

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

Introdução à engenharia de software

3

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”.

4

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.

5

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;

6

Tipos de Softwares

- Software básico;
- Software de tempo real;
- Software comercial;
- Software científico e de engenharia;
- Software embutido ou embarcado;
- Software pessoal;
- Jogos;
- Inteligência artificial;

7

Modelos de Processos

8

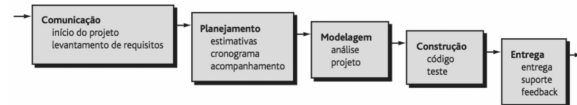
Processos de Softwares

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



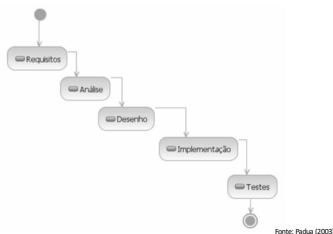
9

Processos de Softwares



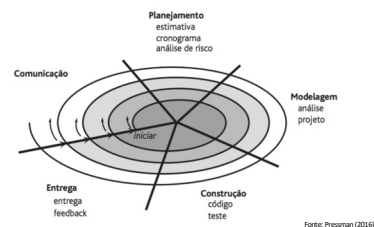
10

Modelo Cascata

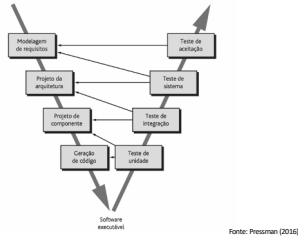


11

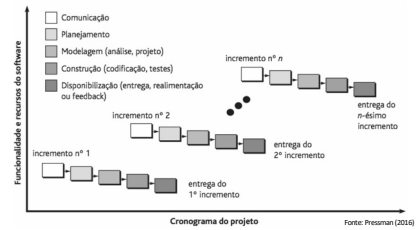
Modelo Espiral



12

Modelo V

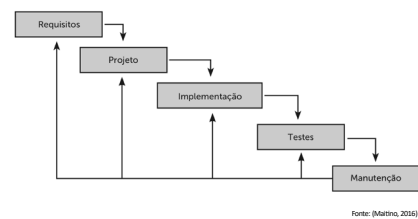
13

Modelo Incremental

14

Metodologia Ágil

15

Metodologia Tradicional

16

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;

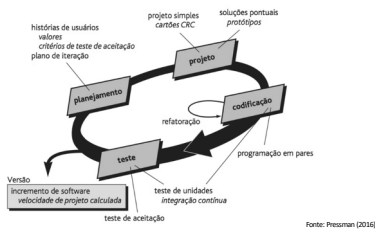
17

Metodologia Ágil- *Extreme Programming* (XP)

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

18

Metodologia Ágil- Extreme Programming (XP)



19

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;

20

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.

21

SP1

22

- Levantar pontos frágeis da metodologia atual.
- Planejar a introdução de práticas do XP relacionadas aos princípios da comunicação e *feedback*.
- Adotar práticas contínuas de aprimoramento do modelo e de encantamento de novos clientes.

23

Levantar pontos frágeis da metodologia atual:

- Falta de comunicação com o cliente;
- O cliente não faz parte do processo, logo não conhece as funcionalidades;
- as etapas do processo devem ser integralmente concluídas ;

24

Planejar a introdução de práticas do XP relacionadas aos princípios da comunicação e *feedback*:

- Chamar o cliente ao projeto;
- Nova maneira de coletar e tratar os requisitos
- O código coletivo, a *stand up meeting* e o uso de metáforas também inspiram relativa facilidade em suas implementações.

25

Adotar práticas contínuas de aprimoramento do modelo e de encantamento de novos clientes:

- Colher percepções e sugestões da equipe sobre sua rotina.
- Treinamento dos novos desenvolvedores que venham a ingressar na instituição;

26

Dúvidas?

27

SCRUM

28

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**.

29

Metodologia Ágil- *Scrum*

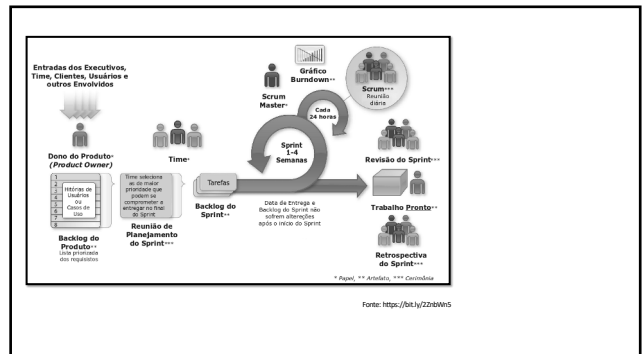
- Gestão e planejamento de Software;
- Reuniões Regulares (*Stand-up Meeting*);
- Projetos são divididos em ciclos;
- Entrega incremental;
- Envolvimento do cliente;

30

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.

31



32

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.

33

Metodologia Ágil- Scrum

- Kanban: "quadro de trabalho", organizar as atividades dos itens de Backlog da Sprint;
- Daily Scrum: breve reunião de no máximo 15 minutos com todos os participantes em pé.
- Sprint Review Meeting: funcionalidades implementadas e o que foi alcançado neste Sprint.



Fonte: https://bit.ly/396ac7f

34

Gestão de Configuração – Parte I

35

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]

36

Gerenciamento de Configuração

- Define procedimentos e padrões para gerenciar um produto de software.
- Agrega ao processo de gerenciamento de qualidade.
- Possibilita o versionamento controlada.
- Permite o retorno até uma determinada *baseline*.
- Visa uma versão de produto entregável (*Deliverable*).

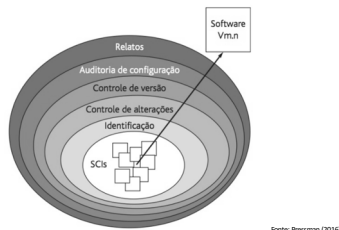
37

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;

38

Gerenciamento de Configuração



39

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

40

Identificação de item de configuração

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

41

Gestão de Configuração – Parte II

42

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;

43

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;

44

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;

45

Identificação de Versões

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

46

Ferramentas



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

47

SP2

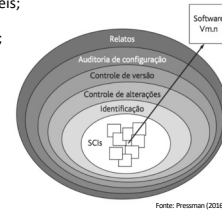
48

- Levantar os pontos importantes para implantar uma gerência de configuração;
- Planejamento para a implantação;
- Modo de identificação das versões;

49

Planejamento para a implantação:

- Estabelecer responsáveis;
- Definir as ferramentas;
- Definir as políticas;



50

**É possível utilizar
duas metodologias
ágeis em conjunto?**

51

Recapitulando

52

Recapitulando

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

53



54