INTRODUÇÃO A MÉTODOS ÁGEIS

Profa. Karin Becker
Engenharia de Software N
Instituto de Informática - UFRGS

Introdução

- Santo Graal em engenharia de software: como desenvolver software
  - Com funcionalidades esperadas
  - Dentro do prazo e custo
  - De qualidade
- Tendência até os anos 90:
  - Processos Prescritivos e maior Formalidade
    - CMM, CMM-I
    - Processo unificado
  - " Heavy-weight process "
- Tendência atual:
  - Processos Descritivos e Comportamentais
  - " Light-weight process "
  - **Métodos ágeis**

Por que ser ágil?

- Cerca de **30%** do projetos de desenvolvimento de software são bem sucedidos

Fonte: Standish Caos Report 2009

- Cerca de **55 %** do projetos de desenvolvimento de software são bem sucedidos*

Fonte: Agile Surveys 2010

*Obs: pela percepção ágil de sucesso, este número pode chegar a **80%**

A Crise de Software em Números

- Standish Chaos Report
  - Análise regular sobre milhares de projetos de desenvolvimento de software nos EUA
    - Desde 1994, a cada 2 anos
  - Dividem projetos pesquisados em três categorias
    - Sucesso: prazo, orçamento, funcionalidades
    - Desafiado: acima do orçamento, desvio de cronograma, menos funcionalidades
    - Falho: cancelado
  - Análise dos fatores de sucesso/insucesso

Para outras análises e questionamentos
www.galorath.com/wp/software-project-failure-costs-billions-better-estimation-planning-can-help.php

A Crise de Software em Números

Standish project benchmarks over the years

Year	Successful (%)	Challenged (%)	Failed (%)
1994	16	53	31
1996	27	33	40
1998	26	46	28
2000	28	49	23
2004	29	53	18
2006	35	46	19
2009	32	44	24

A Crise de Software em Números

Standish project benchmarks over the years

Year	Successful (%)	Challenged (%)	Failed (%)
1994	16	53	31
1996	27	33	40
1998	26	46	28
2000	28	49	23
2004	29	53	18
2006	35	46	19
2009	32	44	24

IID

IID: Incremental Iterative Development

A Crise de Software em Números

Standish project benchmarks over the years

Year	Successful (%)	Challenged (%)	Failed (%)
1994	16	53	31
1996	27	33	40
1998	26	46	28
2000	28	49	23
2004	29	53	18
2006	35	46	19
2009	32	44	24

IID

Ágil

A crise e métodos ágeis

Standish project benchmarks over the years

Year	Successful (%)	Challenged (%)	Failed (%)
1994	16		
1996	27		
1998	26		
2000	28		
2004	29		
2006	35		
2009	32		

CHAOS Success Factors

1. User Involvement
2. Executive Support
3. Clear Business Objectives
4. Emotional Maturity
5. Optimization
6. Agile Process
7. Project Management Expertise
8. Skilled Resources
9. Execution
10. Tools and infrastructure

IID

Ágil

Standish Caos Report 2009

Caos Report Summary 2009

- "somente 20% dos custos estão relacionados ao desenvolvimento de software: o restante é necessário para apoiar a burocracia do negócio"
- "neste sentido, fizemos mudanças em nossos fatores de sucesso. … A grande mudança é que **removemos processos formais** … processos formais são um fator importante, mas não mais que uma coisa boa …"
- "… estamos em um **círculo vicioso**: taxas de sucesso são baixas, adiciona-se controle; se elas permanecem baixas, mais controle … "
- "**Precisamos quebrar este círculo, parar de adicionar controles, avaliar nosso ambiente, adicionar valores aos nossos esforços, acordar e refinar nossa execução**"

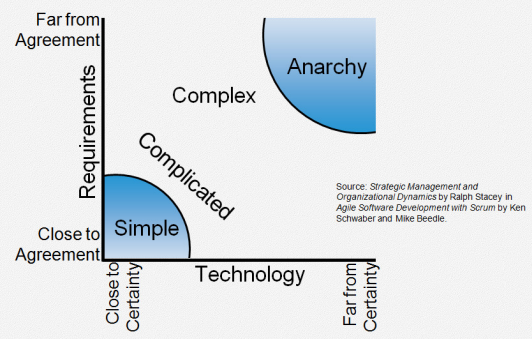
*Fonte: www.portal.state.pa.us/portal/server.pt/.../**chaos**_summary_**2009**_pdf*

Manifesto Ágil

- ***Indivíduos e interações*** são mais importantes que *processos e ferramentas*
- ***Software funcionando*** é mais importante do que *documentação completa e detalhada*
- ***Colaboração com o cliente*** é mais importante do que *negociação de contratos*
- **Adaptação a mudanças** é mais importante do que *seguir o plano inicial*

http://www.agilemanifesto.org/

Níveis de ruído



O que é um método ágil?

1. É uma atitude, não um processo prescritivo
2. É um suplemento aos métodos existentes, e não uma metodologia completa
3. É uma forma efetiva de se trabalhar em conjunto para atingir as necessidades das partes interessadas no projeto.
4. É uma coisa que funciona na prática, não é teoria acadêmica
5. É para o desenvolvedor médio, mas não é um substituto de pessoas competentes
6. Não é um ataque à documentação, pelo contrário aconselha a criação de documentos que têm valor
7. Não é um ataque às ferramentas CASE

Métodos ágeis enfatizam...

- Comunicação face a face
- Colaboração entre cliente e desenvedores
- Software operacional como a principal demonstração de progresso
- Demonstração frequente de progresso
- Técnicas de engenharia que agreguem valor (ex. Test Driven Development, Continuous Integration, Refactoring to Design Patterns),
- Retrospectivas e melhoria contínua

Princípios Ágeis (1/4)

- maior prioridade é satisfazer o cliente através da entrega rápida e contínua de software com valor agregado
- acolher mudanças nos requisitos, até mesmo em estágios avançados do processo. Mudanças são em benefício da competitividade do negócio do cliente
- entregar software operacional frequentemente (poucas semanas a a poucos meses - quanto menor o período, melhor)

Princípios Ágeis (2/4)

- Pessoas do negócio e desenvolvedores devem trabalhar juntas no projeto diariamente
- Contruir projetos com pessoas motivadas. Dar o ambiente e apoio que elas precisam, é um voto de confiança de que o trabalho será feito
- A forma mais eficiente e efetiva de passar informação para e dentro do time de desenvolvimento é conversar face a face

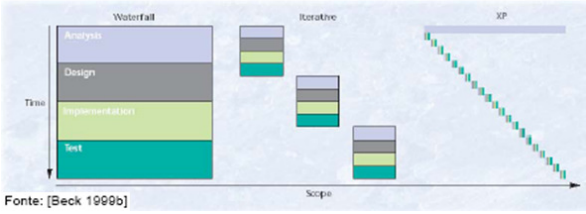
Princípios Ágeis (3/4)

- Software pronto é a principal medida de progresso
  - "DONE WHEN IT'S DONE"
- Processos ágeis mantêm o desenvolvimento sustentável. Patrocinadores, desenvolvedores e usuários devem ser capazes de manter um passo constante indefinidamente
- Atenção contínua à excelência técnica e bom projeto aumentam a agilidade

Princípios Ágeis (4/4)

- Simplicidade é essencial – arte de maximizar a quantidade de trabalho NÃO FEITO
- As melhores arquiteturas, requisitos e projetos são resultados de times auto-gerenciáveis (self organizing teams)
- Regularmente, o time reflete sobre como se tornar mais eficiente, e ajusta seu comportamento de acordo

Iterações



Fonte: [Beck 1999b]

Valores
- duração curta
- tempo fixo
- incremento palpável

Alguns critérios de comparação

	Tradicional	Ágil
Abordagem	**Previsibilidade**	**Adaptabilidade**
Mudança	Controlar	Planejar
Medida de sucesso	Aderência ao plano	Valor para o negócio
Valor	Documentação	Comunicação
Planejamento antecipado	Amplo	Mínimo
Iterações	Poucas, pré-definidas	Muitas
Ênfase	**Orientado a processos**	**Orientado a pessoas**
Estilo de Gerência	Gerência centralizada	Gerência decentralizada
Cultura	Comando/Controle	Liderança/colaboração
Retorno do Investimento	Fim do projeto	Fases iniciais projeto

Trocando o Paradigma : O que é Sucesso?

- Prazo/Cronograma
  - 62% preferem entregar de acordo com cronograma
  - 34% preferem entregar quando está pronto
- Custo
  - 28% preferem entregar dentro do orçamento
  - 60% preferem ter bom retorno de investimento (ROI)
- Funcionalidade
  - 15% construir um sistema de acordo com sua especificação
  - 82% preferem atender às necessidades de seus clientes
- Qualidade
  - 29% preferem entregar dentro do prazo/custo
  - 66% preferem entregarsistemas de alta qualidade e fáceis de manter

Trocando o Paradigma : O que é Sucesso?

- Em ordem de importância
  1. Funcionalidade
  2. Qualidade
  3. Custo
  4. Prazo/Cronograma

Métodos Ágeis e vertentes

- Extreme Programming (XP)
- Scrum
- Crystal
- Lean software Development
- Feature – Driven Design (FDD)
- Agile Modeling (e seus desdobramentos ...)
- Dynamic Systems Software Development (DSDS)
- Adaptive Software Development (ASD)
- etc

Para saber mais

- Martin Fowler. The new methodology. (resume a corrente de pensamento que levou ao manifesto ágil) www.martinfowler.com/articles/newMethodology.html (se procurar, vai achar traduzido)
- Don Wells. Agile Software Development: A gentle introduction ("tutorial pró-XP")
  www.agile-process.org/
- Acompanhe alguns links/blogs
  - Organizações
    - www.agilealliance.org/
    - www.**scrumalliance**.org
  - Mike Cohn (Mountain Goat): blog.mountaingoatsoftware.com
  - Martin Fowler (Thoughtworks): www.martinfowler.com
  - Scott Ambler: www.agilemodeling.com

Referências

- Kent Beck. Extreme Programming Explained: Embrace Change* (2nd Edition). Addisson –wesley.
- Ambler, S. Agile modeling: effective practices for eXtreme programming and the unified proces. John Wiley&Sons.*
- Cohn, M. User Stories Applied: For Agile Software Development. Pearson.
- Cohn, M. Agile Estimating and Planning. Pearson.
- Cockburn, A. Agile Software Development. Addison-Wesley, 2002.
- Crispin, L. & Gregory, J. Agile Testing. Pearson.
- Schwaber, K.; Beedle, M. Agile Software Development with SCRUM. Prentice Hall, 2001.

* Existem traduções em português.