

Engenharia da Computação

Tópicos Avançados em

Engenharia de Software

Aula 2 (01/03)

mario.godoy@univasf.edu.br

<http://www.univasf.edu.br/~mario.godoy/>

Universidade Federal do Vale do São Francisco - UNIVASF
Colegiado de Engenharia de Computação - CECOMP



Desenvolvimento de Software

Desenvolvimento de Software

O que é metodologia de desenvolvimento?

- Conjunto de **atividades** que **auxiliam** na **produção** de software.

- Existem **atividades** fundamentais **comuns** a **todas** as **metodologias** [Sommerville (2003)]:
 - **Especificação de Software:**
 - **Funcionalidades** (requisitos).
 - **Restrições** do software.

A diagram consisting of a bracket on the left that groups the two sub-items 'Funcionalidades' and 'Restrições'. A line extends from the right side of this bracket to the word 'Clientes'.

 - **Projeto e Implementação de Software:**
 - **Elaboração de Modelos** (diagramas).
 - **Implementação** (linguagem de programação).

 - **Validação de Software:**
 - **Funcionalidades** devem **obedecer** os **requisitos**.

 - **Evolução de Software:**
 - **Adaptação** ao **longo** do **tempo**.

Metodologias Tradicionais

Contexto das Metodologias Tradicionais

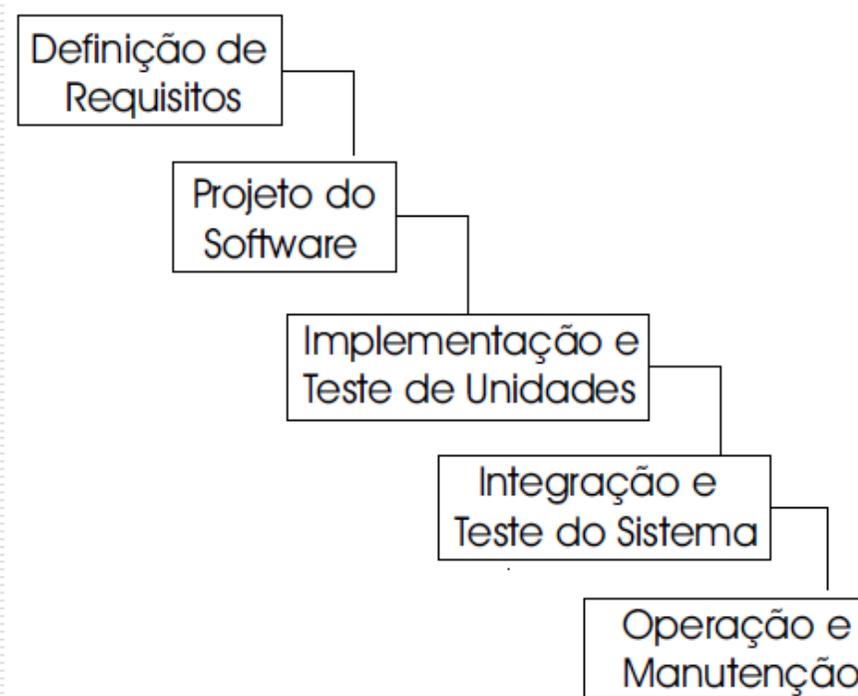
Contexto das Metodologias Tradicionais

- ❑ **Mainframe e terminais burros**
- ❑ **Ferramenta de desenvolvimento carentes:**
 - Depuradores.
 - Analisadores de código.

Metodologias Tradicionais

Exemplos de Metodologias Tradicionais

- ❑ Clássico ou Tradicional (cascata).
- ❑ Ao término de cada etapa, é entregue sua documentação.



Estima-se que caso alguma alteração tenha como **custo "1x" quando feita na fase de requisitos**, ela terá um **custo de "60x a 100x" quando feita na fase de implantação [Pressman (2001)]**.

Metodologias Tradicionais

- ❑ **Dificuldades enfrentadas** [Standish Group, (1995)] baseado em **8380** projetos:
 - **16,2%** respeitaram os **prazos** e os **custos** e possuíam **todas as funcionalidades** especificadas.
 - **31%** dos projetos foram **cancelados** antes de estarem completos.
 - **52,7%** foram entregues, **com prazos maiores, custos maiores ou com menos funcionalidades** do que especificado.

Metodologias Tradicionais

Dificuldades enfrentadas [Standish Group, (1995)] baseado em **8380** projetos:

- **Dentre os projetos que não foram finalizados:**
 - **A média de atrasos foi de 222%.**
 - **A média de custo foi de 189% a mais do que o previsto.**
 - **Apenas 61% das funcionalidades originais foram incluídas.**

Desenvolvimento de Software - Forma tradicional



Como o cliente explicou

--	--	--	--	--

--	--	--	--	--

Introdução a Ágil

A maioria das metodologias ágeis nada possuem de novo. O que as diferencia das metodologias tradicionais são o enfoque e os valores [Cockburn et al., (2001)].

Por que Ágil?

Manifesto para Desenvolvimento Ágil de Software

Estamos descobrindo maneiras melhores de desenvolver software, fazendo-o nós mesmos e ajudando outros a fazerem o mesmo. Através deste trabalho, passamos a valorizar:

- Indivíduos e interações** mais que processos e ferramentas.
- Software em funcionamento** mais que documentação abrangente.
- Colaboração com o cliente** mais que negociação de contratos.
- Responder a mudanças** mais que seguir um plano.

<http://agilemanifesto.org/>

Introdução

- ❑ Em 2001, 17 especialistas em metodologias definiram o manifesto ágil.
- ❑ O “Manifesto Ágil” **não rejeita** os **processos** e **ferramentas**, a **documentação**, a **negociação** de **contratos** ou o **planejamento**, mas simplesmente mostra que eles **têm importância secundária**.
- ❑ Metodologias ágeis é que elas são **adaptativas** e não **preditivas**.
- ❑ Métodos **ágeis obtiveram melhores resultados** em termos de cumprimento de **prazos**, de **custos** e padrões de **qualidade** [Charette, R., (2001)].

Princípios do Desenvolvimento Ágil

- 1. A prioridade é satisfazer o cliente** através da **entrega contínua e adiantada** de **software** com **valor agregado**.
- 2. Mudanças nos requisitos são bem-vindas**, mesmo tardiamente no desenvolvimento. **Processos ágeis** tiram **vantagem** das **mudanças** visando vantagem competitiva para o cliente.
- 3. Entregar** frequentemente **software funcionando**, de poucas semanas a poucos meses, com preferência à menor escala de tempo.
- 4. Gestores e desenvolvedores** devem **trabalhar diariamente** em **conjunto** por **todo o projeto**.
- 5. Construa projetos** em torno de **indivíduos motivados**. Dê a eles o **ambiente** e o **suporte necessário** e **confie neles** para fazer o trabalho.

<http://agilemanifesto.org/>

Princípios do Desenvolvimento Ágil

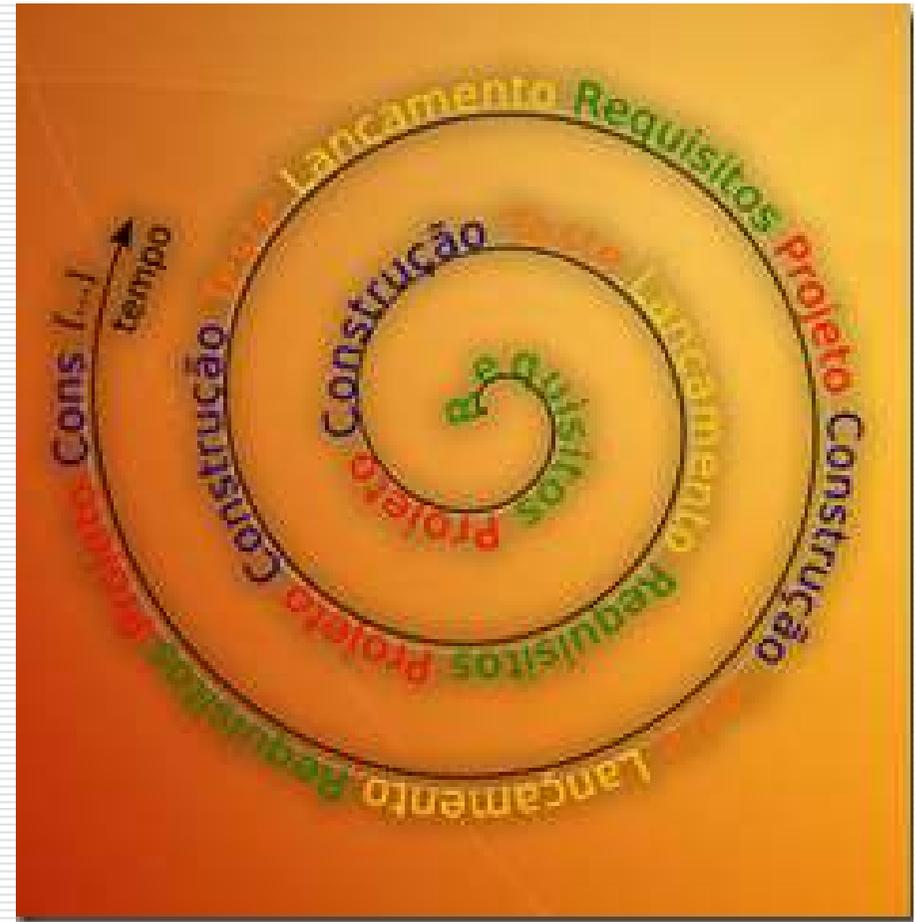
6. **O método** mais **eficiente e eficaz** de **transmitir informações** para e entre uma **equipe de desenvolvimento** é através de **conversa face a face**.
7. **Software funcionando** é a medida primária de progresso.
8. **Desenvolvimento sustentável**. Os **patrocinadores, desenvolvedores e usuários** devem ser capazes de **manter um ritmo constante indefinidamente**.
9. **Contínua atenção à excelência técnica** e bom **design** aumenta a agilidade.
10. **Simplicidade**: a arte de maximizar a quantidade de trabalho não realizado é essencial.
11. As **melhores arquiteturas, requisitos e designs** emergem de **equipes auto-organizáveis**.
12. Em **intervalos regulares**, a equipe reflete sobre **como se tornar mais eficaz** e então **refina e ajusta** seu comportamento **de acordo**.

<http://agilemanifesto.org/>

Introdução

Metodologias ágeis são baseadas no **processo iterativo**.

- Mecanismos de comunicação:
 - **Testes.**
 - **Feedback de protótipos.**
 - **Entrevista** com clientes.



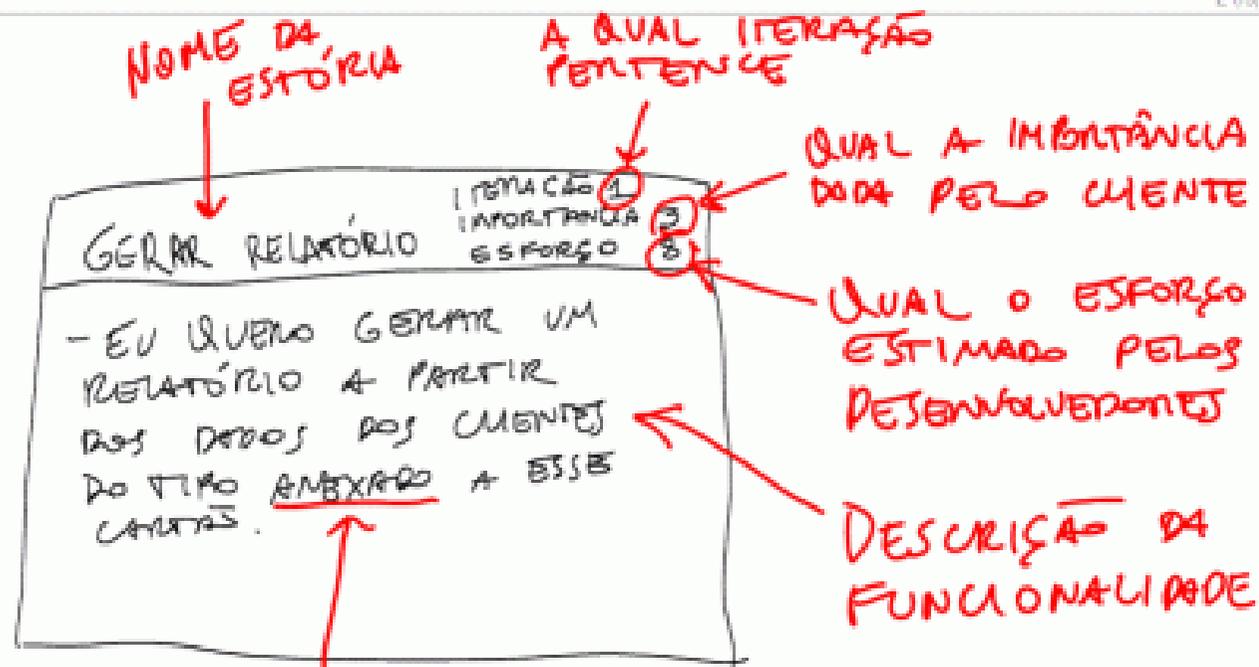
Modelos de Histórias do Usuário

- Pode ser ao estilo "usuário".

ESTÓRIAS DO USUÁRIO

Note Title

24/5/2006



ANEXAR tudo o que for útil
AO DESENVOLVIMENTO DA ESTÓRIA

Ferramenta

□ XPlanner

- Ferramenta de **código livre** que suporta a XP.
- Auxiliando a **fase de planejamento** [XPlanner, (2004)].

<http://www.xplanner.org/>

□ *TargetProcess*

- Ferramenta propõe gerenciamento de projetos ágeis.

www.targetprocess.com

Referências

- ❑ [Sommerville (2003)] Sommerville, I. Engenharia de Software. Editora Addison-Wesley. 592p, 2003.
- ❑ [XPlanner, (2004)] XPlanner, disponível em <http://www.xplanner.org/>, acessado em 10 de fevereiro de 2011.
- ❑ [Pressman (2001)] Pressman, R. Engenharia de Software. McGraw-Hill, 2001.
- ❑ [Standish Group, (1995)] CHAOS report, 586 Olde Kings Highway. Dennis, MA 02638, USA, 1995.
- ❑ [Cockburn et al., (2001)] Cockburn, A. e Highsmith, J. "Agile Software Development: The Business of Innovation", IEEE Computer, Sept., pp. 120-122, 2001.
- ❑ [Charette, R., (2001)] Charette, R. "Fair Fight? Agile Versus Heavy Methodologies", Cutter Consortium E-project Management Advisory Service, 2, 13, 2001.