



Testes de Software

Prof^a. Patrícia Vilain

2020/1

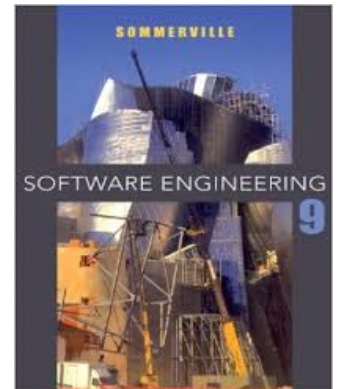


Introdução

O que é Engenharia de Software?

❑ É uma disciplina de engenharia que está relacionada com todos os aspectos da produção de software, desde os estágios iniciais da especificação até a manutenção. [Sommerville]

- processos técnicos de desenvolvimento de sw
- atividades como gerenciamento de projetos de sw
- desenvolvimento de ferramentas, métodos e teorias que dêem apoio à produção de sw.



Engenharia de Software

Áreas de conhecimento da Engenharia de Software listadas no SWEBOK (Software Engineering Body of Knowledge) 3.0:

- Software Requirements
- Software Design
- Software Construction
- Software Testing
- Software Maintenance
- Software Configuration Management
- Software Engineering Management
- Software Engineering Process

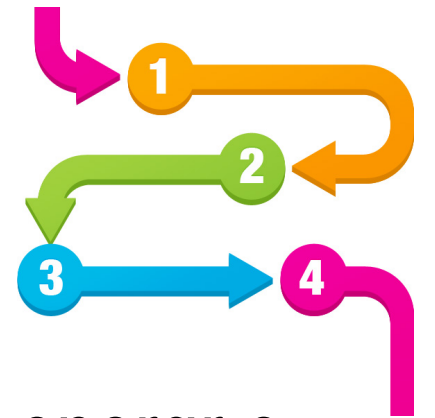
Engenharia de Software

... continuação

- Software Engineering Models and Methods
- Software Quality
- Software Engineering Professional Practice
- Software Engineering Economics
- Computing Foundations
- Mathematical Foundations
- Engineering Foundations

O que é Processo de Software?

- ❑ Um processo de software é um conjunto de atividades e tarefas relacionadas que transformam produtos de entrada em produtos de saída.
- ❑ Os processos de engenharia de software estão relacionados com as atividades para desenvolver, manter e operar o software, como requisitos, design, construção, testes, gerenciamento de configuração, entre outros processos.



Etapas do Processo de Software

Apesar de existirem diferentes processos para o desenvolvimento de software, no geral, os processos apresentam as seguintes atividades:

- **Especificação**
- **Projeto e Implementação**
- **Validação**
- **Evolução**

Processo Unificado (UP)

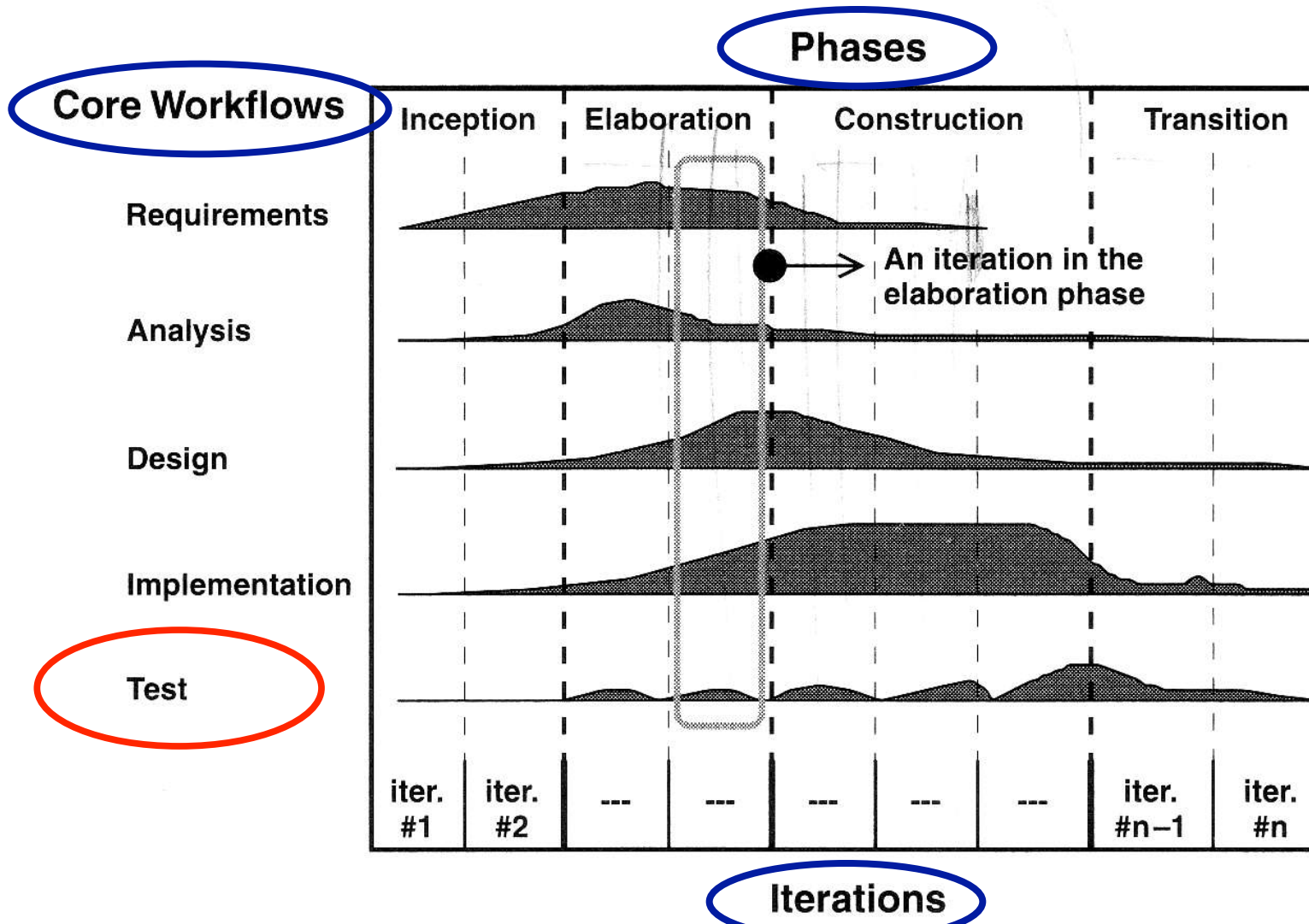
É um processo genérico de software que pode ser especializado

Utiliza a UML (Unified Modeling Language) como notação.

Características:

- Centrado na Arquitetura
- Direcionado por Caso de Uso
- Iterativo e Incremental

Processo Unificado



Métodos Ágeis

- Características dos métodos ágeis:
 - aplicam o desenvolvimento iterativo e evolucionário,
 - empregam o planejamento adaptativo,
 - promovem a entrega incremental,
 - e incluem outros valores e práticas que encorajam a agilidade.
- Incorporam técnicas de engenharia de software aprovadas, mas sem o *overhead* dos métodos tradicionais.

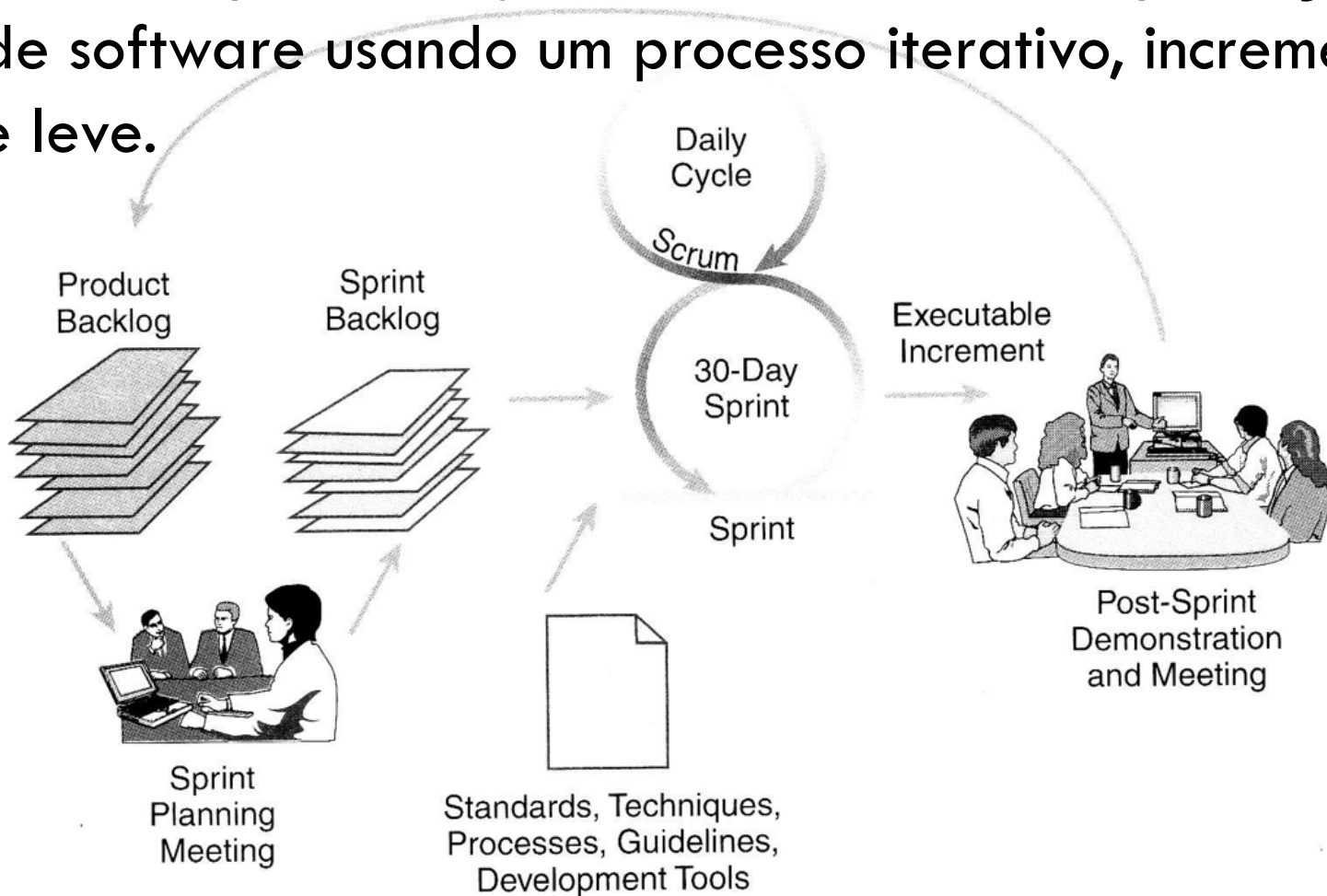
Métodos Ágeis

Alguns métodos bastante conhecidos:

- Scrum
 - XP
 - FDD
- Estes métodos apresentam diferentes ênfases e podem ser combinados.

Scrum

- Tem o objetivo de gerenciar e controlar a produção de software usando um processo iterativo, incremental e leve.



Scrum

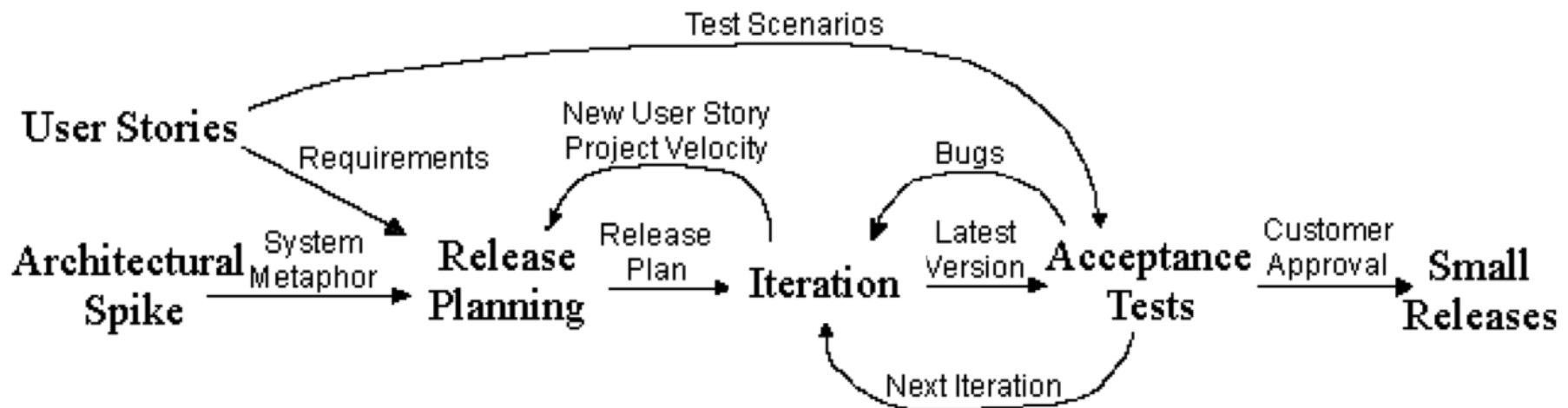
- Criado por Ken Schwaber e Jeff Sutherland nos anos 90.
- O Scrum define um framework de gerenciamento de projeto.
- Iterações de, no máximo, 1 mês.

Práticas do Scrum



XP (eXtreme Programming)

- Definido para dar suporte a pequenas equipes de desenvolvimento que trabalham com requisitos incertos e que são modificados.



XP

- XP foi criado por Kent Beck em 1996 e publicado em 2000.
- Iterações de, no máximo, 4 semanas.
- É baseado em 5 valores básicos: comunicação, simplicidade, feedback, coragem, respeito.

Práticas do XP

- Para seguir o XP é necessário adotar todas as suas práticas. As 12 práticas são relacionadas.

Jogo de Planejamento

Refatoração

Cliente Presente

Pequenas Versões

Programação em Par

Padrões de Codificação

Projeto Simples

Propriedade Coletiva

40 Horas Semanais

Testes

Integração Contínua

Metáfora do Sistema

Práticas do XP

Testes

- Testes são produzidos antes de qualquer código de produção ser escrito.
- Testes de aceitação são gerados assim que os requisitos são escritos.
- ❑ Se você não conhece os requisitos suficientemente bem para poder escrever os testes, como vai implementá-los?

Novas Práticas do XP (2004)

Práticas Primárias:

- Sentem-se juntos (Sit together)
- Equipe inteira (Whole team)
- Ambiente de trabalho informativo (Informative workspace)
- Trabalho energizado (Energized work)
- Programação em pares (Pair programming)
- Histórias (Stories)
- Ciclo semanal (Weekly cycle)
- Ciclo trimestral (Quarterly cycle)

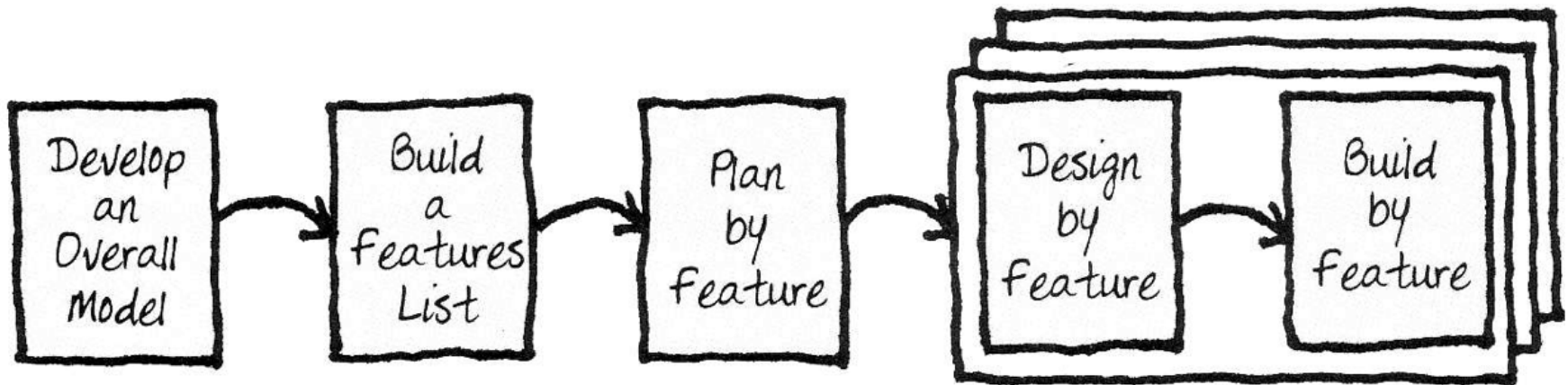
Novas Práticas do XP (2004)

Práticas Primárias: (continuação)

- Negligência (Slack) – incluir tarefas que podem não ser implem.
- Build a cada dez minutos (Ten-minute build)
- Integração contínua (Continuous integration)
- Desenvolvimento orientado a testes (Test-first programming)
- Projeto incremental (Incremental design)

FDD (Feature-Driven Development)

- Baseado em features (requisitos relacionados com o usuário, de comportamento da aplicação, ou internos)



FDD

- Criado por Peter Coad e Jeff De Luca e utilizado pela primeira vez em 1998.
- Iterações de, no máximo, 2 semanas.
- Bastante usado em projetos maiores, onde os requisitos estão sujeitos a mudanças em uma maneira mais controlada.

Práticas do FDD



Práticas do FDD (5)

Inspeções

- São feitas inspeções, durante e ao final de cada iteração, para assegurar a qualidade do projeto e do código.