processo Unificado

- **Fases**
  - **cada ciclo resulta numa nova geração do produto e divide-se em fases**
  - **cada fase divide-se em iterações a definir em cada projeto concreto**
- ***Workflows* (fluxos de trabalho)**
  - **agrupam atividades relacionadas**
  - **genéricos ou especializados por fases**
- **Papéis (*workers*)**
  - **são perfis a que correspondem competências para a realização das atividades**
- **Atividades**
  - **são realizações que podem ser entregues a trabalhadores individuais**
- **Artefatos**
  - **são as entradas e saídas de cada atividade**
- **Modelos**
  - **agrupam artefatos desenvolvidos num *workflow***

Scott (2003)

Processo Unificado



Fases e *Workflows*

Atividades são agrupadas em workflows

O desenvolvimento é expresso em fases, ciclos e iterações

IBM Rational Unified Process

Processo Unificado

Processo Unificado – desenvolvimento incremental

- **ITERAÇÃO**
  - Construção incremental durante o processo
    - Mudanças no final são caras
  - Resulta numa versão do produto, que pode ser validada pelo usuário
  - Auxilia no gerenciamento, organização, acompanhamento e no controle do projeto

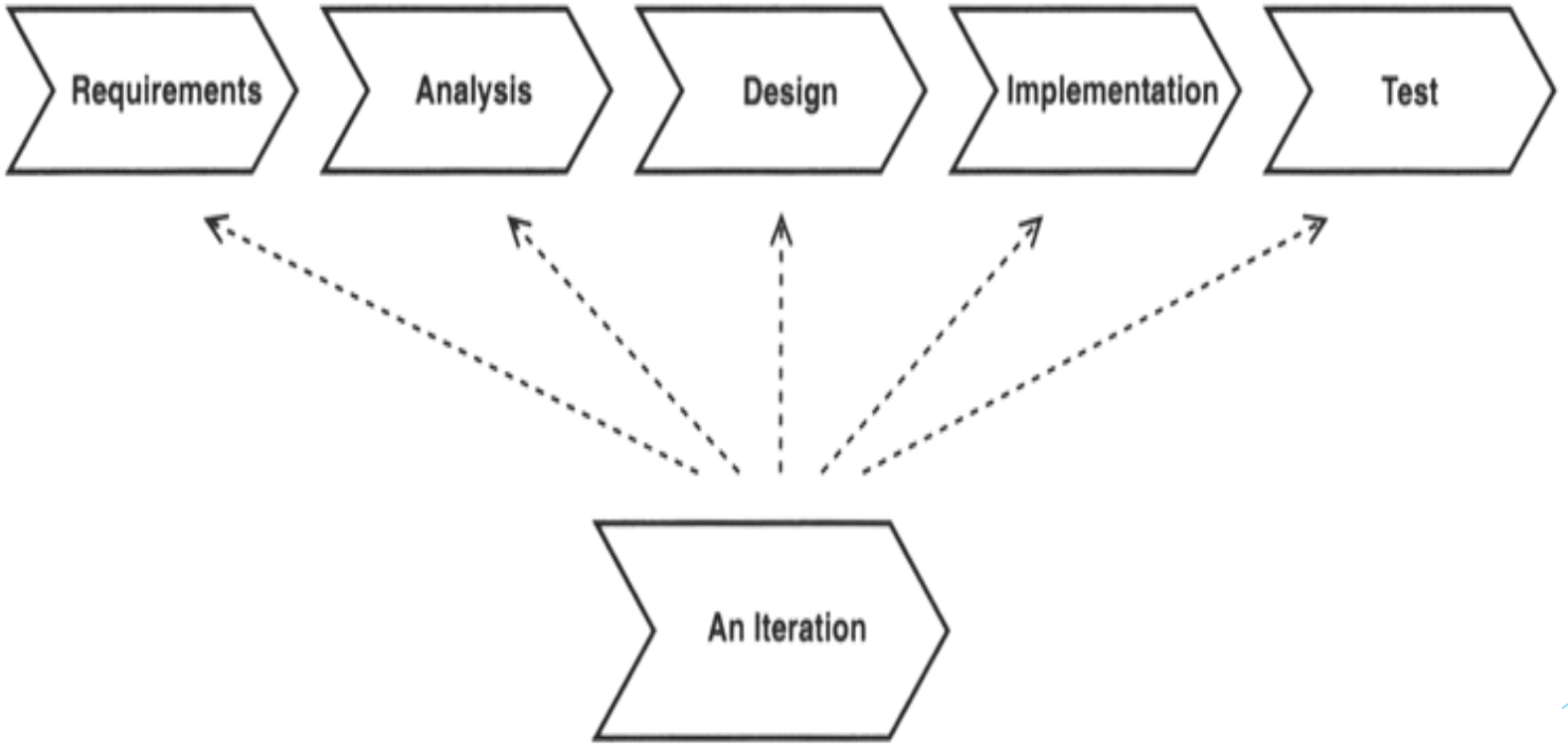
- **INCREMENTO**
  - Diferença entre versões de consecutivas iterações

- **Desenvolvimento por iterações que consideram**
  - Planejamento
  - Especificação, projeto (arquitetura) e implementação
  - Integração e teste
  - Ajustes entre as iterações (*feedback*)

Scott (2003)

Processo Unificado

Processo Unificado – elementos genéricos de uma iteração - (*workflows*) fluxos de trabalho



IBM Rational Unified Process

Processo Unificado

Processo Unificado – vantagens de desenvolvimento incremental

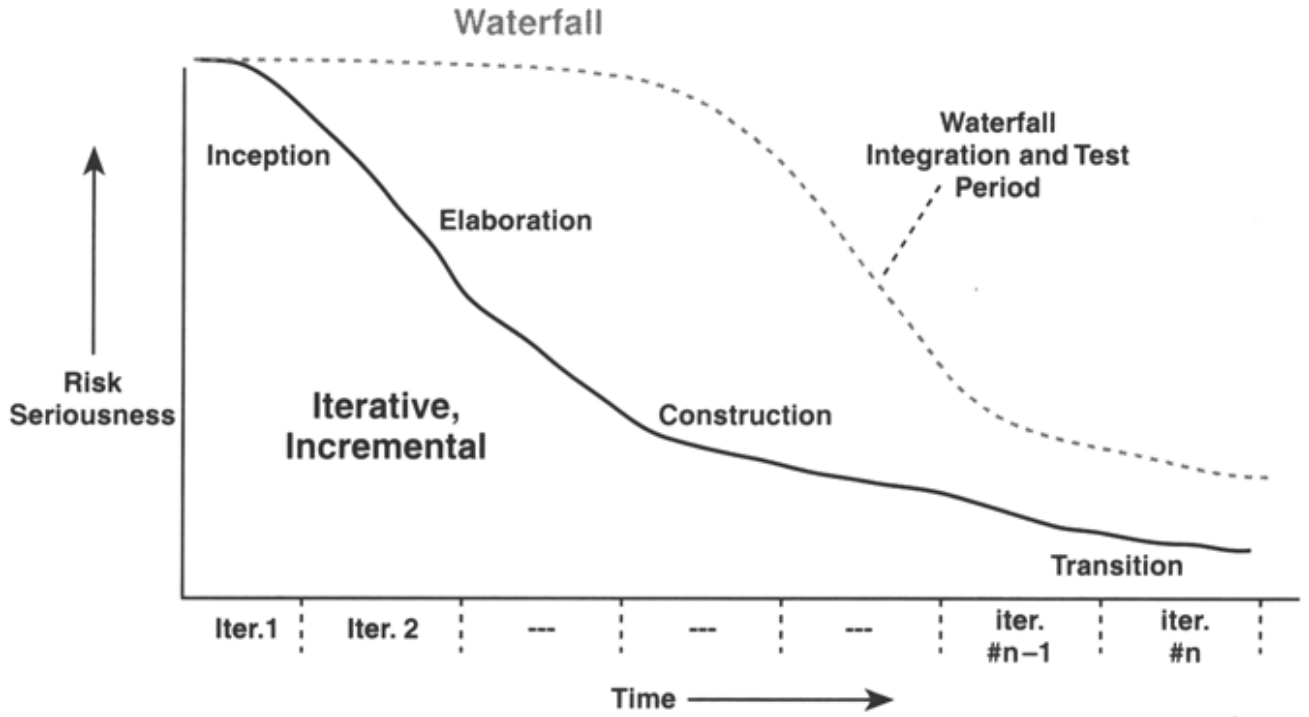
- **Obtenção de uma arquitetura robusta**
  - Propor uma arquitetura que satisfaça os requisitos (concepção)
  - Estabelece-se a linha-base (*baseline*) da arquitetura que guia o desenvolvimento (elaboração)
  - Pequeno investimento inicial
  - Validação da arquitetura
- **A cada versão do produto, os usuários podem validar ou propor mudanças nos requisitos**
  - Mudanças no final do projeto são caras
- **Melhor entendimento dos *workflows***
  - O que deve ser feito após os requisitos e após a análise (otimização)
  - Menor gravidade dos erros
  - Facilidade de treinar novas pessoas
- **Reduzir riscos críticos**

Processo Unificado

Processo Unificado – desenvolvimento incremental

Riscos podem ser identificados e reduzidos mais eficientemente (no início do processo)

Modelo de Processos Incremental



Processo Unificado

Fases do Processo Unificado

Fase 1 – INICIAÇÃO ou CONCEPÇÃO (*Inception*)

Delimitação do escopo do projeto e do Business Case
Identificação dos atores e casos de uso, com a descrição dos mais significativos

Fase 2 - ELABORAÇÃO (*Elaboration*)

Análise do domínio do problema
Definição de uma arquitetura estável e robusta para todo o sistema, a partir de seus requisitos compreendidos e especificados

Fase 3 - CONSTRUÇÃO (*Construction*)

Desenvolvimento incremental do produto em iterações com entregas parciais aos usuários e como uma primeira versão do produto completo

Fase 4 - TRANSIÇÃO (*Transition*)

Desenvolvimento e testes adicionais para ajuste do sistema às novas demandas de requisitos para sua concreta utilização

Processo Unificado

Fase 1 – Iniciação ou Concepção

- Identificar e reduzir riscos críticos (viabilidade do projeto)
- A partir de um subconjunto chave de requisitos propor uma arquitetura para implementação do software
- Elaborar um cronograma inicial (custo, recursos, tempo e qualidade do produto)
- Iniciar o *business case* (viabilidade econômica do projeto)

Processo Unificado

Fase 2 – Elaboração

- Reduzir riscos (construção do sistema)
- Especificar os componentes para a maioria dos *use cases* que ofereçam funcionalidade aos usuários
- Validar a arquitetura proposta
- Preparar um plano de projeto (guiar a fase de construção)
- Finalizar o *business case*

Processo Unificado

Fase 3 – Construção

- O sistema deve ser capaz de operar no ambiente do usuário
- Iterações e incrementos que ao longo da fase torna evidente a viabilidade do sistema
- Desenvolver todos os componentes do sistema

Processo Unificado

Fase 4 – Transição

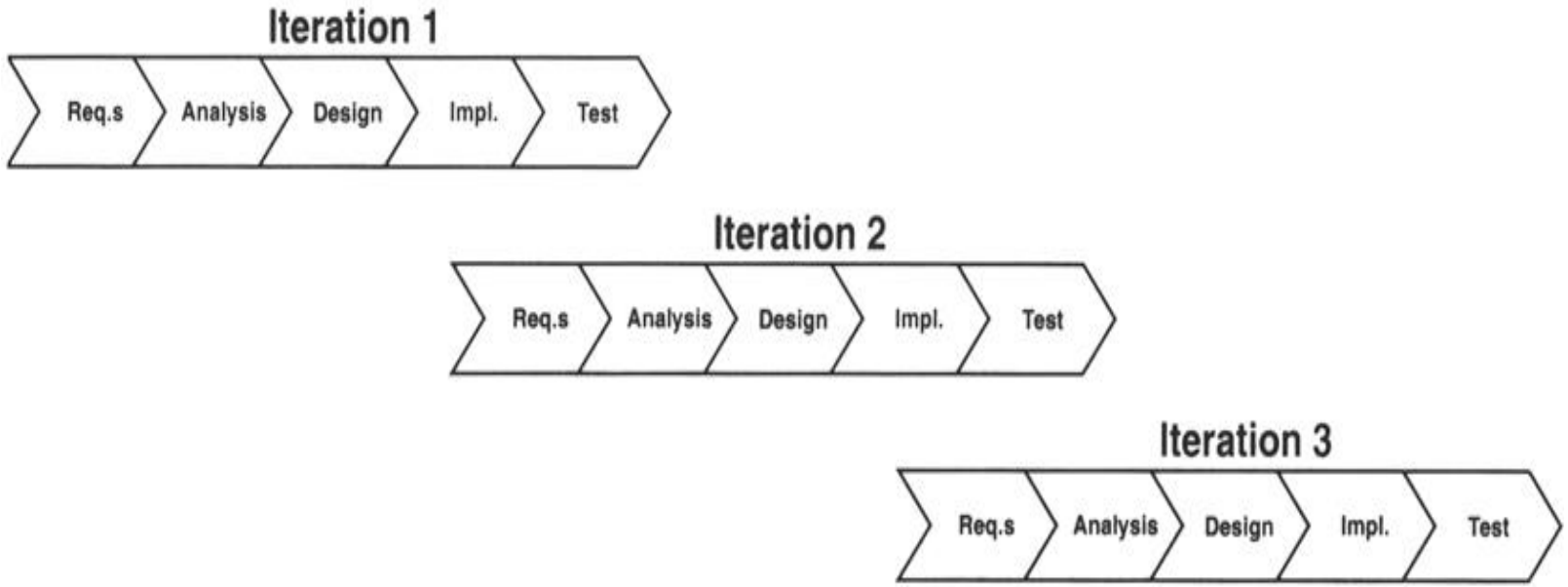
- Atingir a capacidade final de operação
- Finalizar os testes de pré-implantação
- Modificar o produto para diminuir problemas não detectados nas fases anteriores
- Corrigir defeitos
- Garantir que o produto está pronto para ser entregue ao cliente

Processo Unificado – Fases e Iterações

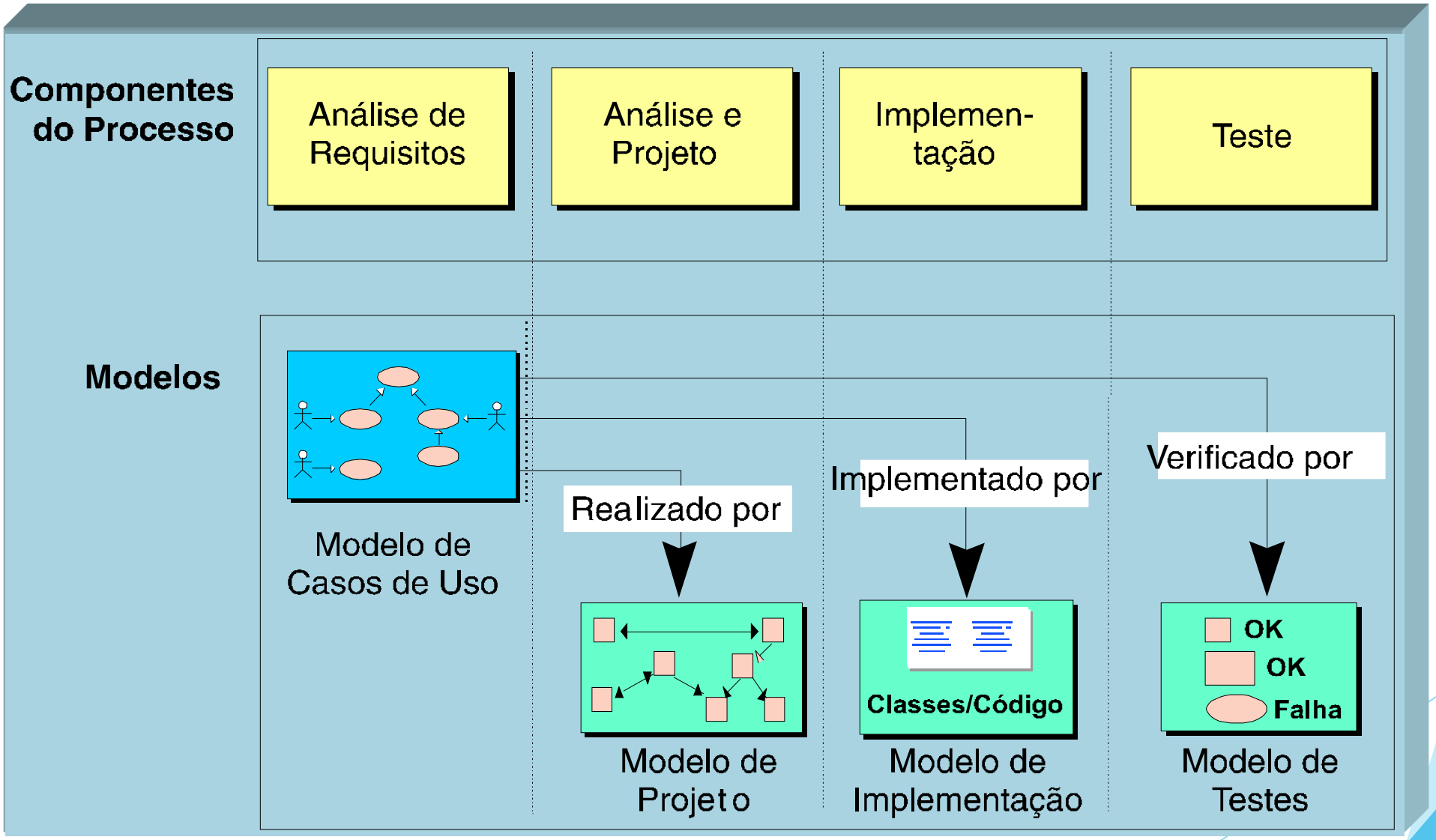
- **Fase ≠ Iteração**
  - **Iteração ∈ Fase**
- Cada **fase** pode ser **decomposta** em **iterações**
- Iterações são definidas e planejadas em cada projeto concreto
- Cada **iteração** resulta num **incremento do produto**
  - Tipicamente é analisado e implementado um **grupo de casos de uso** ou de requisitos expressos em variantes de casos de uso
- Cada **iteração** passa, em **cascata**, pelos ***workflows* técnicos**
  - **Importância** relativa dos ***workflows*** varia com as **fases**
- Uma **iteração** é um **miniprojeto**
  - Requisitos, análise, projeto (*design*), implementação e teste

Processo Unificado

Sobreposição das Iterações



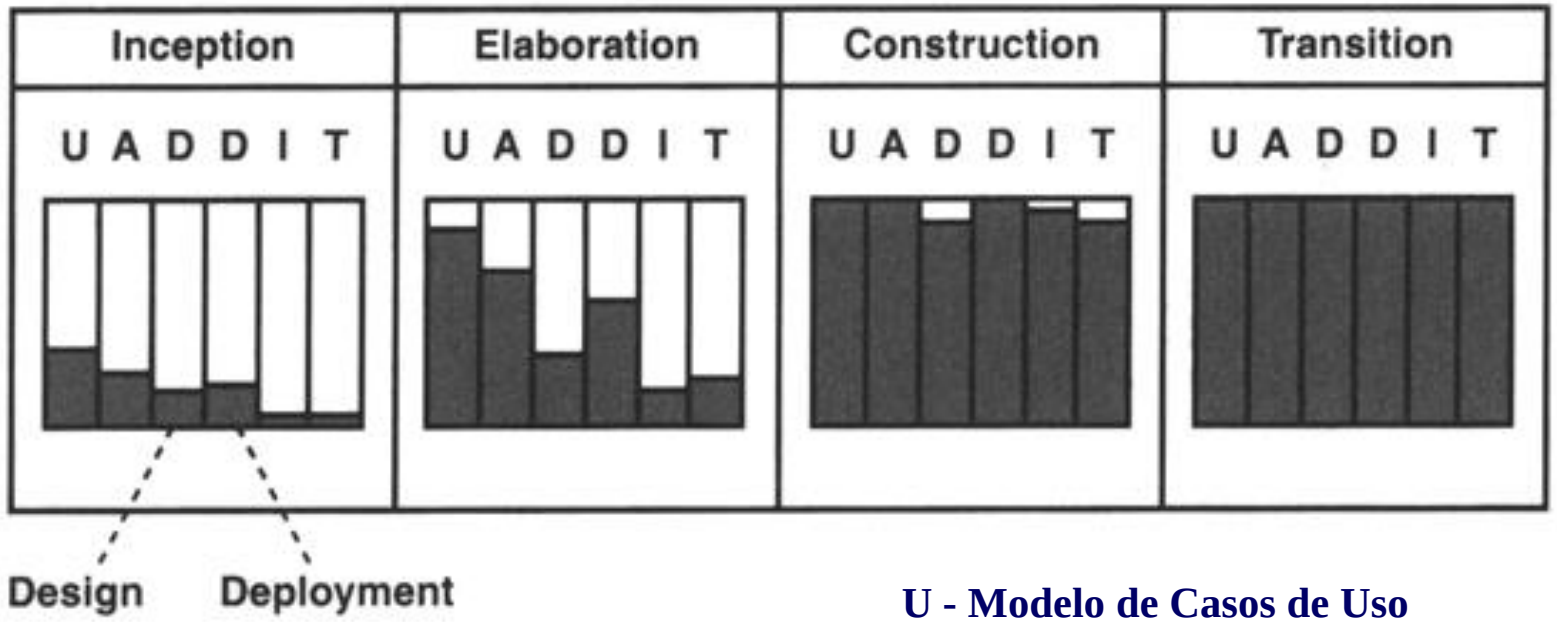
Processo Unificado



IBM Rational Unified Process

Processo Unificado

Evolução dos Modelos nas Fases



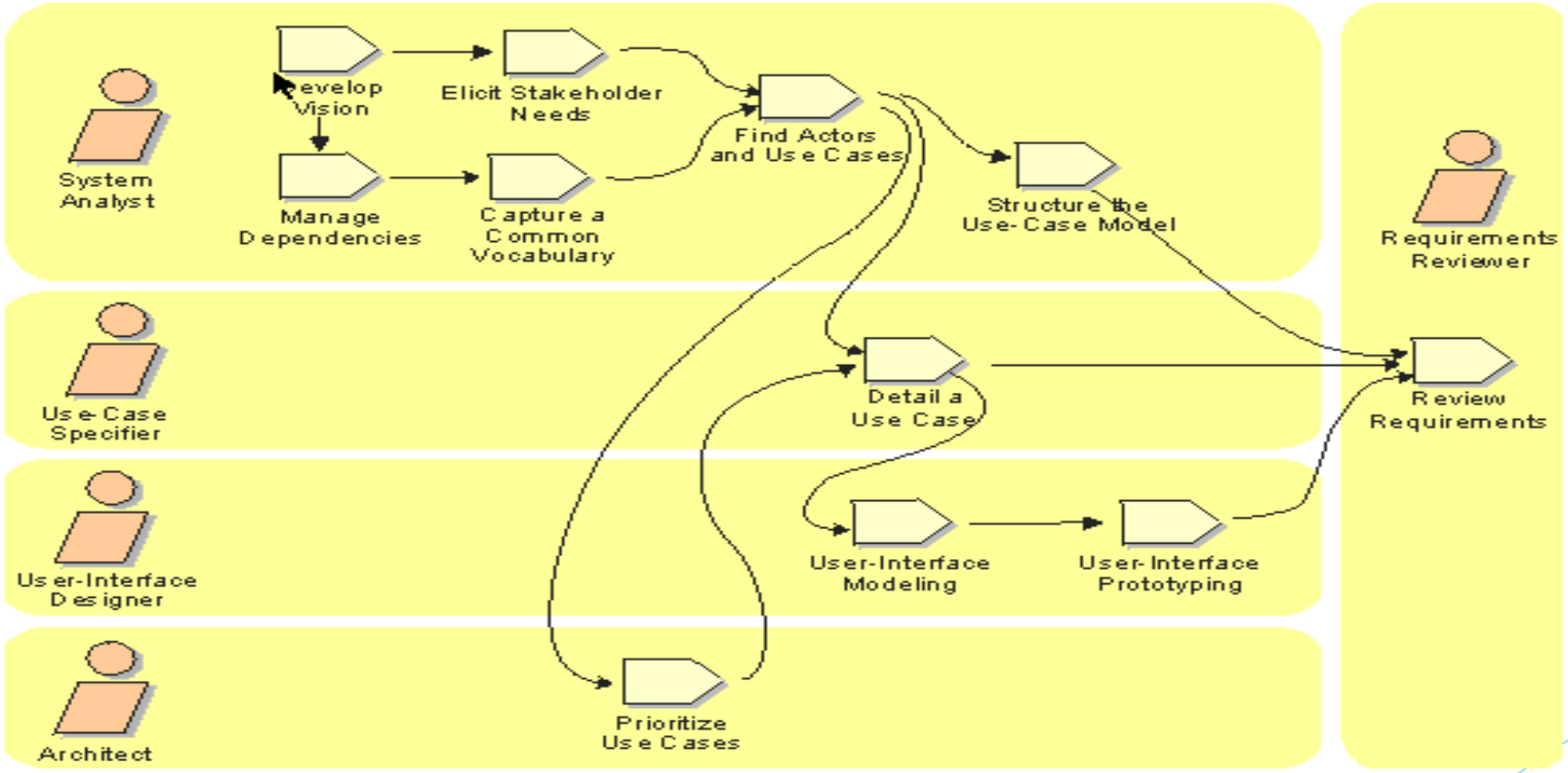
U - Modelo de Casos de Uso
A - Modelo de Análise
D - Modelo de Projeto (*Design*)
D - Modelo de Implantação
I - Modelo de Implementação
T - Modelo de Testes

Requisitos – objetivo

- O objetivo desta etapa do *workflow* é favorecer o **acordo entre os usuários finais e o desenvolvedor** sobre a descrição do que o sistema deve fazer
- Principal resultado é o **Modelo de Casos de Uso**
  - Um caso de uso é um elemento narrativo que descreve a sequência de eventos de um ator sobre um sistema com o objetivo de obter, a partir dele, um resultado observável e de interesse sobre o domínio do problema
  - O **Modelo de Casos de Uso** é a base de todo o processo de desenvolvimento o que facilita a avaliação da concordância do usuário final para o sistema entregue conforme os requisitos iniciais previamente definidos

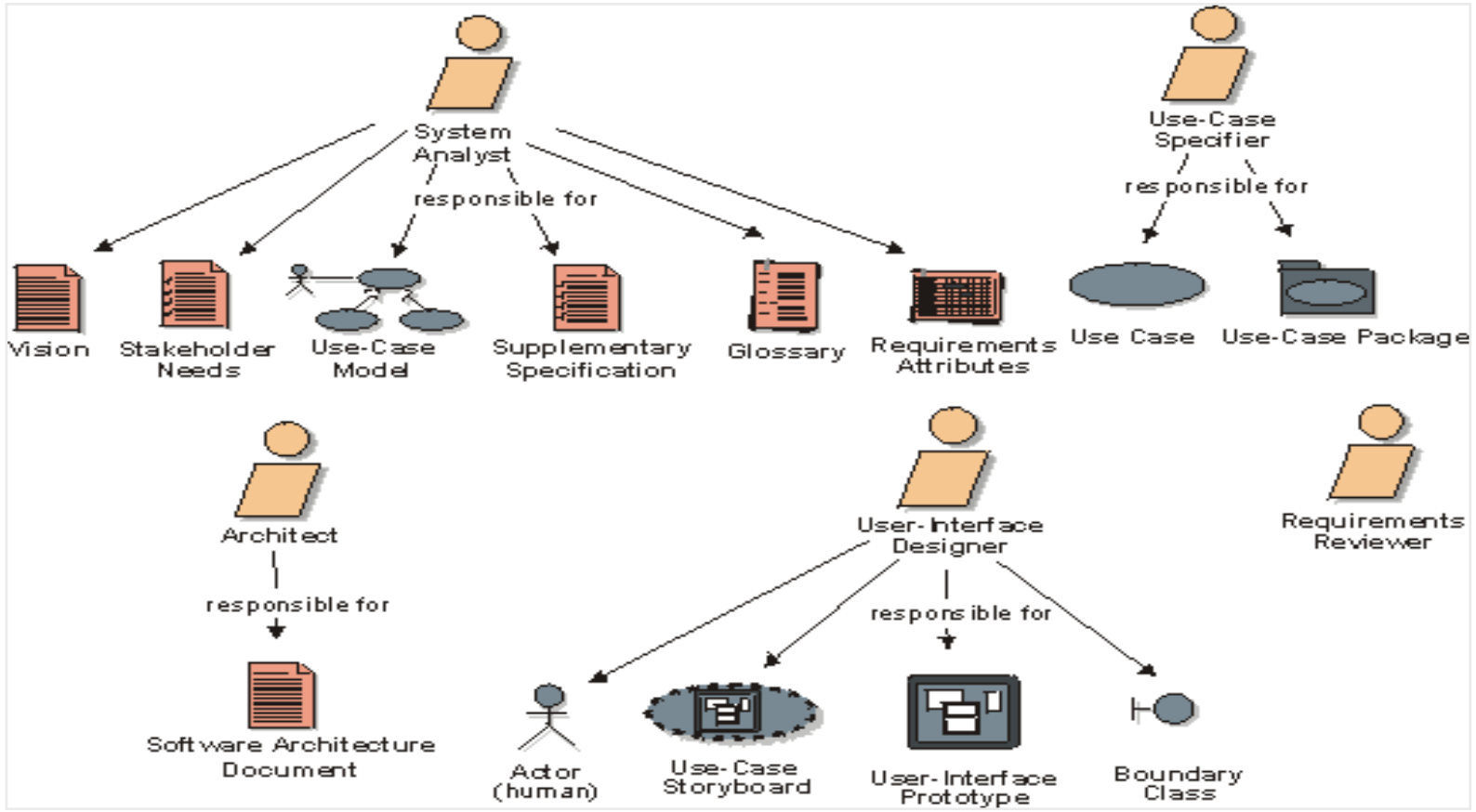
Processo Unificado

Requisitos – atividades



Processo Unificado

Requisitos – artefatos



Processo Unificado

Análise e *Design* – objetivo

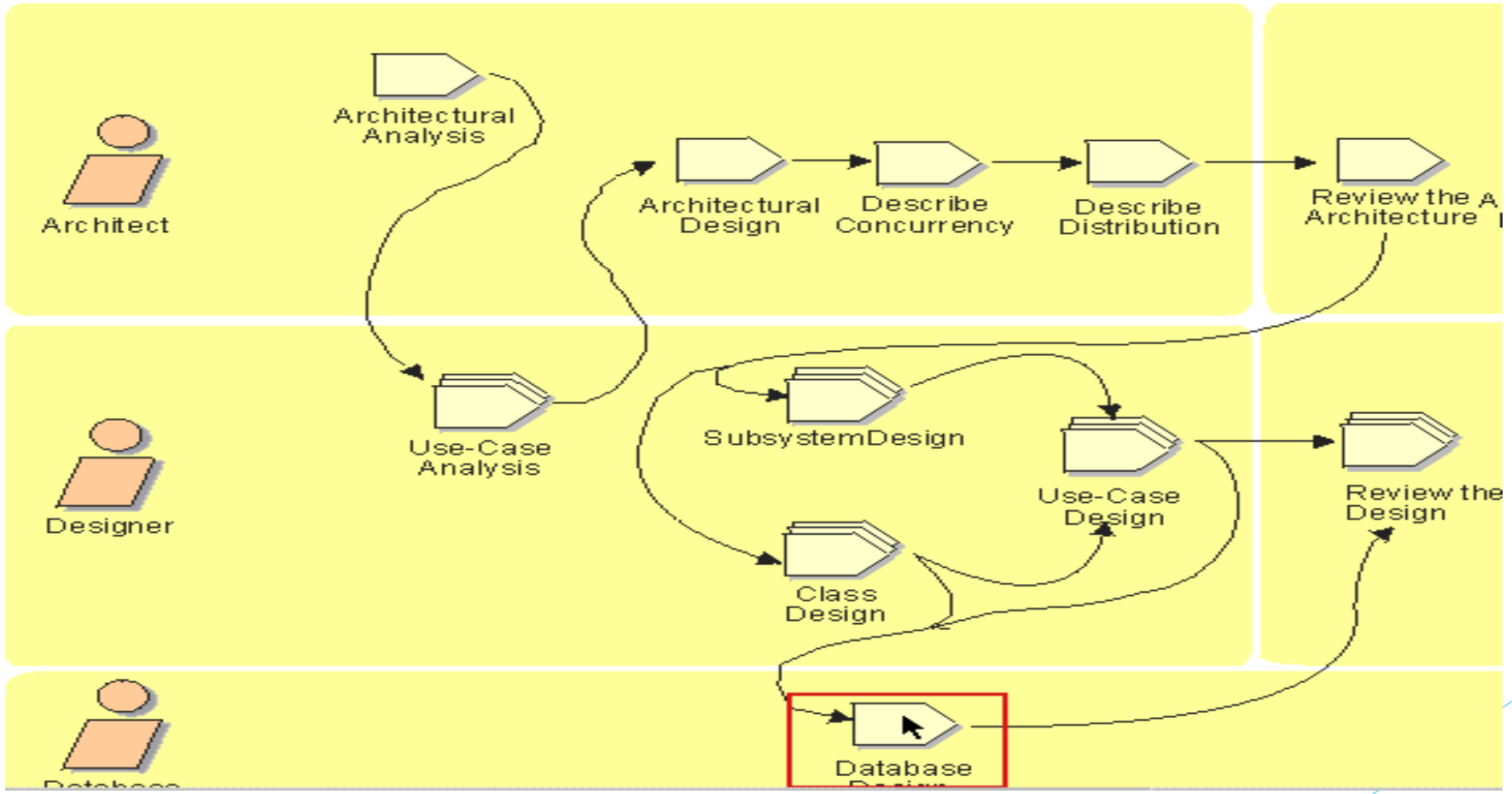
- **ANÁLISE**
  - Modelos de Classes e Objetos ideais para uma melhor compreensão dos requisitos
  - principal resultado é o Modelo de Análise

- ***DESIGN***
  - o objetivo desta etapa do *workflow* é apresentar como o software será construído, de forma a satisfazer a todos os requisitos, tarefas e funções descritas no Modelo de Casos de Uso
  - o software deve ser projetado de modo que possua uma arquitetura robusta e facilmente adaptável a mudanças de requisitos
  - o principal resultado é o Modelo de Projeto (*Design*)
  - também pode ser produzido o Modelo de *Deployment*

IBM Rational Unified Process

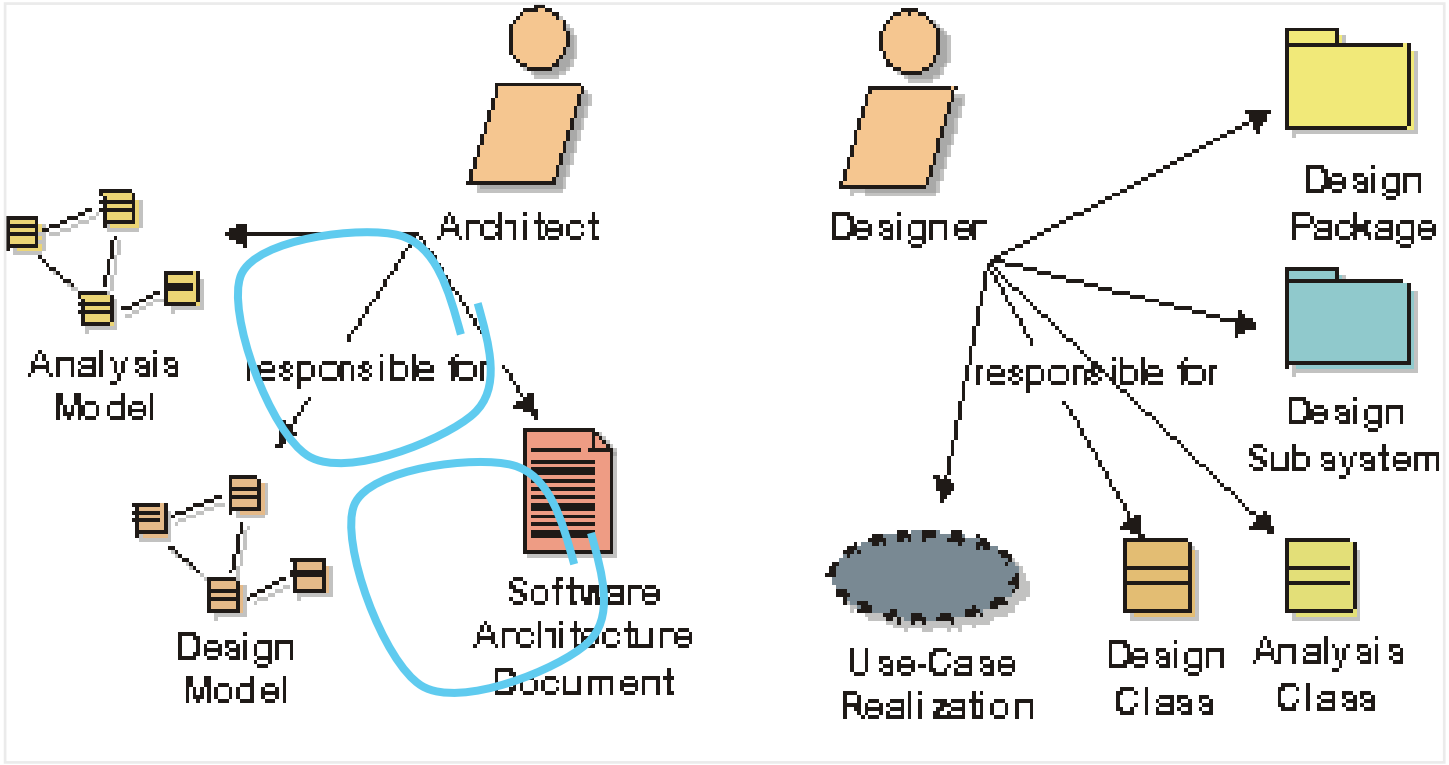
Processo Unificado

Análise e *Design* – atividades



Processo Unificado

Análise e *Design* – artefatos



IBM Rational Unified Process

Processo Unificado

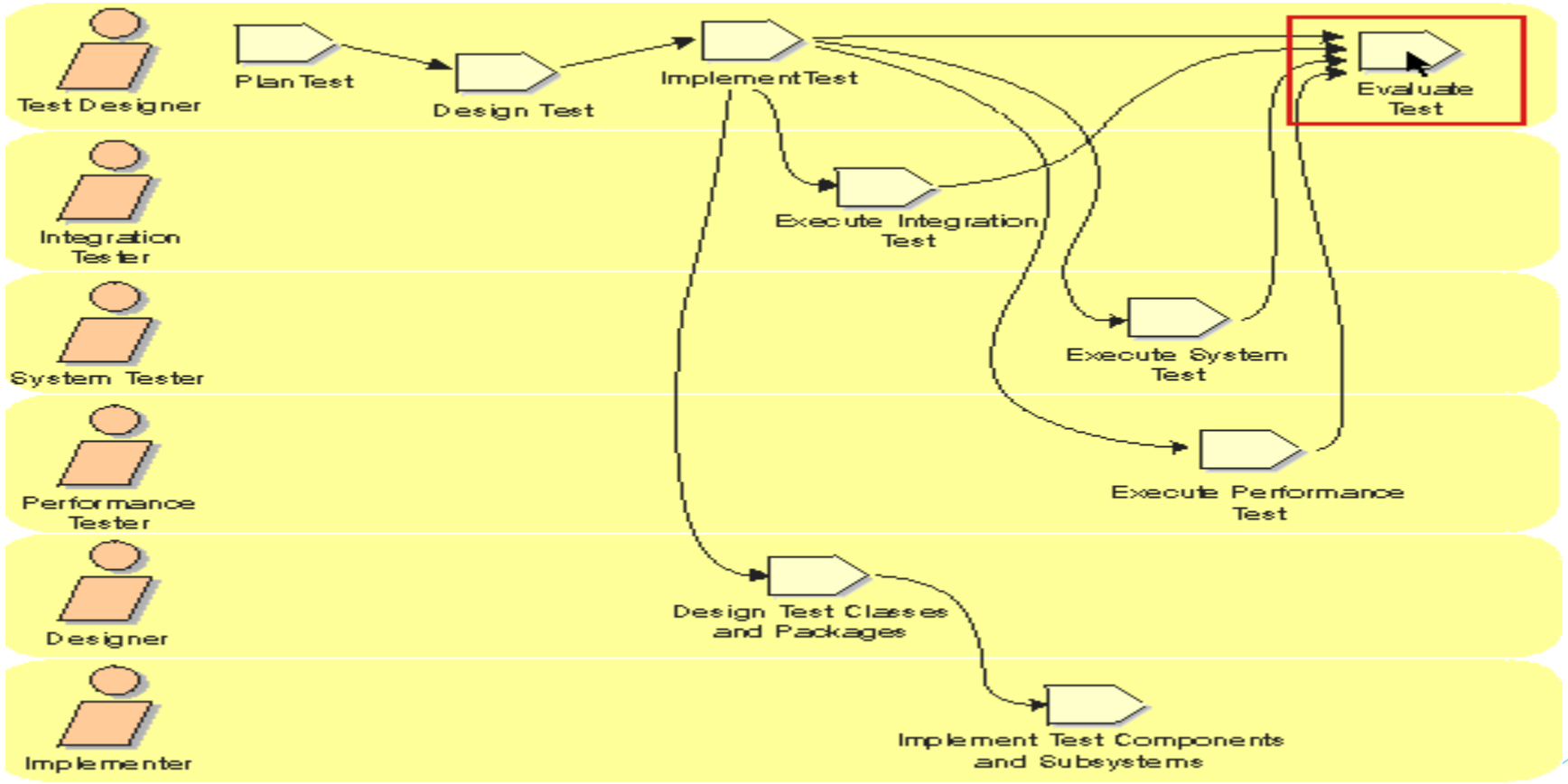
Implementação e Testes – objetivo

- **IMPLEMENTAÇÃO**
  - o objetivo é **construir os componentes do produto**, produzindo o código necessário para a geração do módulo executável
  - o Modelo de Projeto (*Design*) é a base da implementação
  - a implementação inclui o teste de classes e módulos separados, mas não a verificação do seu funcionamento integrado
  - sub-produto: Modelo da Implementação (componentes, dependências e interações)
- **TESTE**
  - o objetivo é verificar o produto completo
  - inicialmente verifica-se cada caso de uso separadamente e posteriormente o software na sua totalidade
  - no final deste componente, o produto estará pronto para ser utilizado
  - sub-produto: Modelo de Teste, com especificação de casos de teste e procedimentos de teste

IBM Rational Unified Process

Processo Unificado

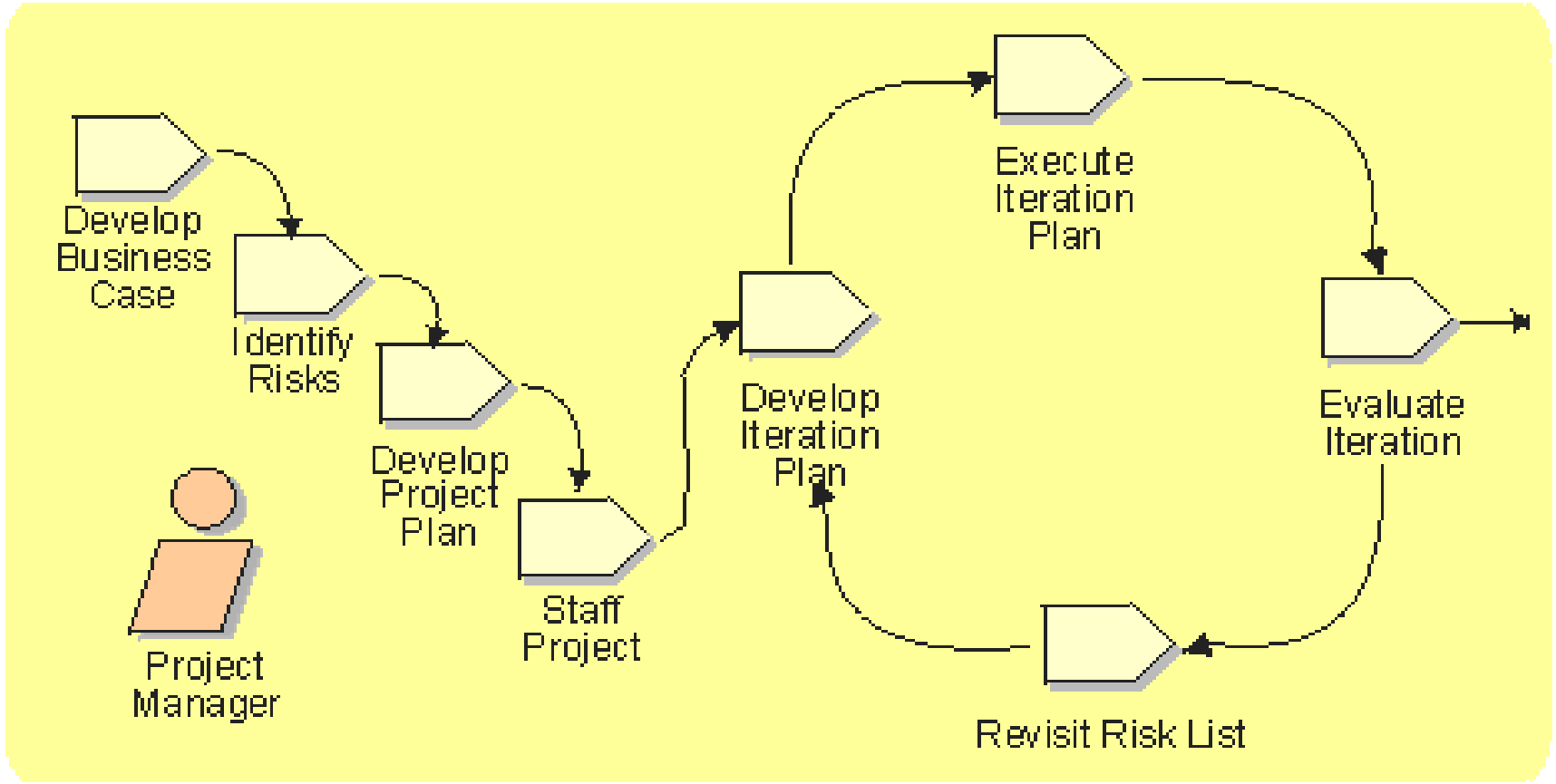
Implementação e Testes – artefatos



IBM Rational Unified Process

Processo Unificado

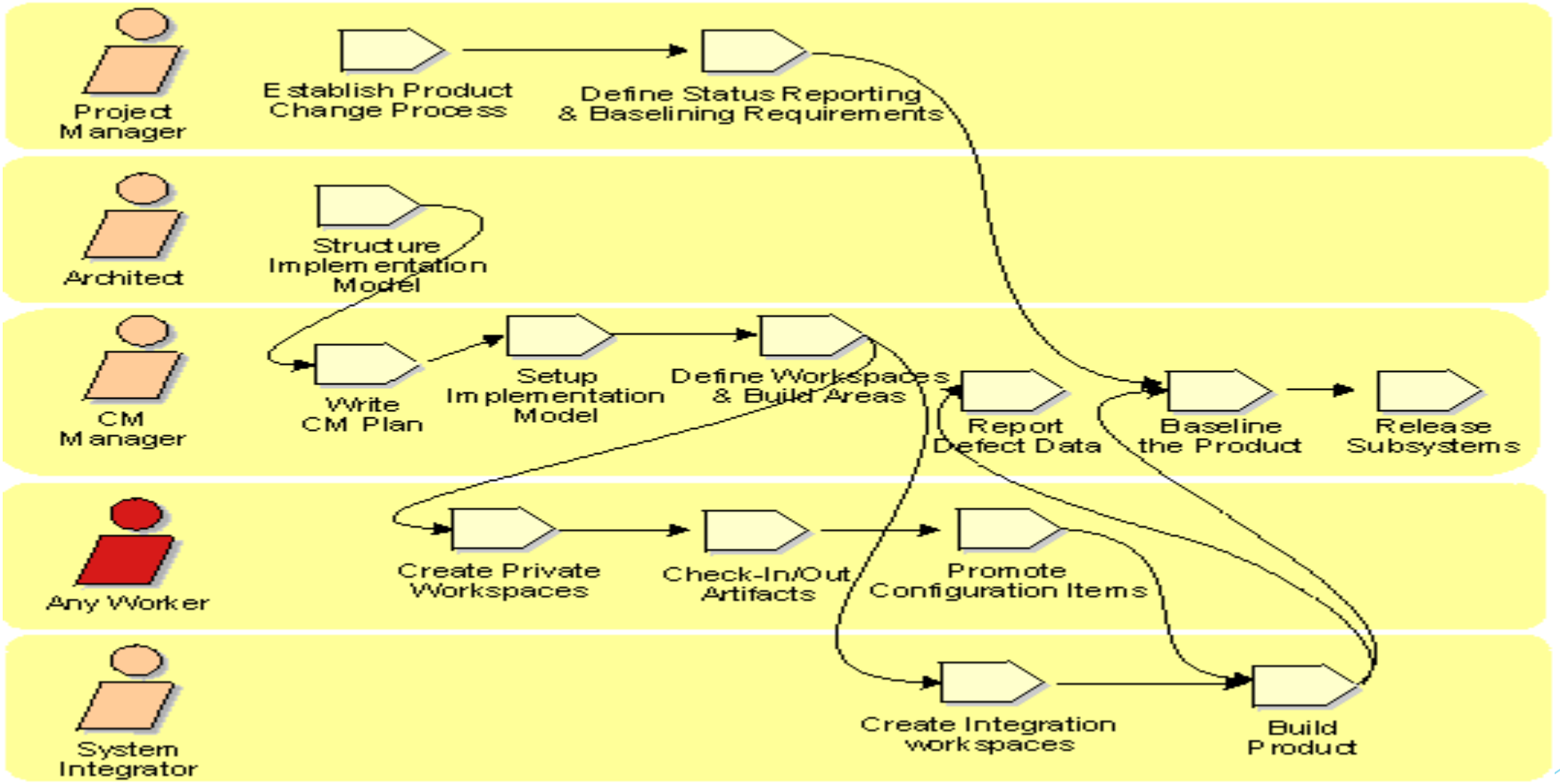
Gerenciamento de Projetos - atividades



IBM Rational Unified Process

Processo Unificado

Gerenciamento de Configuração e Mudança - atividades



IBM Rational Unified Process

Modelos de Processos

Criteria	Waterfall	Incremental	Evolutionary/Iterative
Understanding of user requirements	User requirements are well defined and are measurable and testable	User requirements are difficult to define	• User requirements are difficult to define • Developers are making assumptions about the needs of the users
Requirements stability	Requirements are well defined and stable	Requirements are subject to minimal change	Requirements are highly volatile
Domain Knowledge	Strong understanding of problem domain	Some or average familiarity	Little or no familiarity
Complexity of project	Factors of complexity such as technology, subsystems, interfaces etc. are low	Factors of complexity such as technology, subsystems, interfaces etc. are medium	Factors of complexity such as technology, subsystems, interfaces etc. are high
Technology	Existing and well understood technical environments	Moderate expertise in the technologies	Technology has not been applied by the project team
Risk management perspective	Risks are manageable without extra efforts	Risks are moderately manageable without extra efforts	Significant effort required to manage risks
Schedule constraint	Moderate level of schedule constraints	Moderate level of schedule constraints	Drop-dead date with high priority schedules
Availability of resources	Resources can be identified and are available	• Resources cannot be estimated • Resources are not available.	• Resources cannot be estimated • Resources are not available.
Product availability to end user	Complete product as part of single deployment	Intermediate usable product required before all functions are available	Intermediate usable product required before all functions are available

Modelos de Processos Prescritivos

Modelos de Processos Prescritivos (*People-Oriented*)

"À medida que os profissionais de desenvolvimento de software aprendem a medir os seus trabalhos, a analisar essas medidas e a definir e atingir metas de melhoria, eles passam a enxergar os benefícios de usar o processo definido e são motivados constantemente a utilizá-lo".

(Pressman , 2011)

Modelos de Processos Prescritivos

Modelos de Processos Prescritivos (*People-Oriented*)

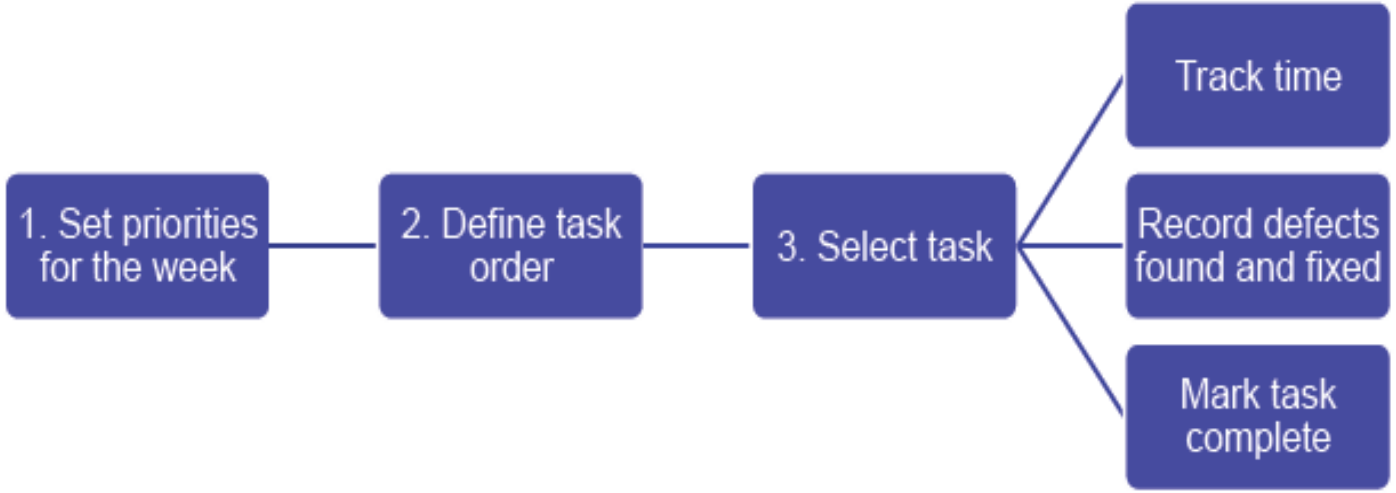
- **PSP (*People Software Process*)** é um processo de software que enfatiza a medição pessoal e responsabiliza o profissional pelo planejamento por meio das estimativas permitindo controlar os resultados alcançados.

- O PSP define cinco atividades estruturais:
    1. Planejamento
    2. Projeto de alto nível
    3. Revisão de projeto de alto nível
    4. Desenvolvimento
    5. Autópsia (exame, inspeção de si mesmo)

(Pressman , 2011)

Modelos de Processos Prescritivos

PSP (*Personal Software Process*)
atividades de acompanhamento pessoal



(SEI (asset files presentation), 2010)
http://resources.sei.cmu.edu/asset_files/Presentation/2010_017_001_24387.pdf

Modelos de Processos Prescritivos

Modelos de Processos Prescritivos (*People-Oriented*)

- **TSP (*Team Software Process*)** é um processo de software que busca criar uma equipe de projeto “autodirigida”, que se organize por si mesma, que use processos e medições em engenharia por meio de ações criativas.

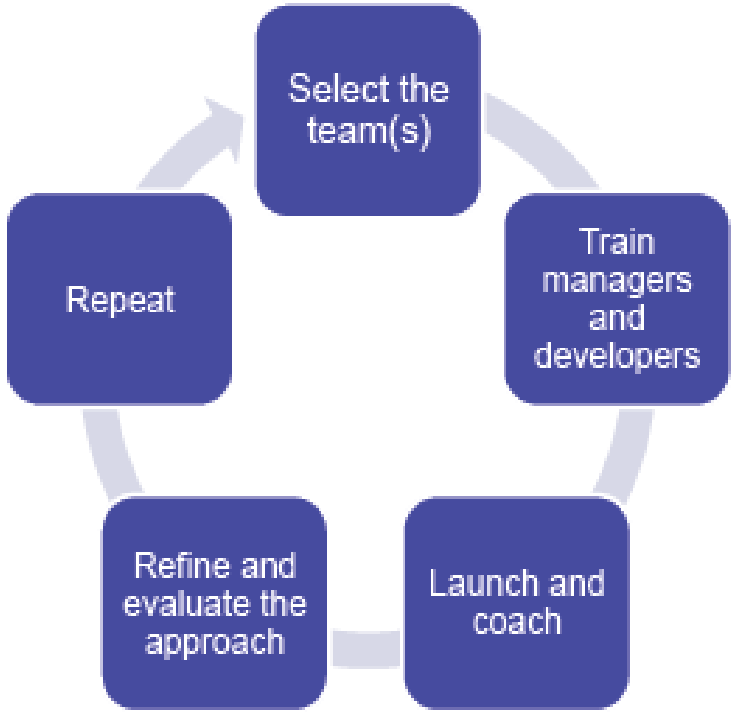
- O TSP considera as seguintes atividades metodológicas:
    1. Lançamento do Projeto
    2. Projeto de alto nível
    3. Implementação
    4. Integração e Testes
    5. Autópsia (exame, inspeção de si mesmo)

(Pressman , 2011)

Modelos de Processos Prescritivos

TSP is implemented project-by-project.

- Select two or three teams.
- Train top-down, starting with senior managers, then project managers, then team members.
- When the managers and team are trained, conduct a TSP Launch to kick-off each project.
- Evaluate and fine tune the approach.
- Repeat this cycle increasing scope at a sustainable pace.



(SEI (asset files presentation), 2010)
http://resources.sei.cmu.edu/asset_files/Presentation/2010_017_001_24387.pdf

Referências

ACUNÃ, S.T. & JURISTO, N. **Software Process Modeling**, New York: Springer, 2005.

ACUNÃ, S. T., DE ANTONIO, A., FERRE, X., MATÉ , L., & LÓPEZ, M. (2001). **The Software process: modeling, evaluation and improvement**. In *Handbook of Software Engineering and Knowledge Engineering: Volume I:Fundamentals* (pp. 193-237).

IBM Rational Unified Process. Disponível em: https://www.ibm.com/developerworks/rational/library/content/03July/1000/1251/1251_bestpractices_TP026B.pdf

KRUCHTEN, P. **The Rational Unified Process: an introduction**. Addison Wesley, 1999.

PRESSMAN, R.S. **Engenharia de Software: uma abordagem profissional**. SP:McGrawHill, 2011.

SCOTT, Kendall. O processo unificado explicado. Porto Alegre: Bookman, 2003.