

CI 221

DISCIPLINA: Engenharia de Software

AULA NÚMERO: 4

DATA: ____/____/____

PROFESSOR: Andrey

APRESENTAÇÃO

Nesta aula serão apresentados e discutidos os conceitos de Desenvolvimento Ágil de Sistemas

DESENVOLVIMENTO

Manifesto Ágil

Em 2001, vários desenvolvedores famosos, entre eles Kent Beck criaram a Aliança Ágil e o Manifesto para o Desenvolvimento Ágil de Software. Neste manifesto eles priorizavam:

- Indivíduos e Interações no lugar de processos e ferramentas
- Softwares funcionando no lugar de documentação abrangente
- Colaboração do Cliente no lugar de Negociação de contratos
- Resposta a modificações no lugar de seguir um plano.

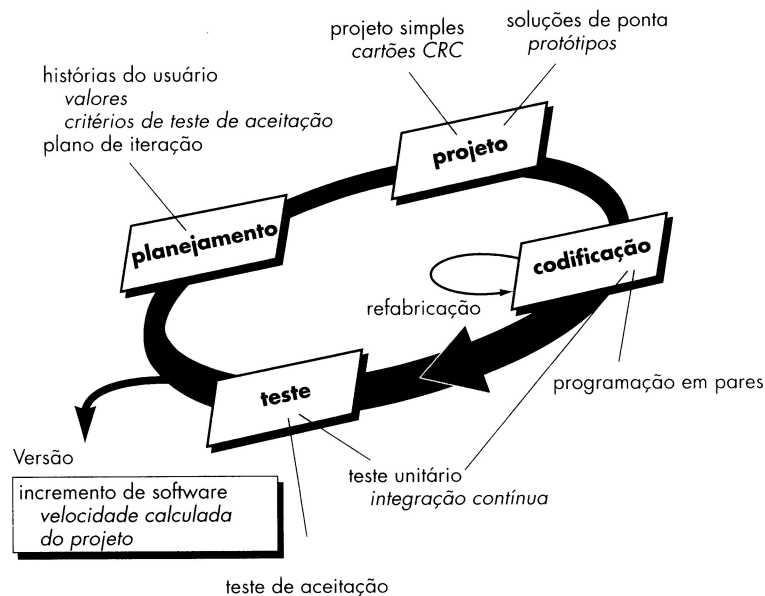
A Aliança Ágil definiu 12 princípios para alcançar agilidade:

1. Nossa prioridade é satisfazer o cliente desde o início por meio de entrega contínua de software valioso.
2. Modificações de Requisitos são bem-vindas, mesmo que tardias no desenvolvimento. Os processos ágeis aproveitam as modificações como vantagens para a competitividade do cliente.
3. Entrega de software funcionando frequentemente, a cada duas semanas até dois meses, de preferência no menor espaço de tempo.
4. O pessoal de negócios e os desenvolvedores devem trabalhar juntos diariamente durante todo o projeto.
5. Construção de projetos em torno de indivíduos motivados
6. O método mais eficiente e efetivo de comunicação entre os membros da equipe é conversa cara a cara.
7. A principal medida de progresso é o software funcionando
8. Processos ágeis promovem desenvolvimento sustentável
9. Atenção contínua à excelência técnica e ao bom projeto facilitam a agilidade
10. Simplicidade
11. As melhores soluções surgem de equipes autoorganizadas
12. A equipe auto-aperfeiçoa seu comportamento.

Equipe Ágil

- Competência
- Foco comum
- Colaboração
- Capacidade de tomada de decisão
- Habilidade de resolver problemas vagos
- Respeito e confiança mútua
- Autoorganização
- Extreme Programming (XP)

Inclui um conjunto de regras e práticas que ocorrem durante as 4 atividades de arcabouço: Planejamento, Projeto, Codificação e Testes.



Ciclo de vida do XP (Pressman, 2006)

Práticas da Extreme Programming (Wikipedia)

Jogo de Planejamento (Planning Game): O desenvolvimento é feito em iterações semanais. No início da semana, desenvolvedores e cliente reúnem-se para priorizar as funcionalidades. Essa reunião recebe o nome de Jogo do Planejamento. Nela, o cliente identifica prioridades e os desenvolvedores as estimam.

Pequenas Versões (Small Releases): A liberação de pequenas versões funcionais do projeto auxilia muito no processo de aceitação por parte do cliente, que já pode testar uma parte do sistema que está comprando.

Metáfora (Metaphor): Procura facilitar a comunicação com o cliente, entendendo a realidade dele.

Projeto Simples (Simple Design): Simplicidade é um princípio da XP. Projeto simples significa dizer que caso o cliente tenha pedido que na primeira versão apenas o usuário "teste" possa entrar no sistema com a senha "123" e assim ter acesso a todo o sistema, você vai fazer o código exato para que esta funcionalidade seja implementada, sem se preocupar com sistemas de autenticação e restrições de acesso.

Time Coeso (Whole Team): A equipe de desenvolvimento é formada pelo cliente e pela equipe de desenvolvimento.

Testes de Aceitação (Customer Tests): São testes construídos pelo cliente e conjunto de analistas e testadores, para aceitar um determinado requisito do sistema.

Ritmo Sustentável (Sustainable Pace): Trabalhar com qualidade, buscando ter ritmo de trabalho saudável (40 horas/semana, 8 horas/dia), sem horas extras.

Reuniões em pé (Standup Meeting): Reuniões em pé para não se perder o foco nos assuntos, produzindo reuniões rápidas, apenas abordando tarefas realizadas e tarefas a realizar pela equipe.

Posse Coletiva (Collective Ownership): O código fonte não tem dono e ninguém precisa solicitar permissão para poder modificar o mesmo. O objetivo com isto é fazer a equipe conhecer todas as partes do sistema.

Programação em Pares (Pair Programming): é a programação em par/dupla num único computador. Geralmente a dupla é formada por um iniciante na linguagem e outra pessoa funcionando como um instrutor.

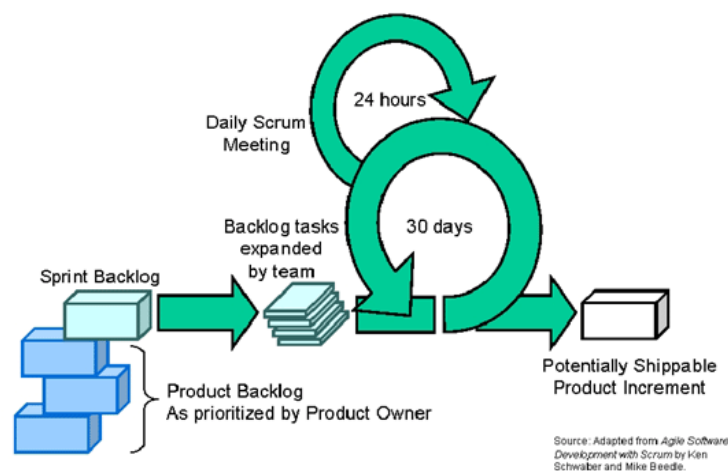
Padrões de Codificação (Coding Standards): A equipe de desenvolvimento precisa estabelecer regras para programar e todos devem seguir estas regras. Desta forma parecerá que todo o código fonte foi editado pela mesma pessoa, mesmo quando a equipe possui 10 ou 100 membros.

Desenvolvimento Orientado a Testes (Test Driven Development): Primeiro crie os testes unitários (unit tests) e depois crie o código para que os testes funcionem. Esta abordagem é complexa no início, pois vai contra o processo de desenvolvimento de muitos anos. Só que os testes unitários são essenciais para que a qualidade do projeto seja mantida.

Refatoração (Refactoring): É um processo que permite a melhoria continua da programação, com o mínimo de introdução de erros e mantendo a compatibilidade com o código já existente.

Integração Contínua (Continuous Integration): Sempre que produzir uma nova funcionalidade, nunca esperar uma semana para integrar à versão atual do sistema.

SCRUM



Ciclo de vida do Scrum (tirado de <<http://www.methodsandtools.com/archive/scrum1.gif>>)

- Scrum é um processo ágil com as seguintes características:
 - permite manter o foco na entrega do maior valor de negócio, no menor tempo possível.

UNIVERSIDADE FEDERAL DO PARANÁ - UFPR
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

- permite a rápida e contínua inspeção do software em produção (em intervalos de duas a quatro semanas).
- as necessidades do negócio é que determinam as prioridades do desenvolvimento de um sistema.

Principais Atefatos:

- Sprint BackLog
- Product Backlog
- Daily Scrum Meeting
- Incremento de produto potencialmente entregável

Principais papeis:

- Dono do produto
- Scrum master
- Equipe

ATIVIDADES

1. O que é solução de ponta em XP?
2. Descreva os conceitos de Refabricação e Programação em Pares.
3. Descreva as vantagens da aplicação das técnicas Refabricação, Código sem dono, Ritmo sustentável e Programação em Pares.

BIBLIOGRAFIA

PRESSMAN, R. S. Engenharia de Software, 6a. ed.. McGraw-Hill. São paulo, 2006.

ASTELS, D.; MILLER, G.; NOVAK, M. Extremme Programming: Guia Prático. ed. Campus, Rio de Janeiro, 2002

WIKIPEDIA. Programação Extrema. <http://pt.wikipedia.org/wiki/Programação_extrema> Acessado em 26/08/2009.

COHN, M. Uma introdução ao Scrum. [Online] Disponível na interne via WWW. URL: <<http://www.mountangoatsoftware.com/system/asset/file/51/PortugueseScrum.odp>>. Acessado em 15/03/2010.