Disciplina: Processo de Desenvolvimento de Software

Aula 10: O processo iterativo e incremental

Apresentação

Nesta aula, iremos demonstrar outros modelos de desenvolvimento de software.

Como vimos anteriormente, os modelos seguiam um padrão de organização onde se criavam etapas ou módulos caracterizados pelas ações que se tomariam durante um período determinado. Esses períodos se caracterizavam por fases que, após serem finalizadas, davam inicio a uma outra fase, com características diferentes.

Cada fase representava uma ação que possuía um inicio e um fim, mesmo nos casos de realimentação.

Para os modelos aqui apresentados, não estão somente levados em conta o planejamento e ação, mas também os valores e os recursos envolvidos.

Objetivos

- Conhecer outros processos utilizados no processo de desenvolvimentos de *software*;
- Entender as vantagens dos modelos e suas limitações;
- Analisar as etapas iniciais do processo de desenvolvimento de *software* e compará-los.

Processo de desenvolvimento Ágil

Método Ágil

É um conjunto de diretrizes e metodologias que cria uma estrutura conceitual para desenvolver projetos de desenvolvimento de software.

Baseado em um manifesto criado por programadores veteranos que já tinham passado por inúmeras experiências diferentes no campo de desenvolvimento de software, o **Manifesto Ágil <http://agilemanifesto.org/iso/ptbr/manifesto.html>** tem como foco as pessoas e não as ferramentas.

Fonte: Shutterstock/Llin Sergey.

Método XP

Também conhecido como eXtreme Programming, é um método que pertence à metodologia ágil de desenvolvimento de software.

O modelo propõem uma série de práticas focados em pessoas, ou seja, na equipe de desenvolvimento.

É baseado em 5 valores:

Comunicação

2

Coragem

3

Feedback

4

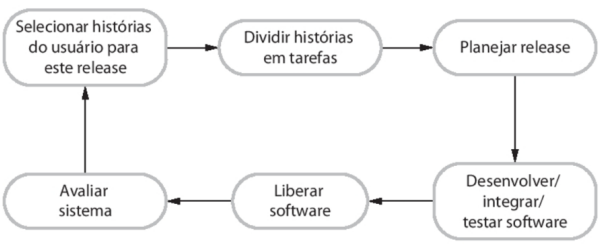
Respeito

5

Simplicidade

Ciclo de um release em XP

Como apresentado por Sommerville, “em XP, os requisitos são expressos em formas de cenários (chamados de histórias do usuário), que são implementados diretamente como uma série de tarefas. Os programadores trabalham em pares e desenvolvem testes para cada tarefa antes de escreverem o código. Quando o novo código é integrado ao sistema, todos os testes devem ser executados com sucesso.”



Vejamos algumas práticas do método XP:

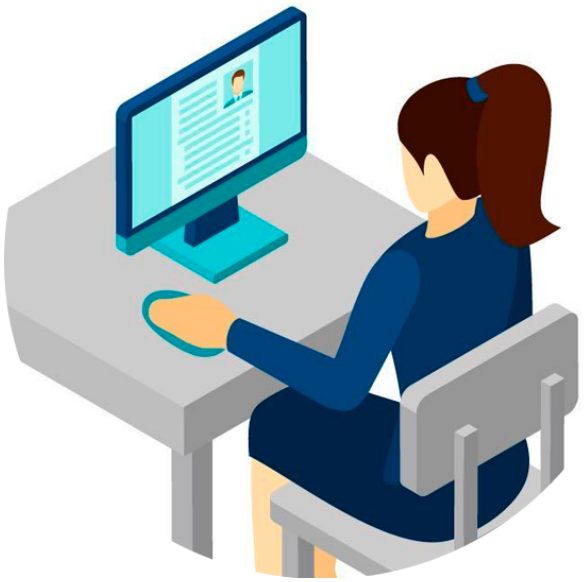
Reuniões em pé

Utilizadas para não perder o foco no assunto.

Programações em par

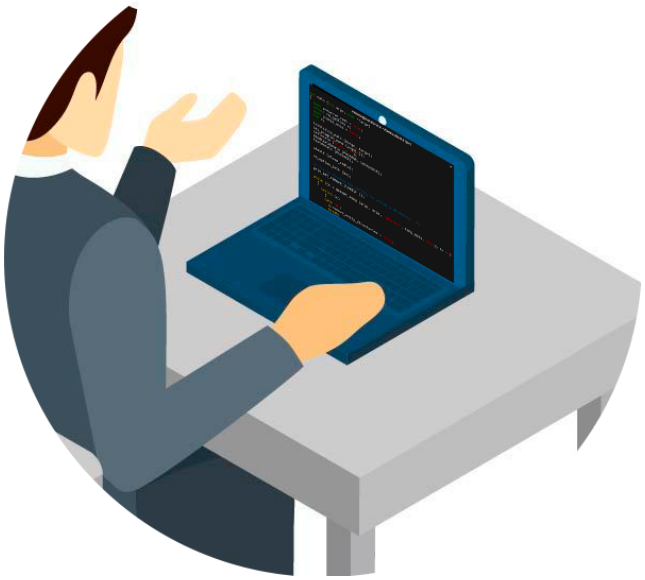
Todo o código é implementado por uma dupla, usando o mesmo computador. Ambos com o objetivo de resolver o mesmo problema.

Um terá o papel do condutor, digitando o código, e o outro o papel de navegador, com o objetivo de revisar o código de forma a evitar erros de programação ou sugerir melhores estratégicas de implementação.



Desenvolvimento guiado por testes

Desenvolvimento guiado por testes: Trabalhar com teste automatizado de forma acompanhar se as funcionalidades implementadas atendem ao requisito e ao cliente. Dentre os testes podemos trabalhar com os testes de unidade e os testes de aceitação.



Posse coletiva

O código fonte não pertence a ninguém, é de todos e todos podem utilizá-lo sem necessidade de permissão.

Pequenas versões

Pequenas versões aceitas pelo cliente ajudam na aceitação do programa completo.

Ritmo sustentável

Utilizar o tempo de trabalho dentro do especificado. Sem horas adicionais. (40 horas por semana).

Padrão de codificação

Após a formação da equipe, deverá ser estabelecida algumas regras, como o padrão de desenvolvimento a ser usado. A escolha de um padrão visa um rápido entendimento dos códigos uns dos outros.

Método Scrum

Metodologia que tem como filosofia o Manifesto Ágil.

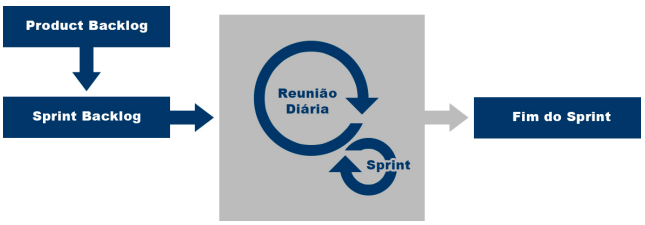
Possui papel bem definido para as atividades durante todo o processo. Uma vez levantadas as questões a serem trabalhadas, é determinado um período de tempo para a realização de um determinado requisito.

Durante esse intervalo, são feitas reuniões diárias para acompanhamento do andamento das atividades.

Os Três Pilares Fundamentais do Scrum

- Transparência: Transparência dos processos, requisitos de entrega e status.
- Inspeção: Inspeção constante de tudo o que está sendo realizado.

- Adaptação: Adaptação, tanto do processo, quanto do produto às mudanças.



Fundamentos Básicos do Scrum

1. Papéis

1.1- Dono do Produto (Product Owner): É o responsável por gerenciar o Backlog do Produto e garantir o valor do trabalho executado pela Equipe de Desenvolvimento.

1.2 - Scrum Master: É a pessoa que mais conhece o Scrum dentre todos os papéis. Ele tem o papel de papel de facilitador, ajudando a equipe a resolver problemas, realizando melhorias no uso do Scrum, além de manter a equipe focada em suas tarefas. Vale ressaltar que o Scrum Master não é um gerente.

1.3 – Time de Desenvolvimento (Team Scrum): É a equipe (time) responsável pelo desenvolvimento do projeto. Os membros da equipe deve ser multidisciplinar e multifuncional, possuindo todo conhecimento necessário para a criar um incremento no trabalho. Geralmente o time de desenvolvimento é constituído por três a nove pessoas. Tanto o Scrum Master o Dono do Produto não estão incluídos no tamanho do Time de desenvolvimento.

2. Artefatos

2.1 – Backlog do Produto: É um documento que contém todos os itens que devem ser desenvolvidos durante o projeto. Sendo o Dono do produto responsável por criar e manter este documento. É importante ressaltar que os itens do Backlog do Produto devem ser priorizados em função do Retorno do Investimento (ROI), onde os itens que apresentam maior valor para o negócio devem ser desenvolvidos primeiro.

2.2 – Backlog da Sprint: É um conjunto de itens selecionados para serem implementados durante a Sprint. Se durante o desenvolvimento da Sprint, for identifico a necessidade de novas tarefas ou a remoção de alguma tarefa,

somente os membros do Time de Desenvolvimento podem realizar essa alteração no Backlog da Sprint.

2.3 – Incremento do Produto: Ao término de cada Sprint, o Time de Desenvolvimento entrega um incremento do produto, o qual é o resultado do que foi produzido durante a Sprint. Ao final de uma Sprint, uma funcionalidade só é considerada pronta depois de passar por todas as etapas determinadas pelo Time de Desenvolvimento. Caso a funcionalidade não tenha sido concluída, a mesma deve retornar ao Backlog do produto para que seja incluída em uma próxima sprint.

3. Cerimônias

3.1 – Sprint: No Scrum, o trabalho é desenvolvido através de interações, denominadas Sprints. Cada Sprint tem duração de 2 a 4 semanas.

3.2 - Reunião de Planejamento da Sprint: Ocorre sempre no início de cada Sprint, com o objetivo de planejar o que será feito na Sprint.

3.3 - Reunião Diária: É uma reunião simples e importante, na qual cada membro do Time de Desenvolvimento deve responder sobre o que já fez, sobre o que pretende fazer e se há algum impedimento para a conclusão da tarefa sob sua responsabilidade. Geralmente essa reunião dura no máximo 15 minutos.

3.4 - Revisão da Sprint: Tem como objetivos apresentar o que a equipe fez durante a Sprint e entregar o produto ao Dono do Produto (Product Owner). Cabe ao Dono do Produto aceitar ou rejeitar a Sprint com base no que foi apresentado.

3.5 - Retrospectiva da Sprint: É uma reunião que tem como foco o aprimoramento do processo, analisando o que houve de bom e o que pode ser melhorado em uma Sprint. Todos os membros da equipe Scrum participa desta reunião.

Processo unificado

Rup

Também conhecido como Rational Unified Process, é um processo que faz parte da engenharia de software.

Ele é baseado em disciplinas em que cada uma distribui tarefas e responsabilidades para os envolvidos no desenvolvimento do software.

Essas disciplinas são semelhantes às que estudamos anteriormente:
• Modelagem de negócios; • Implementação; • Requisitos; • Teste; • Análise e Design; • Implantação.

Ainda no RUP, existem 3 disciplinas que servem de suporte e apoio ao ambiente:

Configuração e mudanças
Acompanham mudanças, configurações e status/medições onde são armazenados e que servirão de base para o andamento do projeto.

Projeto
Abrange questões como gestão de pessoas, orçamento, contratos.

Ambiente
Atividades que dão suporte à equipe de desenvolvimento, como os itens de IT, servidores, ferramentas.

Essas disciplinas têm suas responsabilidades e funções variadas, dependendo da fase que se encontra o projeto.

No processo RUP, o tempo está dividido em 4 fases:

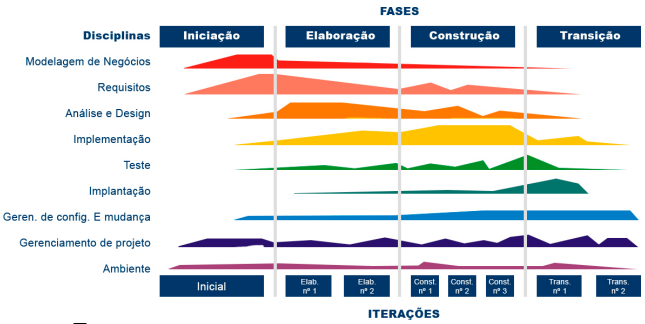


Imagem do software RUP, com as duas dimensões e as fases incluídas.

Referências

GUSTAFSON, Davis A. **Engenharia de software**. 8. ed. São Paulo: Pearson Education, 2007. cap. 8 e 13.

PAULA FILHO, Wilson de. **Engenharia de software**: fundamentos, métodos e padrões. 3. ed. São Paulo: LTC, 2009. cap. 1, 5 e 21.

SOMMERVIILLE, Ian. **Engenharia de software**. 1. ed. Porto Alegre: Artmed, 2003. cap. 10.

Explore Mais

XP: **http://www.extremeprogramming.org <http://www.extremeprogramming.org>**

Scrum: **https://www.scrum.org/resources/what-is-scrum <https://www.scrum.org/resources/what-is-scrum>**