



Faculdade  
**IMPACTA**  
TECNOLOGIA

# Processo de Software

## Pós-Graduação em Engenharia de Software





## Disciplina Processo de Software 24h

04.11 a 09.12

Segundas-feiras





# Processo de Software

## Revisão





## Entendendo o Problema

Profissionais de TI  
Infraestrutura / Tecnologia  
Clientes e Usuários  
Conhecimento  
Processos  
Produto





## Processos e Produto

**Produto:** resultado dos processos.

**Processo:** sequência de estados de um sistema que se transforma.





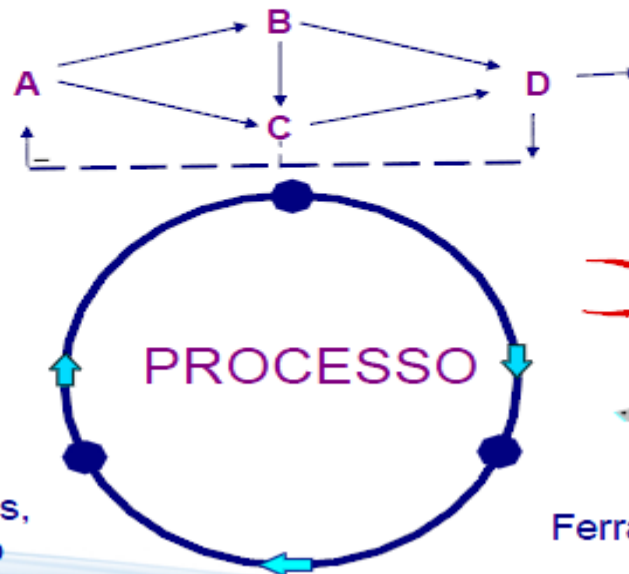
## Processos e Produto

- **Definição de Processo:**
- **Uma sequência de passos realizados para um determinado propósito. (IEEE)**

Procedimentos e métodos que definem o relacionamento de tarefas.



Pessoas com habilidades, treinamento e motivação



Ferramentas e Equipamentos



## Engenharia de Software

Tem a ver com produzir **software** com a Qualidade desejada (Requisitos), com redução de Riscos (Testes), com emprego de alta produtividade (Processos) com os menores custos possíveis (Melhores práticas de Desenvolvimento).





## Produto de Software

Na Visão do Engenheiro de Software:  
É um conjunto de Programas, documentos  
e dados que compoem um software de  
computador

Na Visão do Usuário: é a informação  
resultante, que de algum modo torna  
melhor o mundo do usuários.







## Tipos de Software

Software Comercial:

- ✓ Contas a Pagar, Folha de Pagamento, Contabilidade, ERPs





## Questions





## Processo de Software

- ✓ História
- ✓ Conceitos da Engenharia de Software
- **Processo de Software**
- **Modelos de Processo**
- ✓ Introdução Métodos Ageis
- ✓ Introdução Scrum
- ✓ Introdução UP
- ✓ Introdução RUP





## THE CHAOS MANIFESTO

*The Standish Group International,*

## RESOLUTION

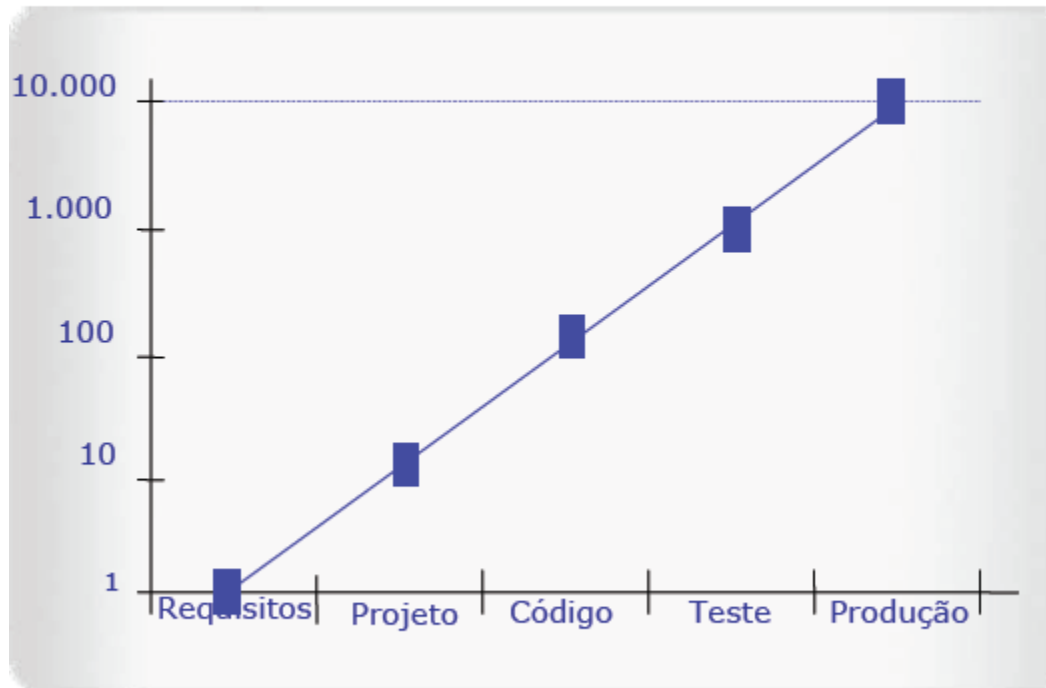
	2008	2010	2012
<b>Successful</b>	32%	37%	39%
<b>Failed</b>	24%	21%	18%
<b>Challenged</b>	44%	42%	43%

Project resolution  
results from CHAOS  
research for years





## Custo do Erro



- Quanto mais tarde descobrimos o erro, maior o custo.

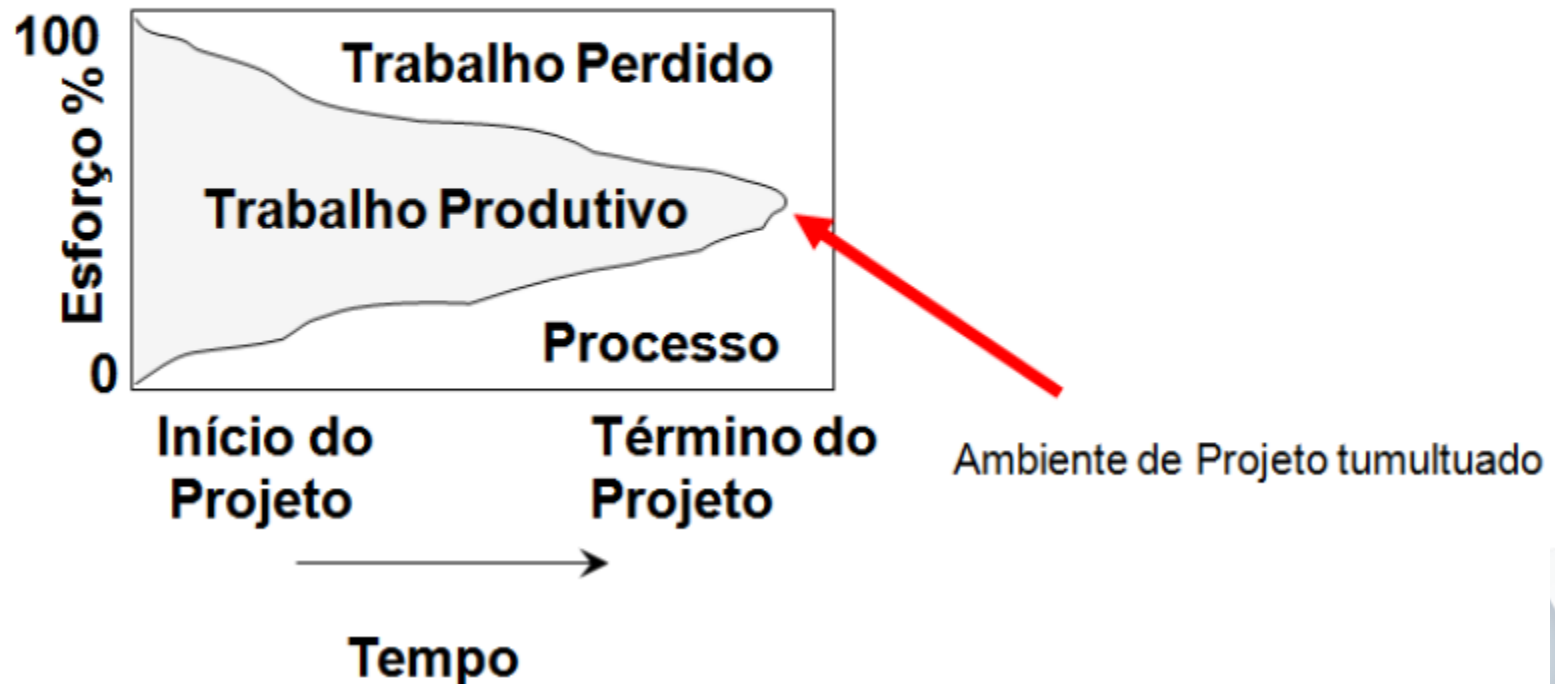
- Erro não identificado em cada fase, o custo para correção é multiplicado por 10.

- Erros em produção, além de possuírem custo financeiro alto, causam impactos significativos no negócio da empresa e na sua imagem.



# Processo de Software

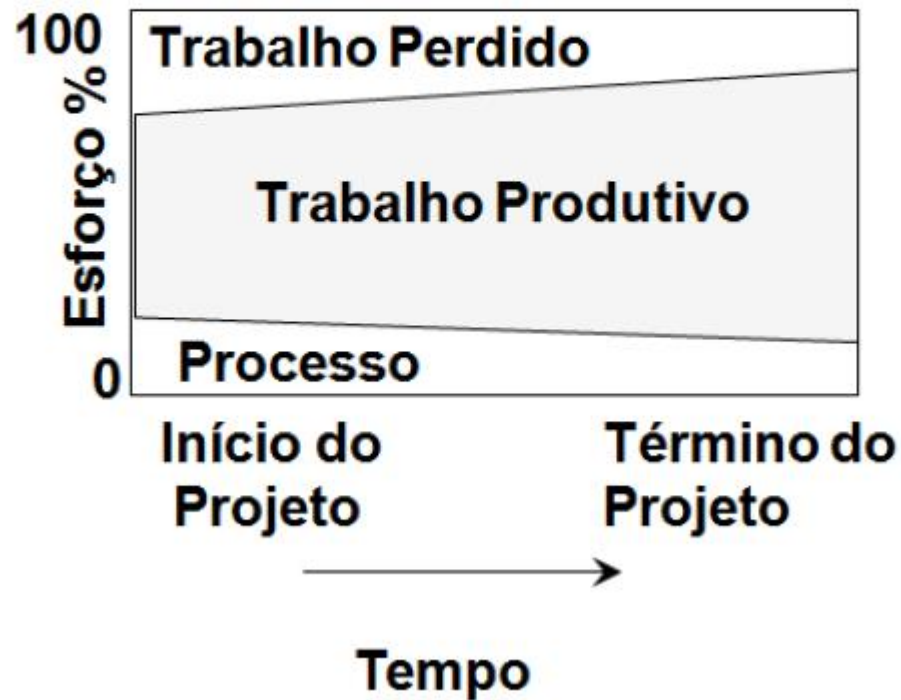
O que realmente acontece com os projetos que não focam em processo?





# Processo de Software

Projetos que focam atenção ao processo nas fases iniciais





# Processo de Software

## Processo

Conjunto de ações ou atividades que quando executadas geram um produto a partir da transformação de insumos.







# Processo de Software

## Processo

É um conjunto de passos ordenados, constituídos por atividades, métodos, práticas e transformações usado para atingir uma meta.





## Processo de Software

Conjunto de ações ou atividades que transformam Requisitos em um produto de Software, de acordo com as regras estabelecidas.





## Processo de Software

### Estrutura Básica:

- ✓ Especificação
  - ✓ Desenho
  - ✓ Validação
  - ✓ Evolução





# Processo de Software

## Processo de Software

O processo define:

- ✓ Quem deve fazer;
- ✓ O que é usado;
- ✓ Como deve ser feito;
- ✓ Quando deve ser feito;
- ✓ O que deve ser produzido;



## Processo e Produto

- ✓ A qualidade de um produto é altamente influenciada pela qualidade do processo utilizado no seu desenvolvimento e manutenção.
- ✓ Por que é correto focar o Processo? Dirigindo o foco somente no produto, deixa-se de lado: assuntos relacionados com a escalabilidade e o conhecimento de como fazê-lo melhor
- ✓ Dirigindo o foco no processo prevê-se: repetição de resultados e tendências futuras para os projetos e características do produto





# Processo de Software

## Processo e Produto

Qualidade do Produto - Modernamente considera-se que a qualidade do produto é conseguida de forma consistente, a longo prazo, a partir da qualidade do processo.

**Prevenção  
de Defeitos**

Antes

**Correção  
de Defeitos**





## Um Processo Imaturo

- ✓ Ad hoc: processo improvisado por profissionais e gestores.
- ✓ Não é rigorosamente seguido e o cumprimento não é controlado.
- ✓ Altamente dependente dos profissionais atuais.
- ✓ Baixa visão do progresso e da qualidade.





## Um Processo Imaturo

- ✓ A funcionalidade e a qualidade do produto podem ficar comprometidas para que prazos sejam cumpridos.
- ✓ Arriscado, do ponto de vista do uso de novas tecnologias.
- ✓ Custos de manutenção excessivos.
- ✓ Qualidade difícil de prever





## Um Processo Maduro

- ✓ Consistente com a maneira que o trabalho é realmente executado.
- ✓ Definido, documentado e melhorado constantemente:
- ✓ Compreendido; utilizado; vivo e ativo
- ✓ Apoio visível da alta administração e outras gerências.





# Processo de Software

## Um Processo Maduro

- ✓ Bem controlado - fidelidade ao processo é objeto de auditoria e de controle.
- ✓ São utilizadas medições do produto e do processo.
- ✓ Uso disciplinado da tecnologia.



## Benefícios de Um Processo Maduro

- ✓ Cerca de 85% dos problemas são causados pelo sistema de trabalho, não pelas pessoas.
- ✓ As pessoas desenvolvem mais suas potencialidades e são mais eficientes dentro da organização.





## Benefícios de Um Processo Maduro

- ✓ Através da definição, medição e controle do processo, melhorias têm mais sucesso e são mantidas ao longo do tempo.
- ✓ Há uma crescente introdução bem sucedida de tecnologias, técnicas e ferramentas.





## Ciclo de Vida





## Ciclo de Vida

É o conjunto de transformações porque podem passar os indivíduos de uma espécie para assegurar a sua continuidade





## Ciclo de Vida de Software **software development life-cycle (SDLC)**

Representação abstrata, esquemática e simplificada do processo de desenvolvimento do Software, desde a sua concepção até a sua descontinuidade ou substituição.

Mostrando a sequência dos passos e os produtos e sub-produtos gerados ao longo do processo.





## Ciclo de Vida

Representação abstrata, esquemática e simplificada do processo de desenvolvimento do Software, desde a sua concepção até a sua descontinuidade ou substituição.

Mostrando a sequência dos passos e os produtos e sub-produtos gerados ao longo do processo.





## Passos do ciclo de vida

- 1-Concebido a partir da percepção de uma necessidade;
- 2-Desenvolvido, transformado em um conjunto de itens entregue a um cliente;
- 3-Entra em operação, sendo usado dentro de algum processo de negócio, estando sujeito a atividades de manutenção, quando necessário;
- 4-Retirado de operação, ao final de sua vida útil

## Ciclo de Vida

Em cada ciclo de desenvolvimento, podem ser identificadas as fases de:

- ✓ Análise;
- ✓ Projeto;
- ✓ Implementação;
- ✓ Testes





# Processo de Software

Ciclo de vida	Percepção da necessidade			
	Desenvolvimento	Concepção		
		Elaboração		
		Construção	Desenho inicial	
			Liberação	Desenho detalhado
				Codificação
				Testes de unidade
			Testes alfa	
	Transição			
	Operação			
Retirada				

## Ciclo de Vida de um Software

### **Manutenção Evolutiva**

Envolve todas as mudanças, inclusões, exclusões, alterações e aprimoramentos efetuados em um sistema com o intuito de satisfazer as novas funcionalidades, ampliações e modificações solicitadas pelo usuário.

### **Manutenção Corretiva**

É diagnosticar e corrigir os problemas do sistema como, por exemplo, falhas de funcionalidades, erros software (código fonte), entre outros.



## Questions





## Modelos de Processo





Faculdade  
**IMPACTA**  
TECNOLOGIA



## Modelos de Processo

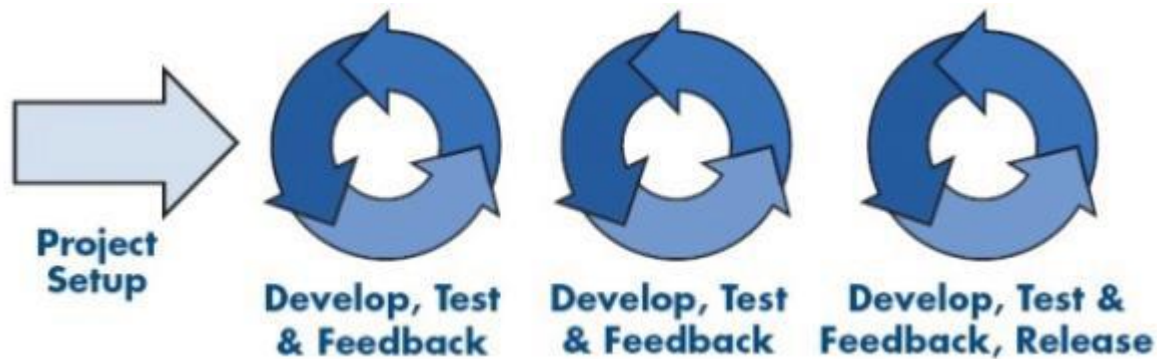
- ✓ Modelos Descritivos - Retratar como um processo é executado em um ambiente em particular;
- ✓ Modelos Prescritivos - Retratar como um processo poderia ser executado. É uma “espécie” de recomendação que pode ser adaptada ou melhorada;  
Ex: Cascata







## Iterativo X Interativo





Você sabe qual a diferença entre "iterativo" e "interativo"?

- **Iterativo:** que serve para iterar; repetido
- **Iteração:** repetição
- **Iterar:** tornar a fazer; repetir
- **Interativo:** que interage; que permite comunicação em dois sentidos
- **Interagir:** agir mutuamente (uma entidade agindo sobre outra e vice-versa)

---

URL of this site: [www.ime.usp.br/~pf/algoritmos/](http://www.ime.usp.br/~pf/algoritmos/)



## O modelo de ciclo de vida incremental e iterativo

- ✓ processo de desenvolvimento, divide o desenvolvimento de um produto de software em ciclos.
- ✓ cada ciclo de desenvolvimento, as fases de análise, projeto, implementação e testes.



## O modelo de ciclo de vida incremental e iterativo

- ✓ Cada um dos ciclos considera um subconjunto de requisitos.
- ✓ No próximo ciclo, um outro subconjunto dos requisitos é considerado para ser desenvolvido, o que produz um novo incremento do sistema que contém extensões e refinamentos sobre o incremento anterior.

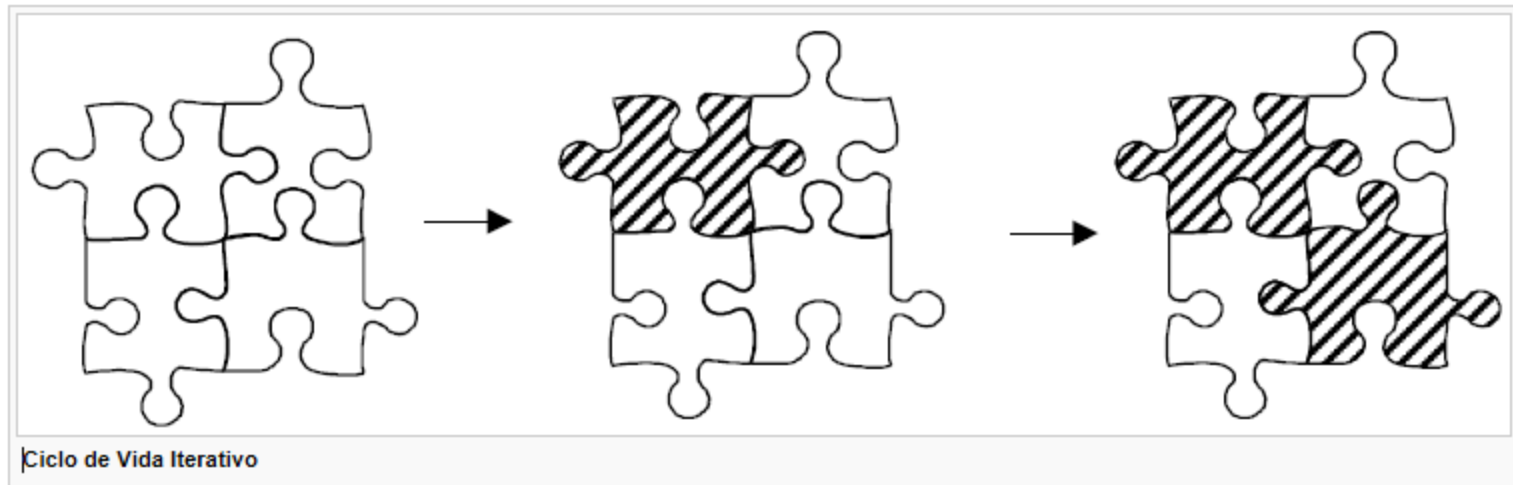


## O PROCESSO ITERATIVO

- ✓ Ideia de “melhorar (ou refinar) pouco - a - pouco” o sistema (iterações);
- ✓ A cada iteração identifica-se e especifica-se os requisitos relevantes, cria um projeto utilizando a arquitetura escolhida como guia, implementa o projeto em componentes e verifica se esses componentes satisfazem os requisitos.
- ✓ Se a iteração atinge os seus objectivos, o desenvolvimento prossegue com a próxima iteração



## O PROCESSO ITERATIVO



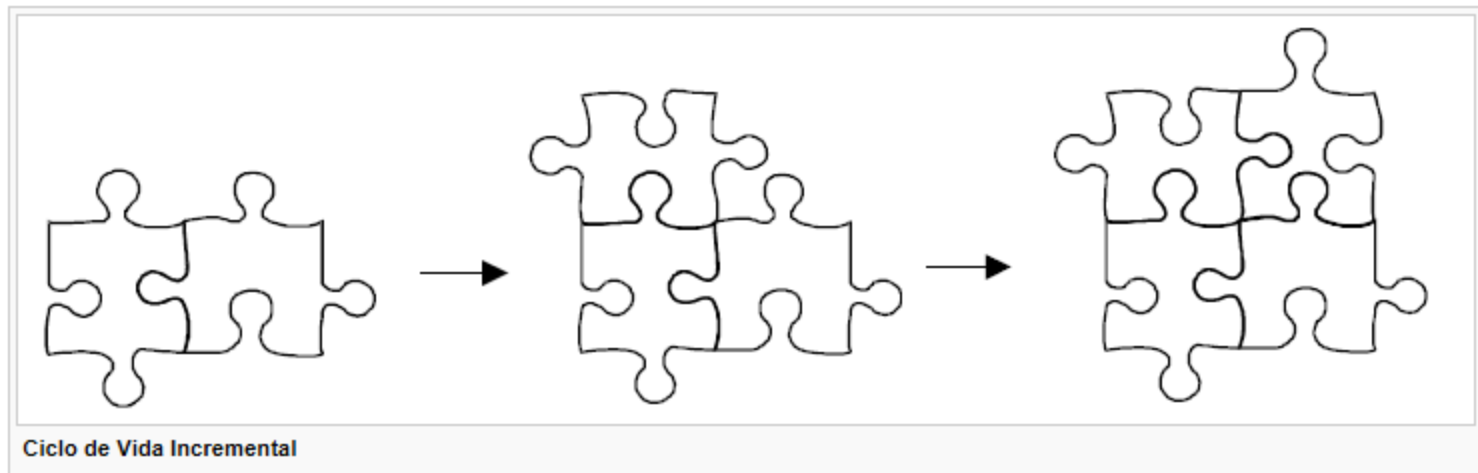


## O PROCESSO INCREMENTAL

- ✓ Ideia de “ aumentar (alargar) pouco-a-pouco ” o âmbito do sistema.
- ✓ Um incremento não é necessariamente a adição do código executável correspondente aos casos de uso que pertencem à iteração em andamento. Especialmente nas primeiras fases do ciclo de desenvolvimento, os desenvolvedores podem substituir um projeto superficial por um mais detalhado ou sofisticado. Em fases avançadas os incrementos são tipicamente aditivos.



## O PROCESSO INCREMENTAL





## Vantagens do processo incremental e iterativo

- ✓ Possibilidade de avaliar mais cedo os riscos e pontos críticos do projeto, e identificar medidas para os eliminar ou controlar;
- ✓ Os requisitos mudam com o tempo e um processo iterativo e incremental mantém frequentes os contatos com o cliente, o que ajuda a manter os requisitos sincronizados;



## Questions





## Modelos de Processo

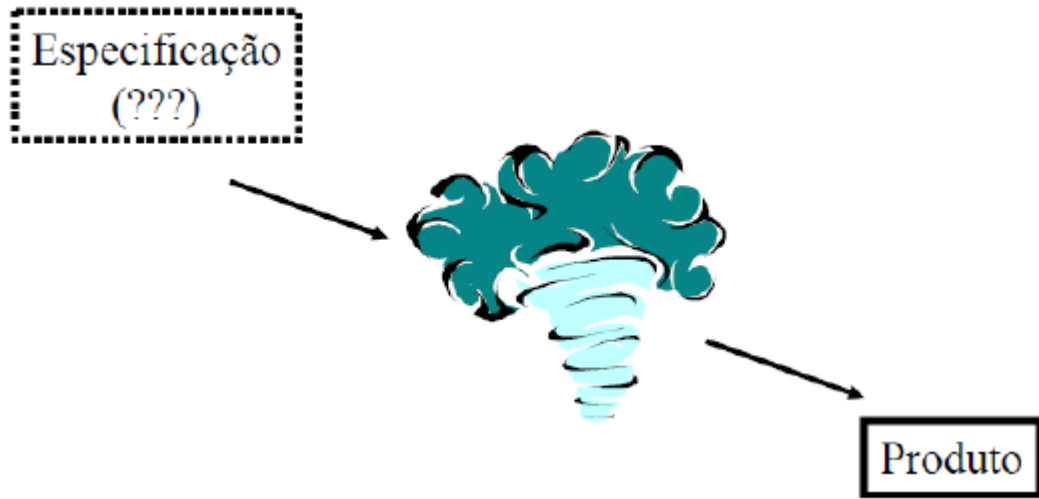
- ✓ Codifica-Remenda;
- ✓ Clássico ou CASCATA / Waterfall;
- ✓ V-Model
- ✓ Sashimi;
- ✓ Espiral;
- ✓ Prototipagem Evolutiva;
- ✓ Entrega por Estágios;
- ✓ Entrega Evolutiva
- ✓ RAD (Rapid Application Development)
- ✓ Orientado a Reuso
- ✓ RUP\* (Rational Unified Process)





# “Codifica-Remenda”

1. Especificação (ou nem isto)
2. Desenvolvedores começam a codificar
3. Nenhum processo definido é seguido





## “Codifica-Remenda”

- ✓ “Fala que vou fazendo”;
- ✓ Para alguns desenvolvedores, este modelo é atraente porque não existe nenhuma sofisticação técnica ou gerencial;
- ✓ Por outro lado é um modelo de alto risco, já que é impossível de gerir e que não permite assumir compromissos confiáveis





## Cascata ou Waterfall





# Cascata ou Waterfall

## Origem: Processo Industrial e Linha de Montagem





# Cascata ou Waterfall

- ✓ Sistemático e sequencial;
- ✓ Passa para o próximo passo apenas após o término completo do anterior;
- ✓ Não prevê feedback;
- ✓ Cada fase prevê a entrega de algum artefato



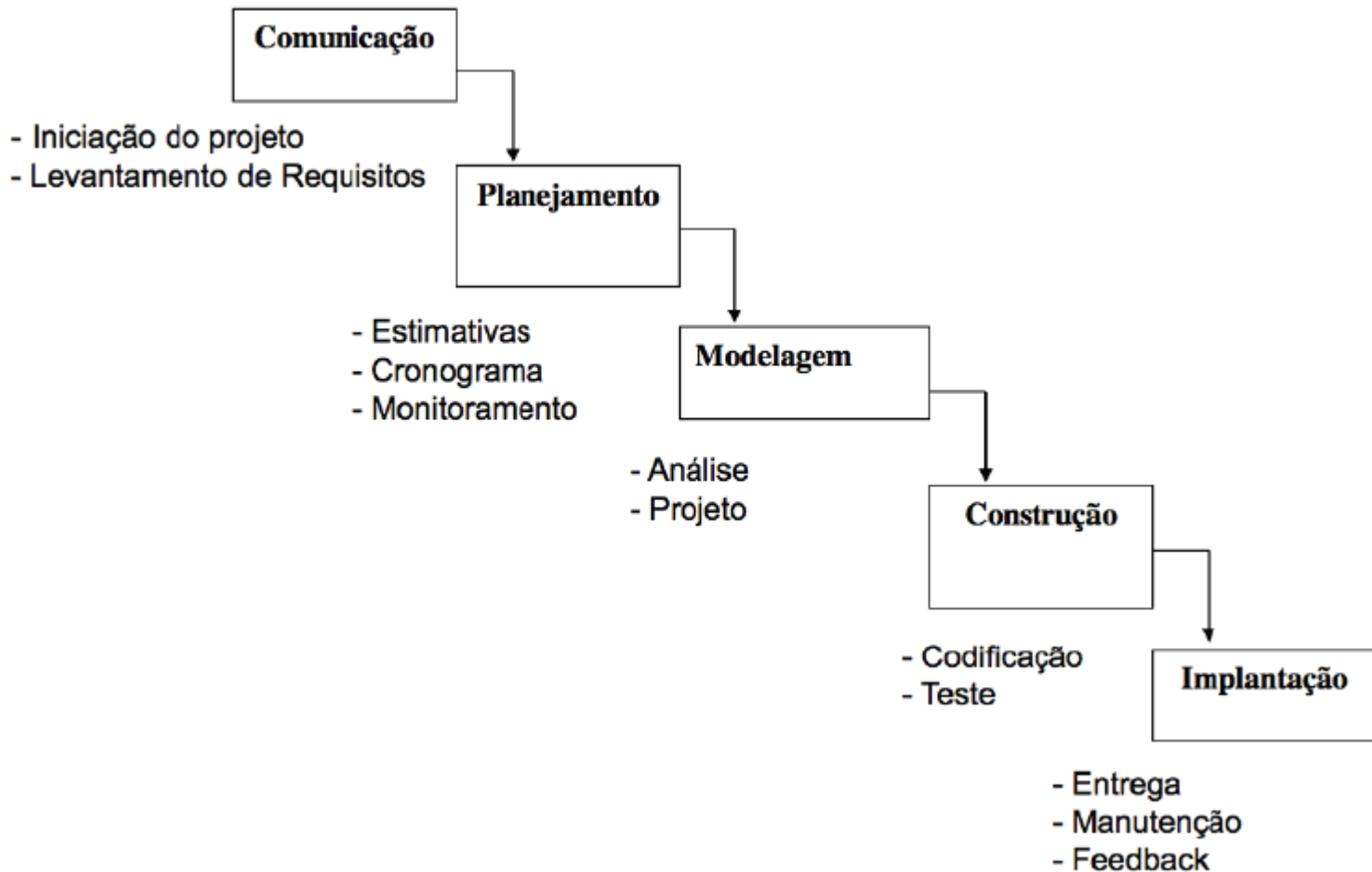


## Vantagens do processo incremental e iterativo

- ✓ Aceleração do tempo de desenvolvimento do projeto como um todo, porque a equipe que desenvolve o software trabalha de maneira mais eficiente quando pretende alcançar resultados de escopo pequeno e claro;
- ✓ Altamente motivador para a equipe de desenvolvimento (e o cliente) ver o software funcionando cedo.

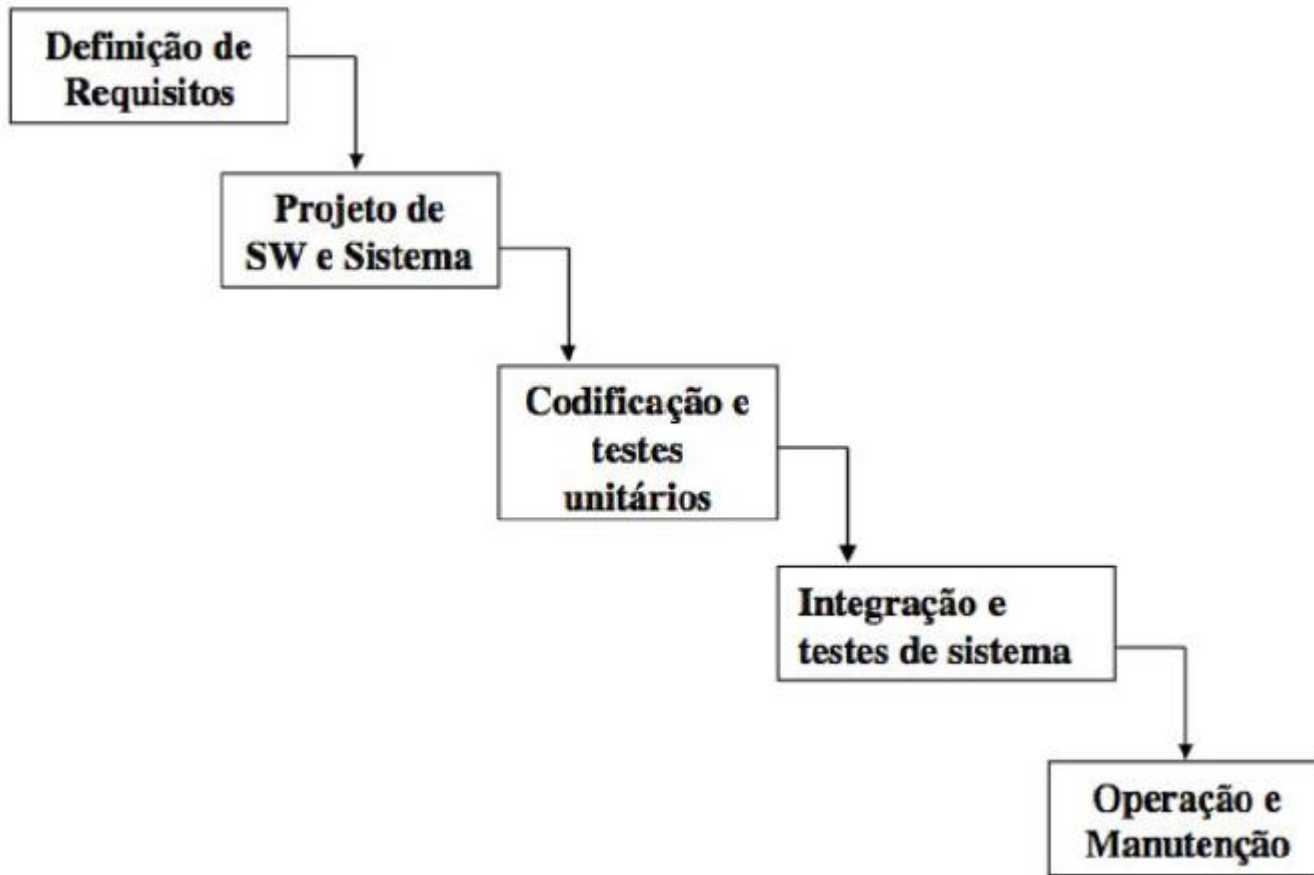


# Cascata ou Waterfall





# Cascata ou Waterfall



Sommerville





## Pontos Fortes

- ✓ Fixa pontos específicos para a entrega de artefatos;
- ✓ Segue uma sequência clara;
- ✓ É simples e fácil de aplicar, facilitando o planejamento.

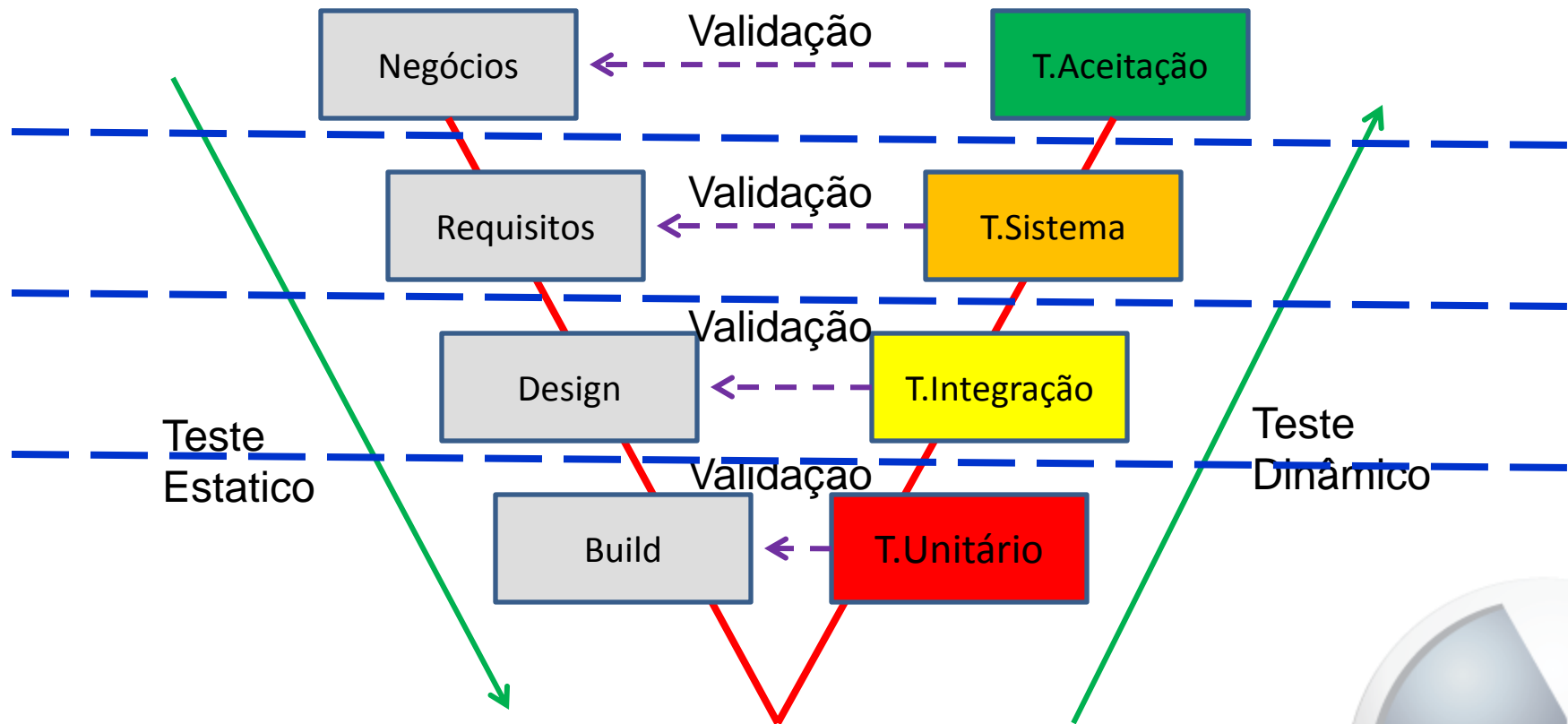




## Pontos Fracos

- ✓ Particionamento inflexível do projeto em fases distintas.
- ✓ Dificuldade em responder ou acomodar as mudanças de requisitos após o processo ter sido iniciado.
- ✓ Só é apropriado quando os requisitos são bem compreendidos.







- ✓ Definido por Paul Rook em 1980;
- ✓ Incluído na UK National Computing Center em 1990 com objetivo de melhorar a eficiência e efetividade do desenvolvimento de software;
- ✓ Apresentado como um modelo alternativo ao modelo Waterfall;
- ✓ Mostra a sequência típica de atividades de desenvolvimento (left-hand) e a sequência correspondente das execuções das atividades de testes (right-hand);

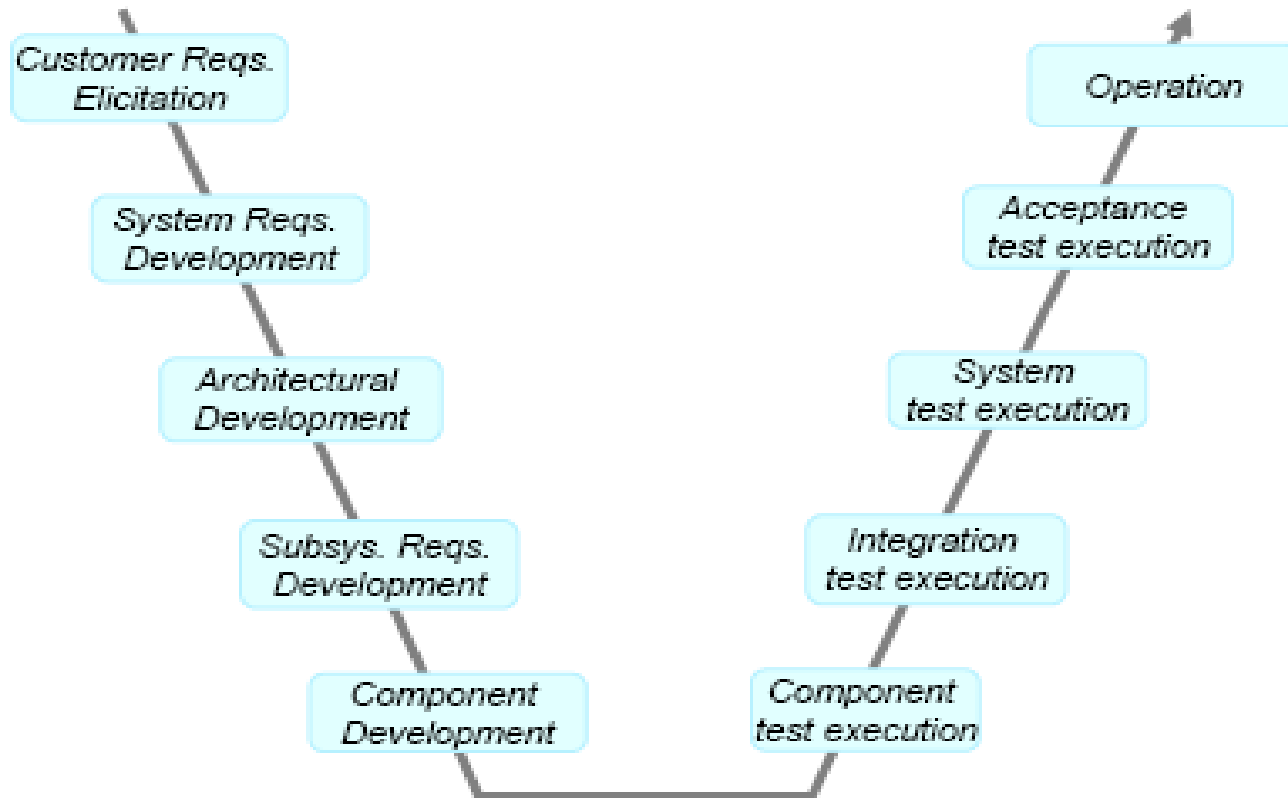


Figure 1: The Classic V-Model



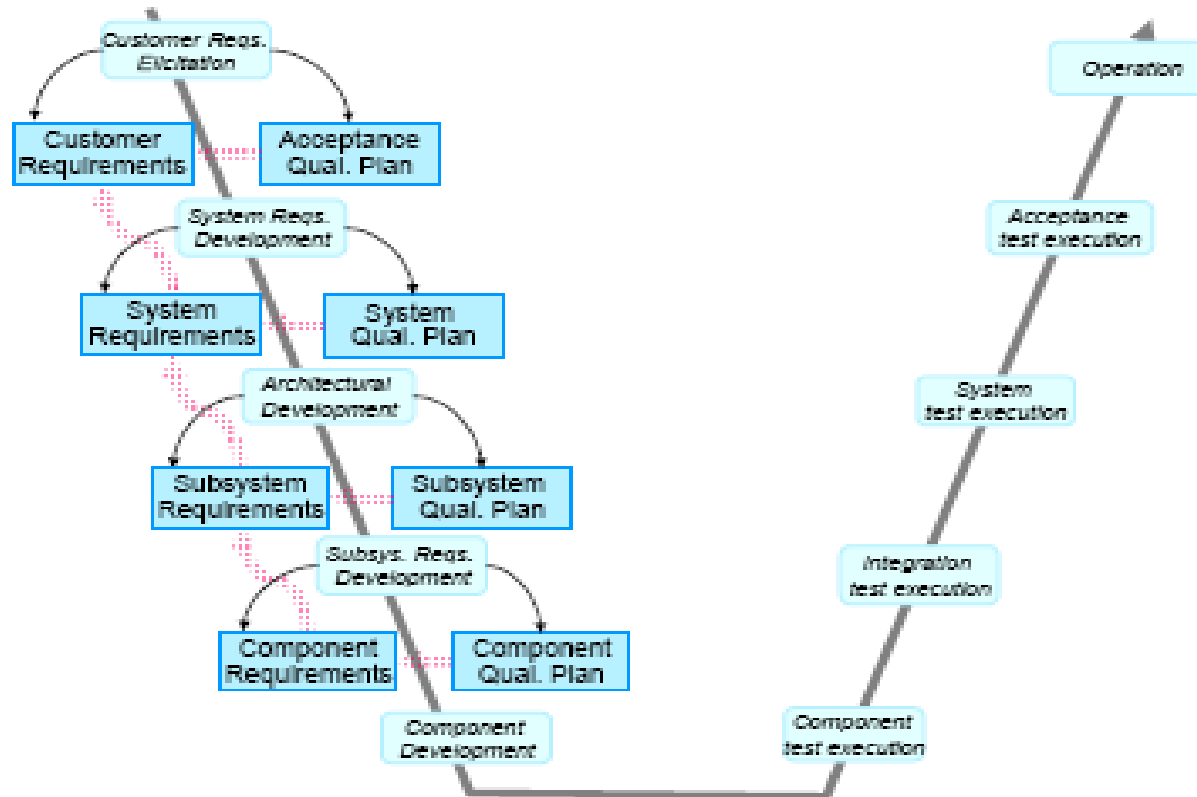
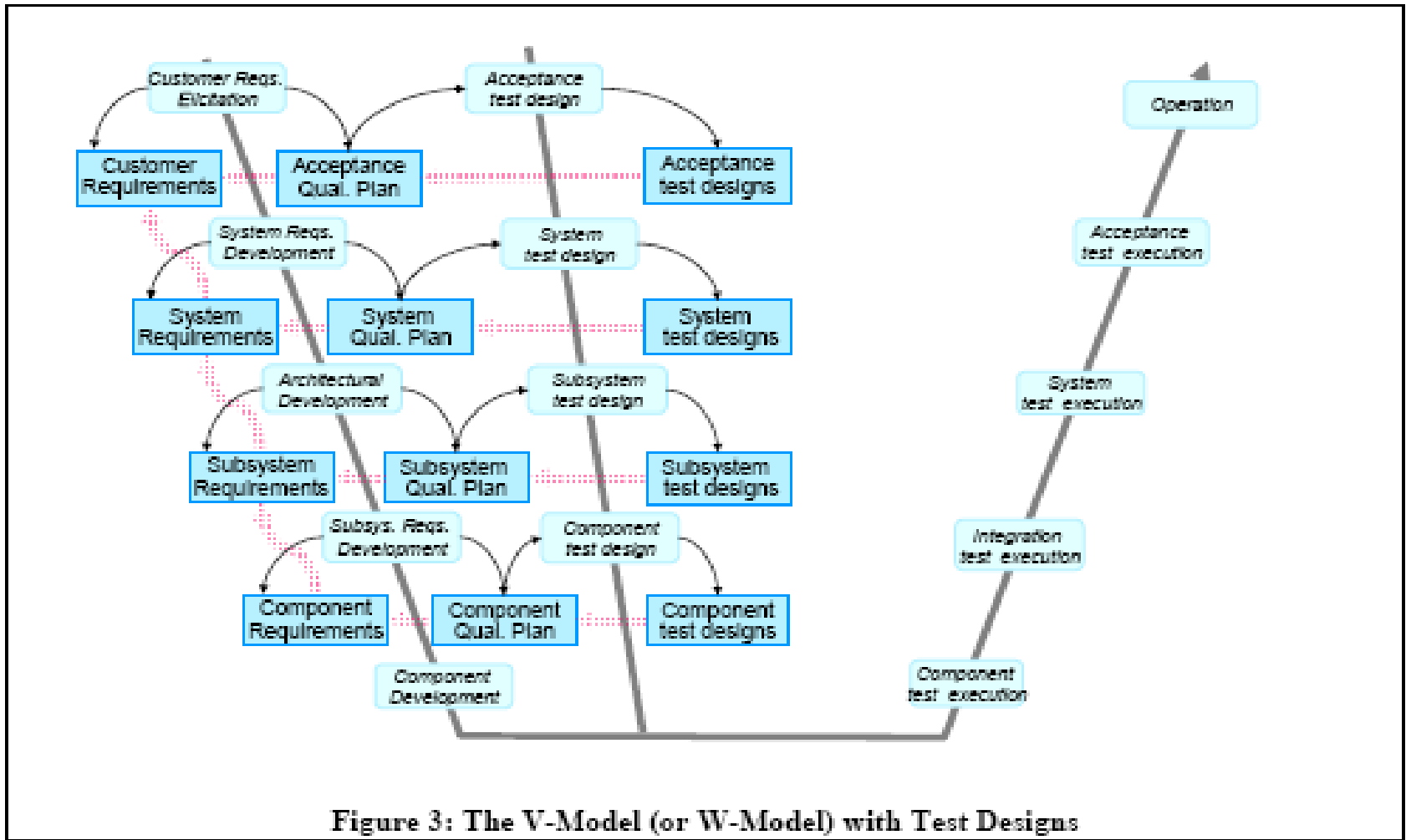
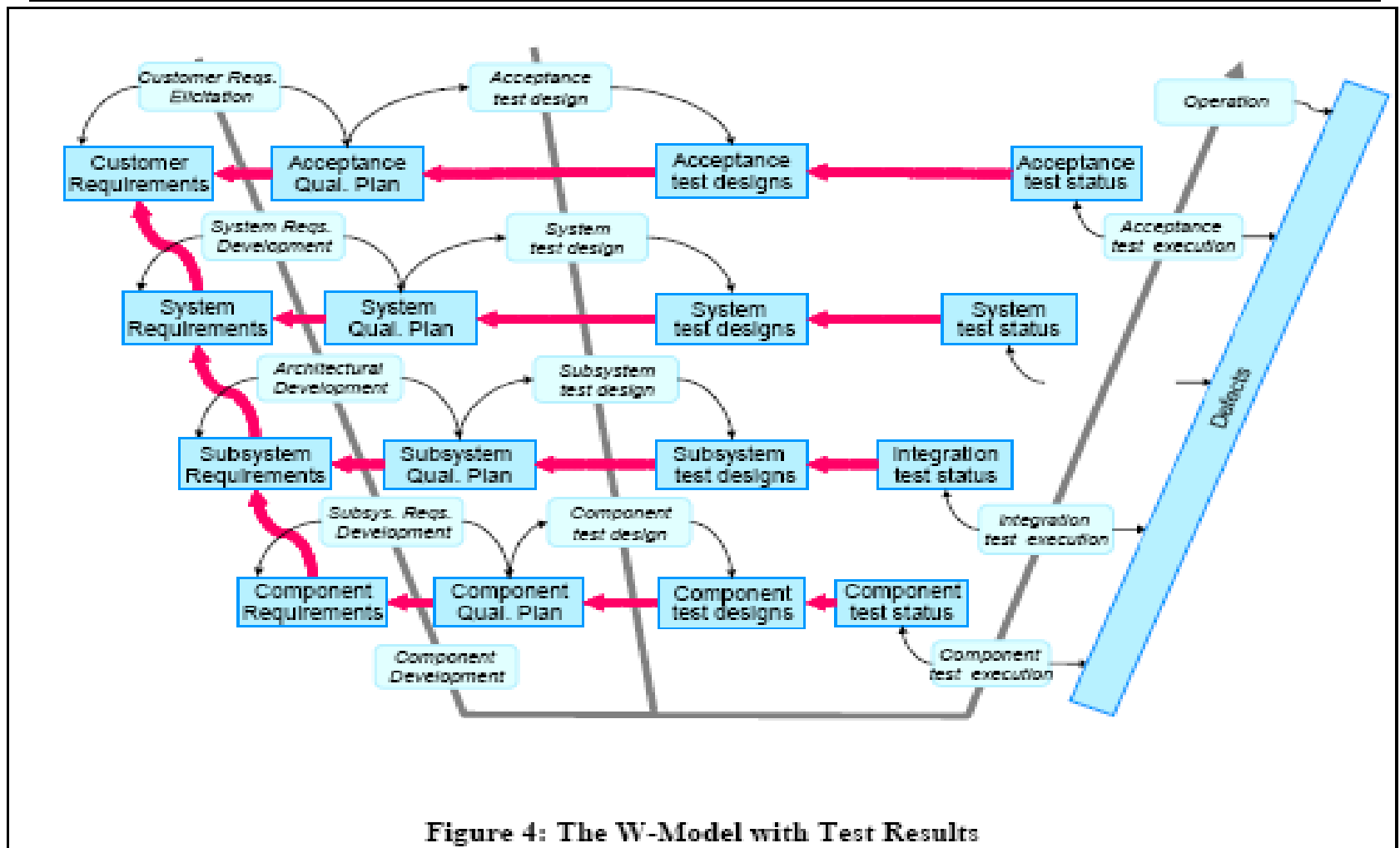
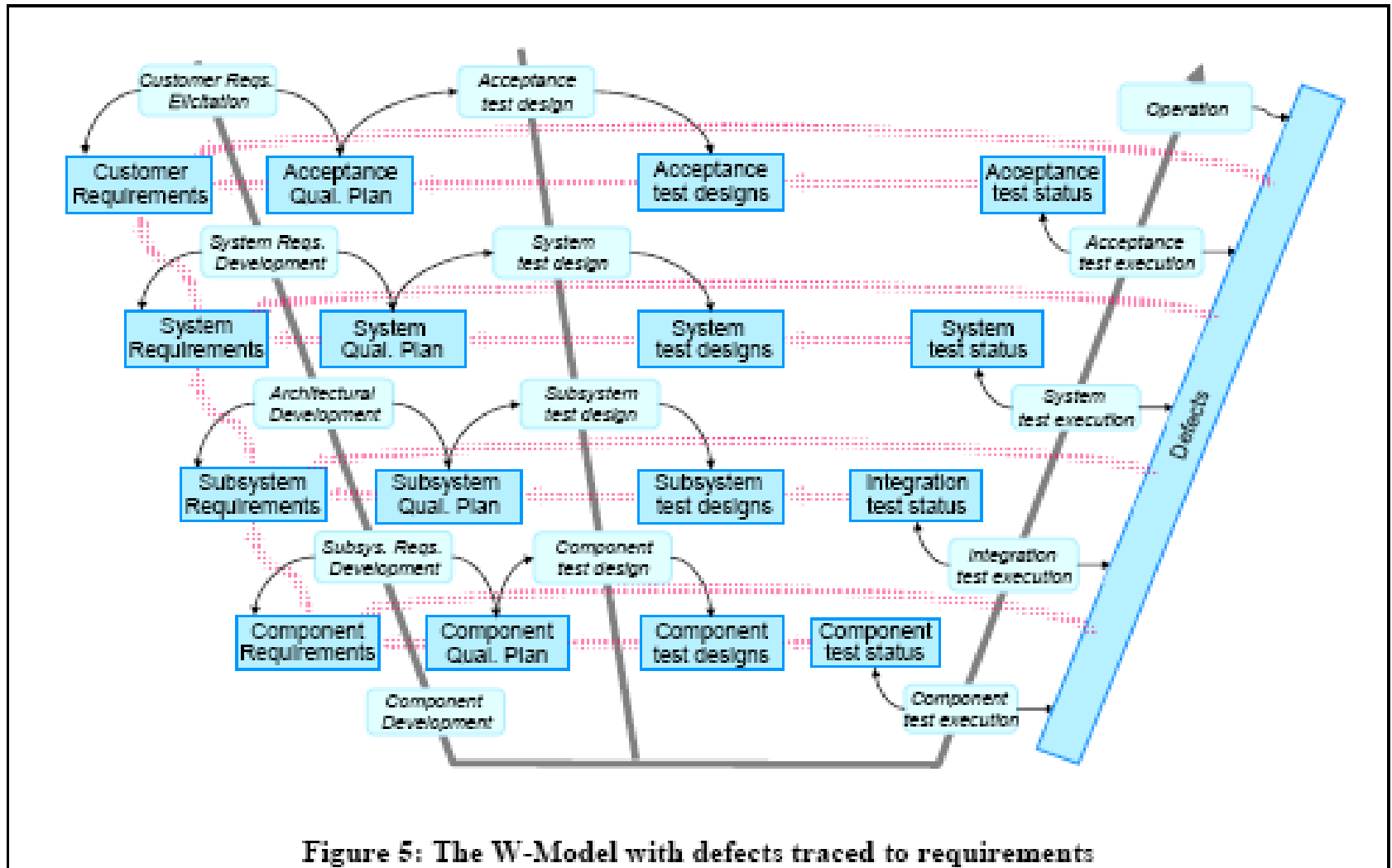


Figure 2: The V-Model with Qualification Plans









## Origem: Variação do Cascata/ Waterfall

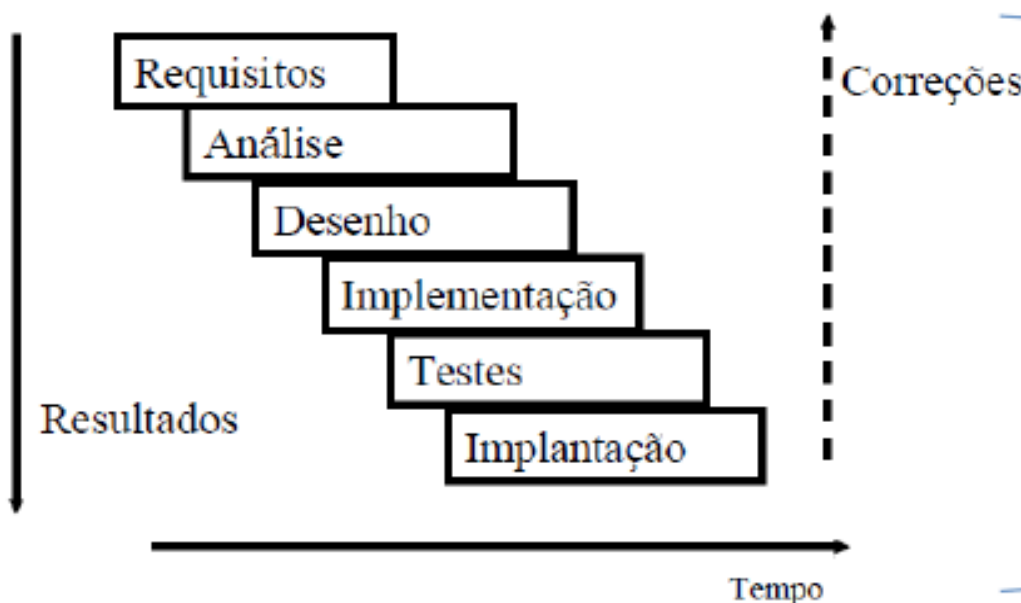


## Consiste de 6 fases:

- ✓ Requisitos
- ✓ Desenho e Arquitetura
- ✓ Desenvolvimento e Codificação
- ✓ Qualidade e Testes
- ✓ Implementação
- ✓ Manutenção e Suporte



- ✓ Permite que haja revisão e alteração de resultados em fases anteriores;
- ✓ A superposição das fases torna difícil gerenciar projetos



Especificações podem ser alteradas na implementação





Faculdade  
**IMPACTA**  
TECNOLOGIA

Questions





# Pós-Graduação

## Engenharia de Software



F a c u l d a d e  
**IMPACTA**  
T E C N O L O G I A

---

Obrigada

Marta Fuzioka

[mrtfuzioka@uol.com.br](mailto:mrtfuzioka@uol.com.br)