

# Engenharia de Software

## Fundamentos de engenharia de Software

Ma. Vanessa Matias Leite

1

- Unidade de Ensino: 01
- Competência da Unidade: Conhecer os fundamentos da Engenharia de Software e da metodologia tradicional de desenvolvimento de produtos de software
- Resumo: Aplicar fundamentos de engenharia de software, metodologias ágeis e controle de versões no processo de desenvolvimento
- Palavras-chave: Engenharia de Software, metodologia ágil, XP, SCRUM, gerenciamento de configuração
- Título da Teleaula: Fundamentos de engenharia de Software
- Teleaula nº: 01

2

## Princípios da Engenharia de Software

- Organização hierárquica;
- Formalidade;
- Completeza;
- Dividir para conquistar;
- Ocultação;
- Localização;
- Integridade conceitual;
- Abstração

3

## Introdução à engenharia de software

4

## Fundamentos da Engenharia de Software

"Engenharia de *software* é uma disciplina cujo objetivo é produzir *software* isento de falhas, entregue dentro do prazo e orçamentos previstos, e que atenda às necessidades do cliente. Além disso, o *software* deve ser fácil de ser modificado quando as necessidades dos usuários mudarem".

5

## Tipos de Software

- Sistema;
- Aplicação;
- Engenharia/científico;
- Embarcado;
- Web;
- Aplicativos móveis;
- Inteligência artificial;

6

### Crise do Software

- Projetos que estouram o cronograma.
- Projetos que estouram o orçamento.
- Produto final de baixa qualidade ou que não atenda aos requisitos.
- Produtos não gerenciáveis e difíceis de manter e evoluir.

7

### Eternos Mitos

- A existência de um manual de procedimentos e padrões é suficiente para a equipe produzir com qualidade;
- Se o projeto estiver atrasado, sempre é possível adicionar mais programadores para cumprir o cronograma;
- Os requisitos mudam com frequência, mas sempre é possível acomodá-los, pois o software é flexível.
- O teste do software ou sua verificação formal pode remover todos os erros;

8

### Modelo de processo

Um modelo de processo de software é uma representação/abstração dos objetos e atividades envolvidas no processo de software.

Modo mais abrangente e fácil de representar o gerenciamento de processo de software e o progresso do projeto.

9

### Processos de Softwares

- Sequência de passos executados com um determinado objetivo.
- Um processo é uma receita que é seguida por um projeto.



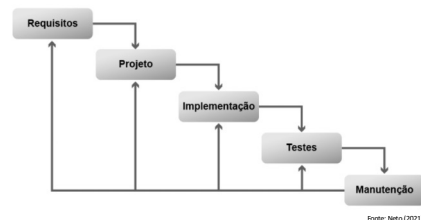
10

### Processos de Softwares



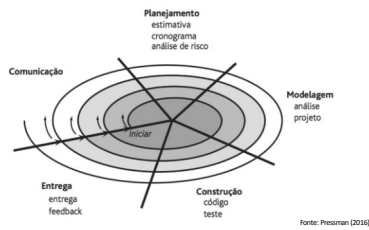
11

### Processos de Softwares



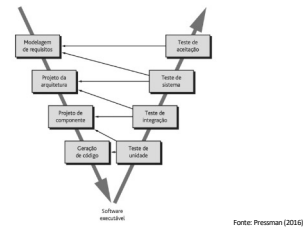
12

### Modelo Espiral



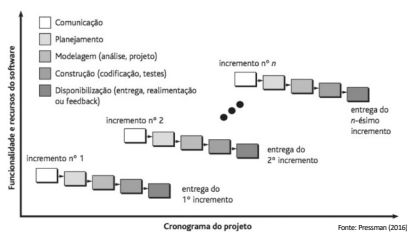
13

### Modelo V



14

### Modelo Incremental



15

### Metodologia Ágil

16

### Histórico

- Meados de 1990, motivados por uma reação adversa aos chamados "métodos pesados" de desenvolvimento de software;
- 2001 surgimento e a propagação de paradigmas de desenvolvimento de software ágeis.
- Manifesto ágil;

17

### Metodologia Ágil

- Planejamento incremental e muito iterativa;
- Menos ênfase nas definições das atividades e mais ênfase nos fatores humanos do desenvolvimento;
- Processos ágeis se adequam a mudanças;
- Mais envolvimento dos clientes;

18

### Motivações

- Menos centradas em documentação;
- São adaptativas em vez de predeterminantes;
- Aceitam mudanças ao longo do desenvolvimento de maneira natural;
- Orientados a pessoas e não a processos.

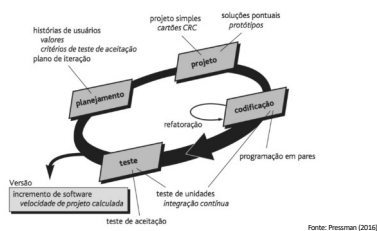
19

### Metodologia Ágil- *Extreme Programming* (XP)

- Rápido desenvolvimento;
- Requisitos se alteram constantemente;
- Valores do XP:
  - Comunicação;
  - Simplicidade;
  - Feedback;
  - Coragem;

20

### Metodologia Ágil- *Extreme Programming* (XP)



21

### Metodologia Ágil- *Extreme Programming* (XP)

- Equipe de Trabalho:
  - **Gerente de Projeto:** responsável pelos assuntos administrativos, incluindo relacionamento com o cliente;
  - **Coach:** responsável técnico pelo projeto;
  - **Analista de teste:** ajuda o cliente a escrever os testes de aceitação e fornece *feedback* para a equipe interna;

22

### Metodologia Ágil- *Extreme Programming* (XP)

- Equipe de Trabalho:
  - **Redator técnico:** ajuda a equipe de desenvolvimento a documentar o sistema;
  - **Desenvolvedor:** realiza análise, projeto e codificação do sistema. No XP, não há divisão entre estas especialidades.

23

### Metodologia Ágil- *Scrum*

- Método ágil mais usado atualmente.
- Aplica-se não só ao desenvolvimento de softwares como a qualquer ambiente de trabalho.
- Focado na gestão de projetos, tem como base o planejamento iterativo e incremental.
- É um framework utilizado na gestão de projetos e **desenvolvimento ágil de software**.

24

### Metodologia Ágil- *Scrum*

- Gestão e planejamento de Software;
- Reuniões Regulares (*Stand-up Meeting*);
- Projetos são divididos em ciclos;
- Entrega incremental;
- Envolvimento do cliente;

25

### Metodologia Ágil- *Scrum*

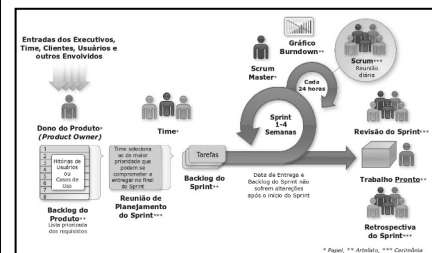
- Equipe de trabalho:
  - **Scrum Master**: trata-se de um facilitador do projeto;
  - **Product Owner**: é a pessoa responsável pelo projeto propriamente dito. Ele tem a missão de indicar os requisitos mais importantes a serem tratados nos *Sprints*.
  - **Scrum Team**: é a equipe de desenvolvimento, composta normalmente por seis a dez pessoas.

26

### Metodologia Ágil- *Scrum*

- **Product Backlog**: trata-se da lista que contém todas as funcionalidades desejadas para o produto.
- **Sprint Backlog**: lista de tarefas que a equipe deverá executar naquele *Sprint*.
- **Sprint**: divide o processo de efetiva construção do *software* em ciclos regulares, que variam de duas a quatro semanas.

27



28

### Metodologia Ágil- *Scrum*

- Kanban: "quadro de trabalho", organizar as atividades dos itens de Backlog da Sprint;
- Pôquer do planejamento;



29

### Quadro Scrum

30

### Quadro Scrum

- Meio pelo qual a equipe realiza a gestão visual das atividades do projeto.
- Não faz parte, oficialmente da metodologia, porém sua adoção foi feita em larga escala pelas equipes e, aparentemente, esse fato não altera sua importância.

31

### Quadro Scrum

- As divisões do quadro (formato matricial) são bem simples: uma coluna identifica a estória e as três colunas seguintes representam as tarefas relacionadas a esta estória que estão: a fazer (*to do*), em execução (*doing*) e feita (*done*).
- Com essa disposição, cada linha do quadro representa uma estória e suas respectivas tarefas, que são, na verdade, extraídas do backlog do produto e que foram selecionadas para uma determinada Sprint.

32

### Quadro Scrum

- Os papéis coloridos colados no quadro são post-its.
- Representam uma determinada característica ou estado daquela tarefa e contêm sua identificação resumida.
- A cor do post-it pode representar uma certa atenção ao estado daquela tarefa;
- A critério da equipe, uma determinada cor pode identificar um determinado membro da equipe.

33

Prioridade da tarefa:  
MA: Muito alta  
A: Alta  
M: Média  
B: Baixa  
MB: Muito baixa

A qual estória esta tarefa pertence?

Aqui vai a descrição da tarefa, escrita de forma reduzida e objetiva.

Nome

O nome do responsável pela execução da tarefa.

Esforço para executar a tarefa.  
1: Algumas horas  
3: Meio dia  
5: 1 dia  
8: De 1,5 dia a 2 dias

Tempo estimado para execução da tarefa.

Fonte: Netto (2021)

34

PROJETO/EQUIPE: EQUIPE SCRUM MARAVILHOSA					
	Pendência	A fazer	Fazendo	Em revisão/garantia de qualidade	Faltou!
História do usuário 1					
História do usuário 2					
História do usuário 3					
História do usuário 4					

Fonte: Netto (2021)

35

### Gestão de Configuração

36

## Gestão de Configuração

"Arte de identificar, organizar e controlar modificações no software que está sendo criado, maximizando a produtividade e reduzindo os erros" [Pressman, 2011]

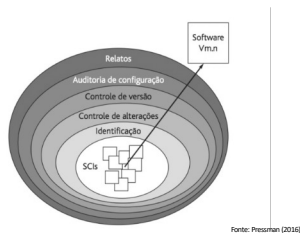
37

## Planejamento de Gerenciamento de Configuração

- Definir o que será gerenciado e o esquema que será usado para identificar as entidades;
- Estabelecer o responsável;
- Estabelecer políticas de gerenciamento de configuração;
- Especificar de ferramentas;
- Descrever a estrutura do banco de dados;

39

## Gerenciamento de Configuração



40

## Gerenciamento de Configuração

- Repositório;
- *Baselines* (linhas de base);
- *Branches*;

41

## Item de configuração

Elemento unitário ou um grupo de elementos para efeito de controle de versão.

- Código
- Documentação
- Diagramas, planos, ferramentas, casos de teste e etc

42

## Identificação de item de configuração

- Manter a rastreabilidade;
- Documentos que podem ser úteis devem ser controlados pelo gerenciamento de configuração;
- Esquemas de nomes hierarquizados;

43

### Banco de Dados de Configuração

- Registrar todas as informações relevantes sobre a configurações de sistemas e os itens de configuração;
- Avaliar o impacto das mudanças no sistema;
- Registro e recuperação de informações de projeto;

44

### Gerenciamento de Versões e Releases

- Preocupa-se com a identificação e a rastreabilidade das versões de um sistema;
- Uma versão é uma instância de um sistema, que difere, de alguma maneira, de outras instâncias;
- Versões com pequenas diferenças são algumas vezes chamadas de variantes;

45

### Gerenciamento de Versões e Releases

- Releases: Versão distribuída aos clientes;
- Os releases devem incorporar novas funcionalidades ou ser planejado para uma plataforma diferente de hardware;
- Há normalmente muito mais versões de um sistema do que liberações;

46

### Identificação de Versões

- Numeração de versões;
- Identificação baseada em atributos;
- Identificação orientada a mudanças;

47

### Ferramentas

**GitHub**Fonte: <https://bit.ly/2W5Aqpl>

48

### Recapitulando

49



### Recapitulando

- Conceito da Engenharia de Software;
- Modelos de Processos;
- Metodologia ágil:
  - XP
  - SCRUM
- Gestão de Configuração;

50



51