

# Metodologias e Processos Ágeis para o Desenvolvimento de Softwares

Disciplina: **Engenharia de Software I**  
Prof. Me. Fernando Roberto Proença

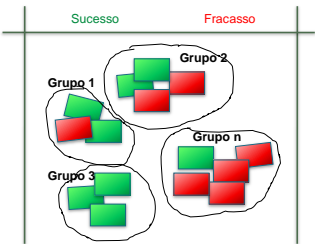
## Tópicos da aula

- Métodos/Modelos Tradicionais de Desenvolvimento de software
- Problemas no Desenvolvimento de software
- Métodos Ágeis
  - O Manifesto Ágil
  - Definição de Métodos Ágeis
  - O que muda com os Métodos Ágeis?
  - Principais Métodos Ágeis

2

## Dinâmica

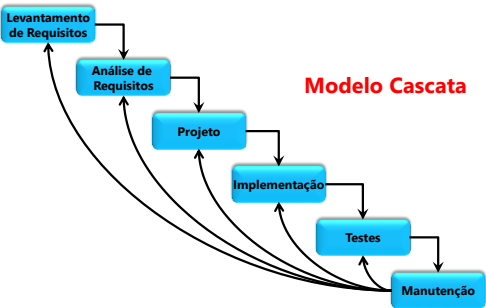
- Por que seu projeto termina com sucesso?
- Por que seu projeto fracassa?
- Discussão em grupo
  - 15 minutos
  - Pode agrupar os problemas e os fracassos em categorias.



Ref: Henrik Kniberg

3

## Engenharia de Software Tradicional (Métodos Tradicionais/Clássicos de Desenvolvimento)



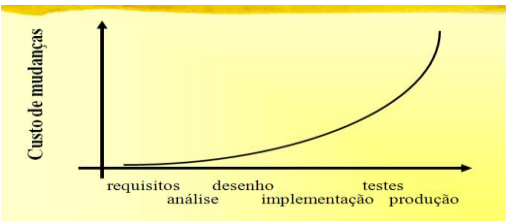
4

## Premissas básicas do métodos / modelos tradicionais

- Coloca grande ênfase em **projetar antes de construir**.
  - É necessário:
    - Fazer uma análise profunda e detalhada dos requisitos antes de projetar a arquitetura do sistema;
    - Fazer um estudo minucioso e elaborar uma descrição detalhada da arquitetura antes de começar a implementá-la
    - Testar o sistema completamente antes de mandar a versão final para o cliente.

5

## Métodos Tradicionais/Clássicos de Desenvolvimento



6

### Métodos Tradicionais/Clássicos de Desenvolvimento

- Algumas empresas adotam metodologias de desenvolvimento e práticas extremamente formais e controladoras, porém ainda não conseguem obter qualidade.
- Por quê?
  - ❑ Pouca preocupação com as pessoas e a interação entre elas
  - ❑ Pouca comunicação com o cliente
  - ❑ Custos muito altos
  - ❑ Excesso de formalismo.
- Qual a consequência disso?
  - ❑ Projeto cancelado
  - ❑ Prazos estourados
  - ❑ No fim o software não serve mais.

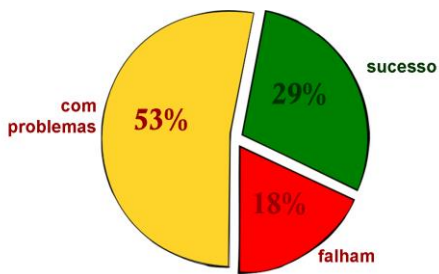
7

### Novos ventos no desenvolvimento de software...

- Sociedade demanda
  - ❑ grande quantidade de sistemas/aplicações
  - ❑ software complexo, sistemas distribuídos, heterogêneos
  - ❑ requisitos mutantes (todo ano, todo mês, todo dia)
- Mas, infelizmente...
  - ❑ não há gente suficiente para desenvolver tanto software com qualidade.

8

### Resultado dos projetos em 2004



Fonte: Relatório do Chaos (Chaos Report)

9

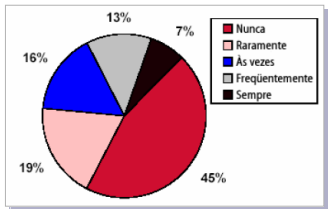
### Resultado dos projetos em 2004

1994	2004
Projetos não concluídos -----> 31%	Projetos não concluídos -----> 18%
Projetos bem sucedidos -----> 16%	Projetos bem sucedidos -----> 29%
Estouro médio de custo -----> 180%	Estouro médio de custo -----> 56%
Estouro médio de prazo -----> 164%	Estouro médio de prazo -----> 84%

Fonte: Relatório do Chaos (Chaos Report)

10

### Utilização das funcionalidades do Software...

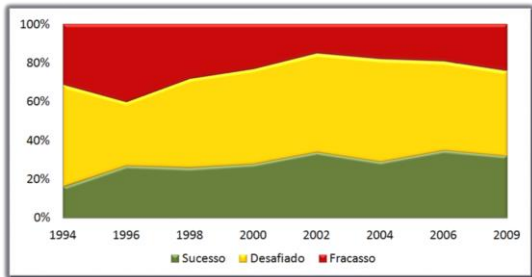


Mais de 64% de um sistema de software quase nunca é utilizado!

Fonte: Relatório do Chaos (Chaos Report)

11

### Resultado dos projetos nos últimos anos



Fonte: Relatório do Chaos (Chaos Report)

12

## No Desenvolvimento de Software...

- Temos que alterar o Software
  - No início do projeto, normalmente não se sabe precisamente o que se quer
  - Software evolui para atender ao negócio
  - Software nunca fica "pronto"
- Portanto...
  - Precisamos parar de tentar evitar mudanças
    - As mudanças são importantes na vida de um software
  - Precisamos de uma metodologia de desenvolvimento que nos permita alterar **constantemente** o código sem comprometer sua qualidade.

13

## Problemas

- "Todos os dias o meu chefe muda de ideia sobre o que nós estamos produzindo."

-----> Falta de planejamento

- "Eu gasto metade dos meus dias em reuniões sobre como aumentar a produtividade." (ao invés de produzir).

-----> Excesso de Planejamento

14

## Problemas

- "Meu chefe leu em uma revista que desenvolvedores que usam a linguagem "x" são 2x mais produtivos. Agora ele comprou uma licença para nós e cortou os prazos pela metade."

-----> Crença em ferramentas e processos

- "Estamos 4 meses em um projeto de 5 meses e recebi os requisitos finais ontem."

-----> Mau Planejamento

15

## Problemas

- Com métodos tradicionais de desenvolvimento
  - Supõem que é possível prever o futuro
  - Pouca interação com os clientes
  - Ênfase em burocracias
    - Documentos, formulários, processos, controles rígidos, etc.
  - Avaliação do progresso baseado na evolução da burocracia e não do código.
- Com o software
  - Grande quantidade de erros
  - Falta de flexibilidade.

16

## Como resolver o impasse?

- Melhores Tecnologias
  - Padrões de Projeto (reutilização de ideias)
  - Componentes (reutilização de código)
  - Middleware/frameworks (aumenta a abstração)
    - Hibernate (Java)
    - Entity (.NET)
- Melhores Metodologias
  - Métodos Ágeis.

17

## Métodos Ágeis

- O que é agilidade?
  - **Agilidade não é somente velocidade...**
    - é velocidade e versatilidade
    - Habilidade tanto para criar quanto para responder à mudança.
- Aspectos importantes dos Métodos Ágeis:
  - foca as pessoas e suas interações;
  - capacitação dos indivíduos para tomar decisões rapidamente;
  - adaptação de seus próprios processos.

18

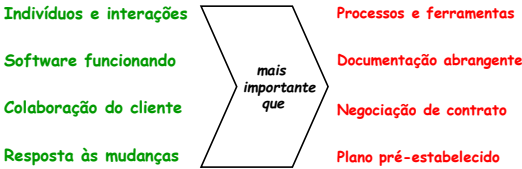
## O Manifesto Ágil – Histórico

- Movimento iniciado por programadores experientes e consultores em desenvolvimento de software.
  - ❑ Questionam e se opõem a uma série de mitos e práticas adotadas em abordagens tradicionais de Engenharia de Software e Gerência de Projetos.
- Manifesto Ágil:
  - ❑ Assinado por 17 desenvolvedores em Utah-EUA em fevereiro de 2001.
  - ❑ <http://agilemanifesto.org>

19

## O que diz o Manifesto Ágil?

*"Estamos descobrindo melhores maneiras de desenvolver software, fazendo software e ajudando outros a fazê-lo. Através deste trabalho passamos a valorizar:*



*Isto é, embora haja valor nos itens do lado direito, nós valorizamos **mais** os do lado esquerdo."*

<http://www.agilemanifesto.org> – 2001

## O Manifesto Ágil

- Indivíduos e interações valem mais que processos e ferramentas
  - ❑ Quem desenvolve o software?
  - ❑ Melhores ferramentas não são segurança de trabalho bem feito.
    - "Um idiota com uma ferramenta ainda é um idiota."

21

## O Manifesto Ágil

- Um software funcionando vale mais do que documentação extensa
  - ❑ O que um cliente gostaria de receber?
    - O software ou um documento explicando o software?
  - ❑ Geralmente clientes não entendem documentos técnicos!

22

## O Manifesto Ágil

- A colaboração do cliente vale mais do que a negociação do contrato
  - ❑ Apenas os clientes podem dizer o que querem (não somos adivinhos!)
  - ❑ Eles nunca acertam da 1ª vez!
  - ❑ Contrato é importante
    - Direitos e deveres
  - ❑ Contrato não substitui a comunicação.

23

## O Manifesto Ágil

- Responder a mudanças vale mais do que seguir um plano
  - ❑ As pessoas mudam de prioridade!
  - ❑ O ambiente, as necessidades também mudam.
  - ❑ Plano de Projeto é essencial, mas deve ser maleável.

24

### O Manifesto Ágil

- Objetivo: satisfazer o cliente entregando, rapidamente e com frequência, sistemas com algum valor.
  - Entregar versões funcionais em prazos curtos;
  - Estar preparado para requisitos mutantes;
  - Pessoal de negócios e desenvolvedores juntos;
  - Troca de informações através de conversas diretas.

25

### O Manifesto Ágil

- 12 princípios por traz do Manifesto Ágil:
  1. Satisfazer o cliente
  2. As mudanças são bem vindas
  3. Entrega periódica do software funcionando
  4. Desenvolvedores e o pessoal de negócio devem trabalhar juntos
  5. Indivíduos Motivados
  6. Conversas face a face

26

### O Manifesto Ágil

- 12 princípios por traz do Manifesto Ágil:
  7. Software funcionando é a principal medida de progresso
  8. Manter um ritmo constante sempre
  9. Atenção contínua, excelência técnica e bom projeto
  10. Simplicidade
  11. Equipes auto-organizáveis ou auto-gerenciáveis – confiança no trabalho da equipe
  12. Periodicamente a equipe deve refletir sobre como se tornar mais efetivo.

27

### Em um projeto ágil ideal...

- O gerente de projeto concorda em prosseguir sem que todos os requisitos estejam bem definidos
- Os desenvolvedores concordam em prosseguir sem ter todos os requisitos estejam documentados
- Os membros da equipe sabem que alguém vai ajudar quando ocorrerem problemas

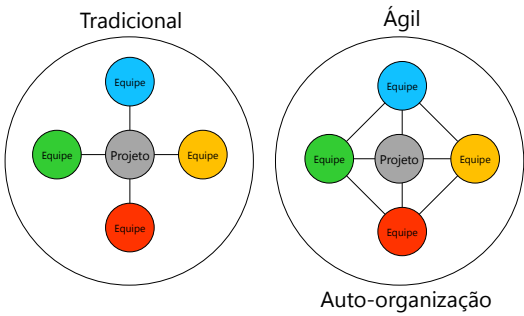
28

### Em um projeto ágil ideal...

- Os gerentes percebem que não precisam dizer à equipe o que fazer, ou garantir o que vai ser feito
- A equipe percebe que ninguém vai dizer o que fazer, isto faz parte do trabalho da equipe
- Não existem mais a impressão de divisão (*testers and programmers*), todos são desenvolvedores.

29

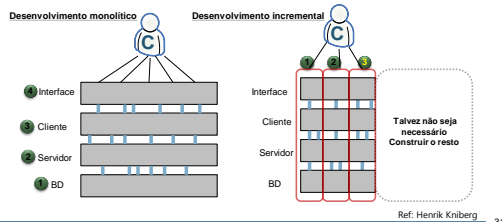
### Mudança de Postura



30

## Modelo Iterativo e Incremental

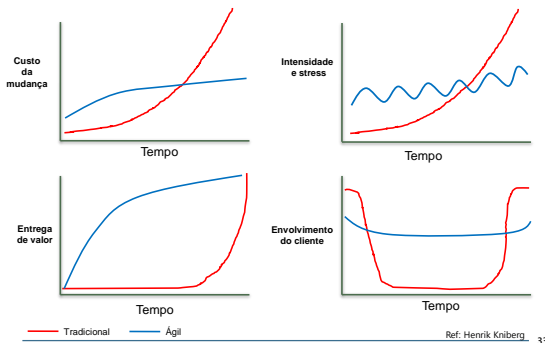
- **Iterativo** – não espere ter tudo correto na primeira vez
- **Incremental** – construa em “pedaços” verticais (funcionalidades) ao invés de horizontais (camadas)



As Metodologias ágeis são tentativas de refinar as metodologias iterativas, tirando o foco do processo em si e dando mais ênfase para a contribuição das pessoas...

32

## O que muda?



## O que muda?

- Métodos tradicionais
  - O planejamento deve propiciar a prevenção de mudanças
- Métodos ágeis
  - A mudança é incorporada ao escopo
- Razões de mudança
  - Necessidades de negócio
  - Novas oportunidades
  - Mudanças de legislação
  - Requisitos incompletos

34

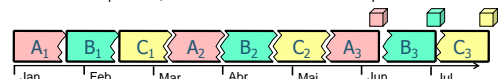
## Com os métodos ágeis temos...

- Orientação a Objetos
  - Facilita e cria oportunidades para mudanças
- Técnicas de Refatoração
  - Melhoraria do código sem alterar seu comportamento.
- Testes automatizados
  - Nos dão segurança quando fazemos mudanças
- Prática / cultura de mudanças
  - Aprendemos técnicas e adquirimos experiência em lidar com código mutante

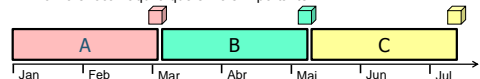
35

## Multitarefa

- Visão tradicional
  - “Tudo é importante, vamos fazer tudo ao mesmo tempo!”



- Visão ágil
  - “Prioriza e foca naquilo que é mais importante!”



Ref: Henrik Kniberg 36

Metodologias ágeis é uma febre? É uma onda passageira?

É o início de uma mudança na forma de trabalho...

37

## Principais Métodos Ágeis

- **Scrum**
- **Programação eXtrema (XP)**
- Adaptive Software Development (Desenvolvimento de Software Adaptável)
- Feature Driven Development (Desenvolvimento Orientado à Característica)
- Crystal (uma família de métodos)

38

## Dúvidas?

Prof. Me. Fernando Roberto Proença

[fernando.proenca@uemg.br](mailto:fernando.proenca@uemg.br)



39

Apresentação da descrição do  
2º Seminário...

40