

Processos Ágeis de Desenvolvimento de Software

- Foco no XP -

Rodrigo Rebouças de Almeida

rodrigor@rodrigor.com



Processo

- Conjunto de atividades ordenadas, restrições e recursos que produzem um resultado de algum tipo. (Pfleeger)
- Em software: Processo de desenvolvimento
 - Define **quem** faz o quê, quando e como.

O processo guia o desenvolvimento do software.

Porque precisamos de um processo?

- Desenvolver software envolve **pessoas**.
 - Precisamos definir suas atividades, artefatos, prazos, prioridades.
- Precisamos **planejar**, estimar, **executar**.
- E **controlar** a evolução do desenvolvimento.

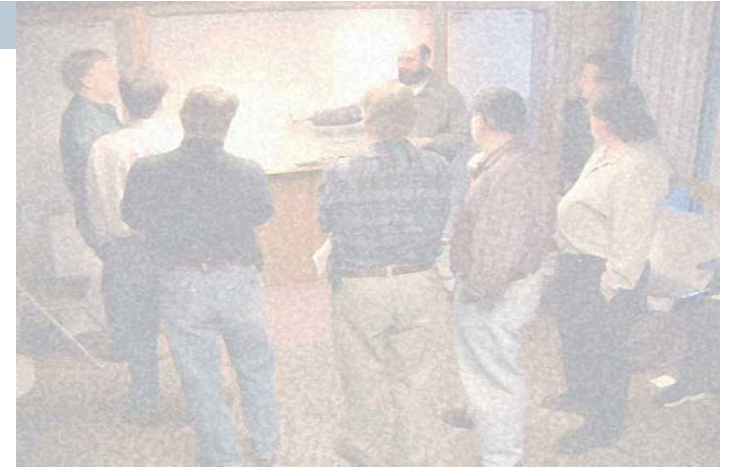
O processo estrutura o *planejamento*, *execução* e *controle* das atividades e artefatos gerados.

Processos tradicionais



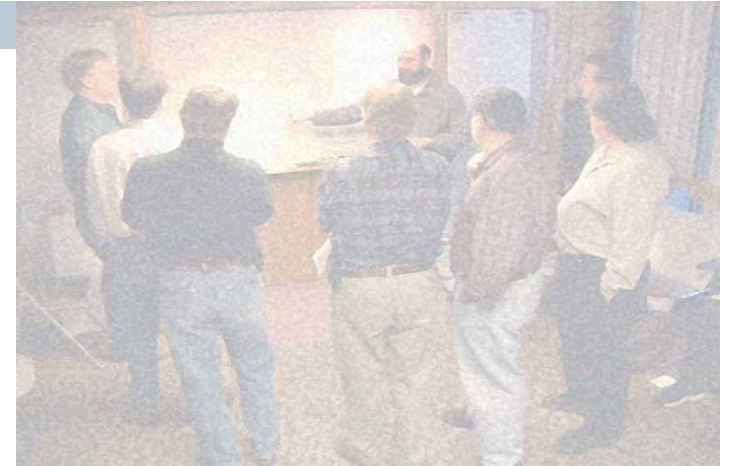
- Processos “pesados” ou focados em planejamento
Ex.: Processo Unificado.
- Contexto:
 - Requisitos não mudam (ou mudam pouco) com o tempo.
 - Grande esforço no planejamento e projeto do software.
 - Gerência focada nos papéis desempenhados pelas pessoas: programador, arquiteto, analista, testador,...

Manifesto Ágil



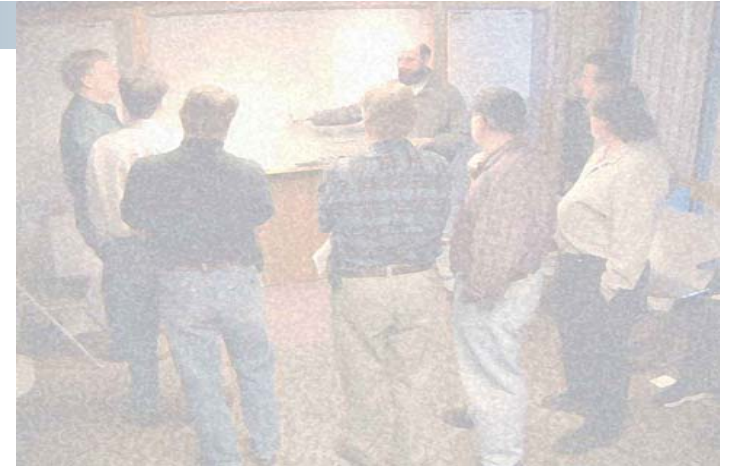
- Resposta à insatisfação com os processos tradicionais de desenvolvimento de software.
- Estabelece os valores:
 - **Indivíduos e Interações** X Processos e Ferramentas.
 - **Software Funcionando** X Documentação.
 - **Colaboração com o cliente** X Contrato.
 - **Resposta a mudanças** X Seguir planejamento.

Processos Ágeis



- Muito mais do que simples definições de processos.
- Afetam várias dimensões das empresas e dos projetos.
 - Relação profissional
 - Atribuição de responsabilidades e confiança.
 - Relação com clientes
 - Parceria ao invés da relação de prestação de serviço.
 - Formato contratual
 - Contratos têm que ser adaptáveis.

Processos Ágeis



- Existem vários processos que seguem os valores definidos no manifesto:
 - **eXtreme Programming** (principal representante)
 - Foco em testes.
 - Scrum
 - Foco na adaptação contínua do processo à equipe.
 - FDD (Feature Driven Development)
 - Foco nas funcionalidades.
 - (...)

eXtreme Programming

- Práticas de Eng. Software ao extremo!
 - Testes de unidade, de aceitação, de integração...
 - Testar antes de codificar!
 - Dois programadores e um teclado (Pair Programming).
 - Releases pequenos (com software funcionando para o cliente).
 - Integração contínua.
 - "Refatoramento" contínuo.
 - Cliente participa da equipe de desenvolvimento.

Valores de XP

- Comunicação
 - Interação constante entre os membros da equipe.
- Simplicidade
 - KISS!
- Feedback
 - Iterações pequenas e interferência constante do cliente.
- Coragem
 - Mudanças constantes, testes e refatoramento constante.

Algumas Práticas de XP

- *Planning Game.*
- Releases pequenas.
- Integração contínua.
- Posse coletiva do código.
- *Design* simples.
- Programação em pares.
- Programação orientada a testes.
- “Refatoramento” constante.

eXtreme Programming na prática

- Etapas do processo:
 - Levantamento de estórias do usuário
 - Funcionalidades
 - Prioridade
 - “Planning Game”
 - Planejamento da Release
 - Planejamento de Iterações
 - Desenvolvimento
 - Programação em pares
 - Programação orientada a testes
 - Envolvimento com o usuário

Pré-requisitos para uso de processos ágeis

- Cliente sempre presente.
- Requisitos mudam com muita frequência
 - Requisitos imprevisíveis.
- Equipe forte, bons profissionais
 - Profissionais devem ser capazes de tomar decisões.
- Não adotar contrato com valor fixo
 - Contrato fixo -> requisitos fixos -> não usar processos ágeis.

Quando não usar processos ágeis

- Cliente ausente
 - Reuniões a cada dois meses.
- Requisitos fixos, com pouca mudança no tempo.
- Equipe volátil e despreparada.
- Contratos com valor e escopo fixos.
- Submissão a programas de certificação de qualidade, como CMMI.
 - É necessário incorporar muitas atividades nos processos ágeis para atenderem aos requisitos dos programas de qualidade.

Processos ágeis são uma boa idéia

- Quando aplicados no contexto adequado.
- Não são tão “leves” assim...
 - *Standup meeting*, testes constantes, iterações curtas, releases pequenas, ...
- Exigem disciplina e competência da equipe.

Referências

- Pfleeger M., ***Software Engineering Theory and Practice***, 2nd ed., 2001, ISBN: 0-13-0290490-1.
- **Manifesto for Agile Software Development**, on-line: <http://agilemanifesto.org/>
- **Extreme Programming: A Gentle Introduction**, on-line: <http://extremeprogramming.org/>
- **Control Chaos**, on-line: <http://www.controlchaos.com/>
- Fowler M., ***The new Methodology***, on-line: <http://www.martinfowler.com/articles/newMethodology.html>