




# Métodos Ágeis

# Métodos Ágeis

- ➔ Métodos de desenvolvimento de software que:
  - aplicam o desenvolvimento iterativo e evolucionário;
  - empregam o planejamento adaptativo;
  - promovem a entrega incremental; e
  - incluem outros valores e práticas que encorajam a agilidade - resposta rápida e flexível às mudanças.
  
- ➔ Quais tipos de problemas são melhores resolvidos pelos métodos ágeis?

Problemas caracterizados por mudança, velocidade e turbulência.



# Métodos Ágeis

- ➔ Incorporam técnicas de engenharia de software aprovadas, mas sem o *overhead* dos métodos tradicionais.

# Métodos Ágeis

→ A Aliança Ágil nasceu no início de 2001.

Participantes:

Kent Beck

**XP**

Mike Beedle

Arie van Bennekum

**DSDM**

Alistair Cockburn

**Crystal**

Ward Cunningham

Martin Fowler

James Grenning

Jim Highsmith

**ASD**

Andrew Hunt

Ron Jeffries

Jon Kern

Brian Marick

Robert C. Martin

Steve Mellor

Ken Schwaber

**Scrum**

Jeff Sutherland

Dave Thomas

# Métodos Ágeis

- ➔ De acordo com o Manifesto Ágil, os métodos ágeis valorizam
  - Indivíduos e interações mais que processos e ferramentas.
  - Um software funcionando mais que uma documentação extensa.
  - A colaboração do cliente mais que a negociação de contrato.
  - Responder a mudanças mais que seguir um plano.

↳ Apesar dos itens da direita serem importantes, os itens da esquerda são considerados mais importantes que os itens da direita.

# Métodos Ágeis


## → Princípios Ágeis

1. A prioridade é satisfazer ao cliente através de entregas de software contínuas e frequentes.
2. Receber bem as mudanças de requisitos, mesmo em uma fase avançada, dando aos clientes vantagens competitivas.
3. Entregar software em funcionamento com frequência de algumas semanas ou meses, sempre na menor escala de tempo.
4. As equipes de negócio e de desenvolvimento devem trabalhar juntas diariamente durante todo o projeto (project).
5. Manter uma equipe motivada fornecendo ambiente, apoio e confiança necessários para a realização do trabalho.
6. A maneira mais eficiente da informação circular dentro da equipe é através de uma conversa face-a-face.

# Métodos Ágeis

## → Princípios Ágeis (continuação)

7. Ter o software funcionando é a melhor medida de progresso.
8. Processos ágeis promovem o desenvolvimento sustentável. Os financiadores, desenvolvedores e usuários devem ser capazes de manter um ritmo constante.
9. Atenção contínua a excelência técnica e a um bom projeto aumentam a agilidade.
10. Simplicidade é essencial.
11. As melhores arquiteturas, requisitos e projetos provêm de equipes organizadas.
12. Em intervalos regulares, a equipe deve refletir sobre como se tornar mais eficaz e então se ajustar e adaptar seu comportamento.



# Métodos Ágeis

➔ Alguns métodos ágeis

- XP
- Scrum
- ASD
- FDD
- AM





# XP

## (eXtreme Programming)

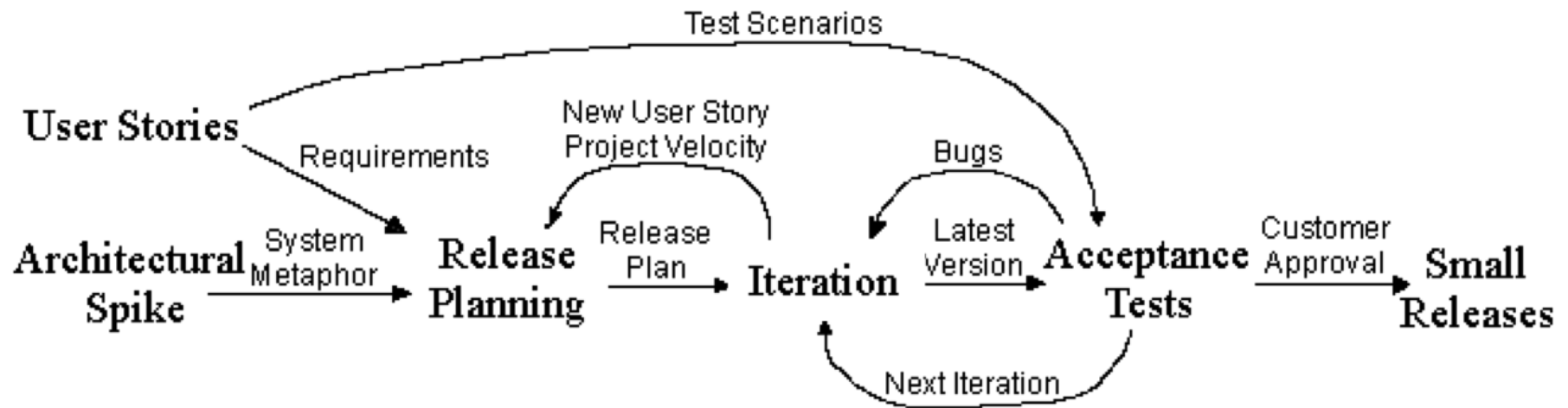
# XP

- XP foi criado por Kent Beck em 1996 e publicado em 2000.
  - Direcionado para equipes pequenas (menos de 10 pessoas).
  - Bastante indicado para projetos de risco com requisitos dinâmicos.
  - Iterações de no máximo 4 semanas.
- ➔ As idéias de XP não são novas. A maioria é antiga como a programação.

# Valores do XP

- Você precisa melhorar a comunicação. Os programadores XP se comunicam com os clientes e programadores.
- Você precisa procurar a simplicidade. Os programadores XP mantêm o projeto simples.
- Você precisa obter feedback para saber quanto bem está se saindo. O software é testado a partir do primeiro dia e entregue aos clientes o mais rápido possível.
- Você precisa sempre continuar com coragem. Os programadores XP devem ser capazes de responder às mudanças nos requisitos e tecnologia com coragem.

# Processo do XP



# Práticas do XP

**Jogo do Planejamento**

**Testes**

**Padrões de codificação**

**Entregas Frequentes**

**Cliente presente**

**Programação em pares**

**Integração contínua**

***Refactoring***

**Propriedade coletiva**

**Projeto Simples**

**Metáfora**

**Semana de 40-horas**

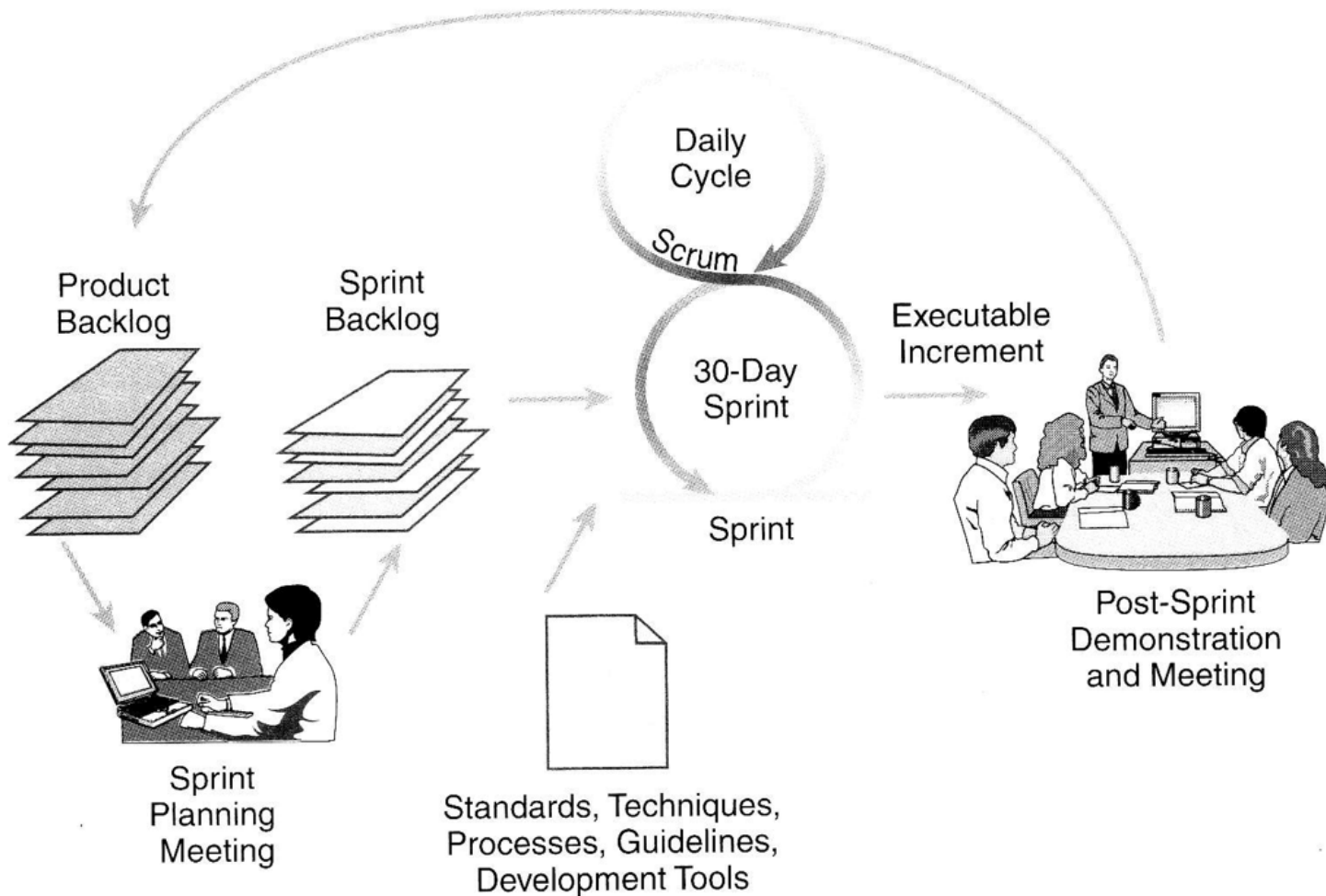


# Scrum

# Scrum

- Criado por Ken Schwaber e Jeff Sutherland nos anos 90.
  - O Scrum define um framework de gerenciamento de projeto.
  - Não existem restrições quanto à complexidade do sistema e tamanho da equipe.
  - Iterações de, no máximo, 1 mês.
  - Reuniões diárias de 15 minutos.
- ➔ Enquanto o XP enfatiza a programação (programação em par, padronização de código, refactoring), o Scrum enfatiza o gerenciamento do projeto.


# Processo do Scrum





# Processo do Scrum

- As atividades de desenvolvimento - requisitos, design, programação - não são especificadas no Scrum.  
↳ A equipe deve usar seu talento para alcançar os resultados esperados.
- No final de um Sprint, o Product Owner pode decidir descartar as características desenvolvidas e reorientar o projeto.  
↳ Entretanto, dentro de um Sprint as prioridades permanecem constantes.



# Práticas do Scrum

**Product Backlog**

**Sprint**

**Reunião de Planejamento  
da Sprint**

**Reuniões Diárias  
do Scrum**

**Sprint Backlog**

**Revisão da Sprint**



# ASD

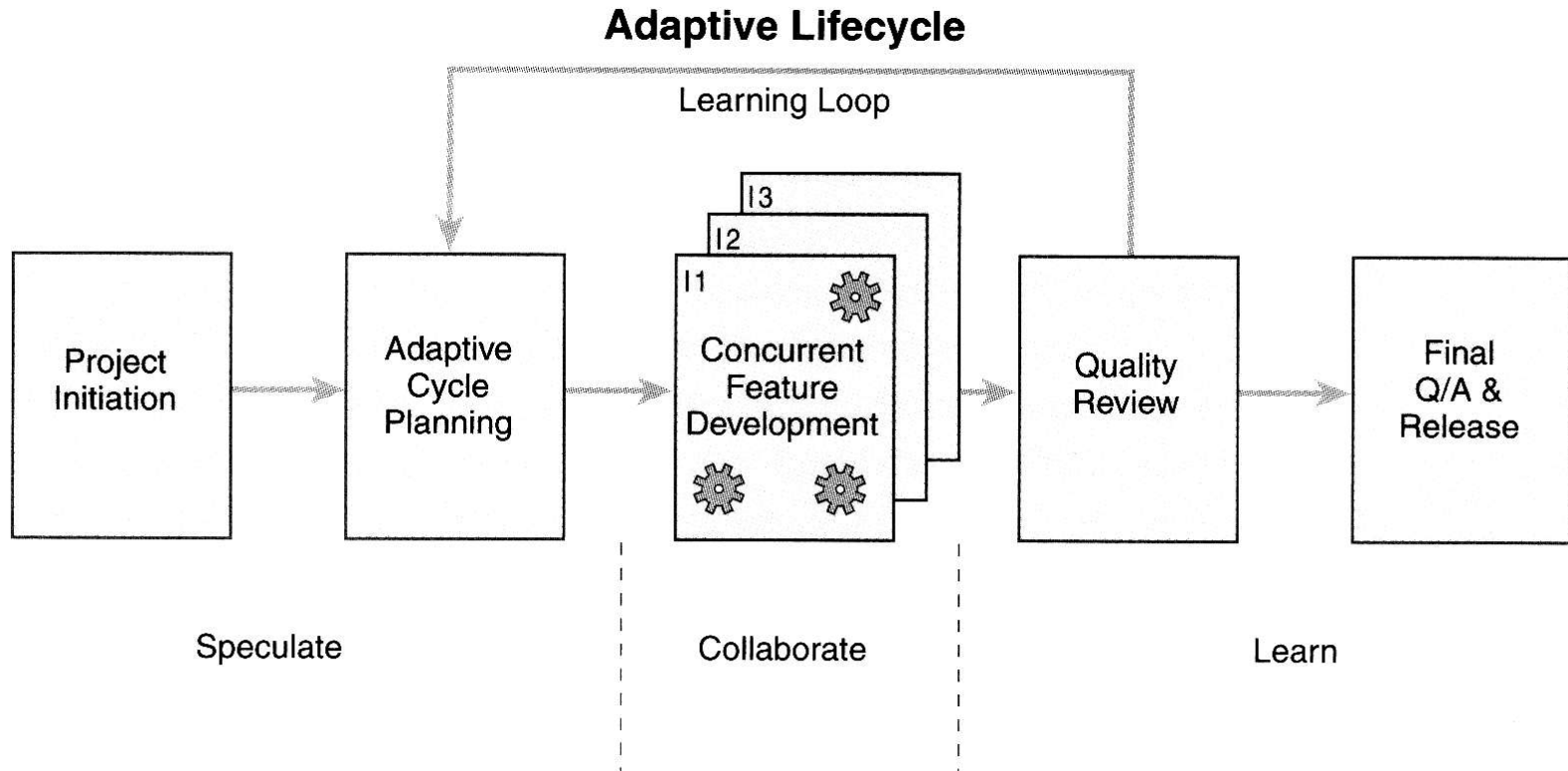
## (Adaptive Software Development)



# ASD

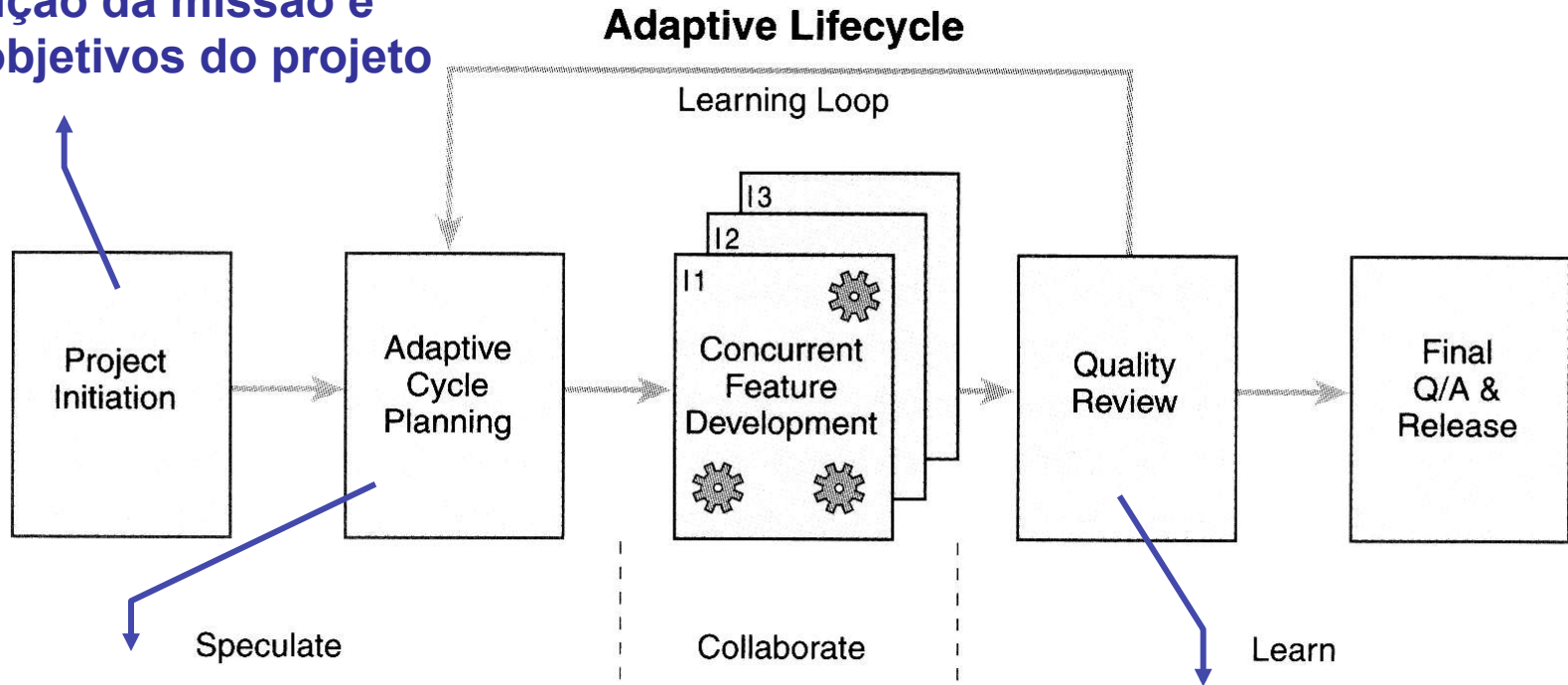
- Criado por Jim Highsmith em 2000.
- Focado nas missões.
- Iterações com duração entre 4 e 8 semanas.
- Não existem restrições quanto a complexidade do sistema e tamanho da equipe.

# Processo do ASD



# Processo do ASD

definição da missão e dos objetivos do projeto



definição do número de iterações e um cronograma para cada; objetivo de cada iteração; atribuição de requisitos para cada iteração

qualidade sob a perspectiva do cliente; qualidade sob a perspectiva técnica; revisão da utilização das práticas por parte da equipe; progresso geral (status) do projeto

# Processo do ASD

- Início: definição da missão e dos objetivos do projeto.
- Planejamento: definição do número de iterações e um cronograma para cada; objetivo de cada iteração; atribuição de requisitos para cada iteração.
- Desenvolvimento Concorrente dos Requisitos:
- Revisão da Qualidade: qualidade sob a perspectiva do cliente; qualidade sob a perspectiva técnica; utilização das práticas por parte da equipe; progresso (status) do projeto.

# Práticas do ASD

**Focado nas missões**

**Prazos pré-fixados**

**Baseado em características**

**Tolerância a mudanças**

**Iterativo**

**Orientado a riscos**





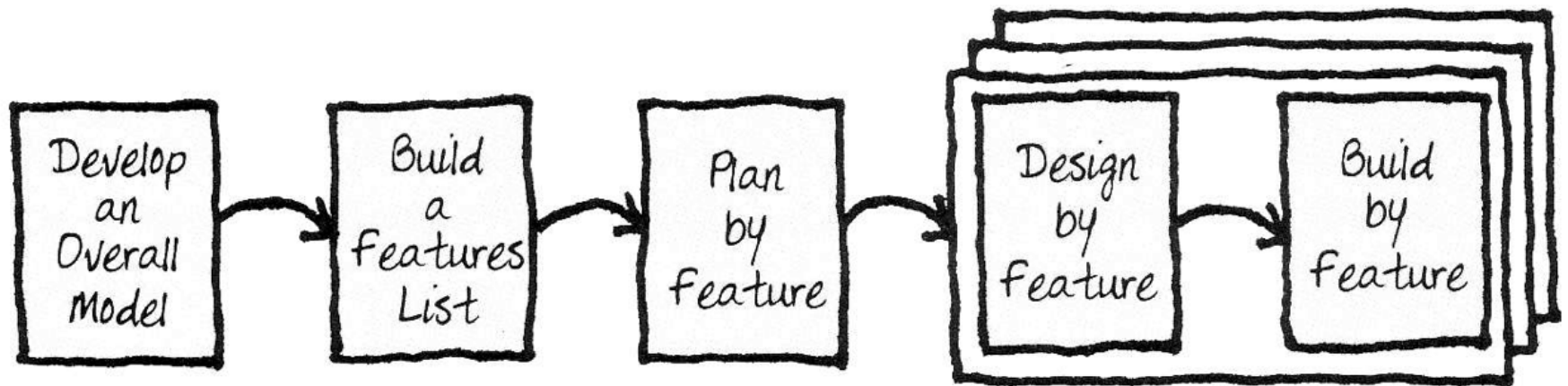
# FDD

## (Feature-Driven Development)



# FDD

- Criado por Peter Coad e Jeff De Luca e utilizado pela primeira vez em 1998.
  - Desenvolvimento dirigido a características.
  - Iterações de no máximo 2 semanas.
  - Não existem restrições quanto à complexidade do sistema e tamanho da equipe.
- 
- ➔ Usado em projetos com sucesso. (1) 50 pessoas durante 15 meses; (2) 250 pessoas durante 18 meses.
  - ➔ No FDD não é feito muito refactoring, como no XP.



# Processo do FDD

- **Desenvolver um modelo global:** Um modelo de objetos em alto-nível. Os especialistas no domínio participam.
- **Construir um Lista de Características:** Característica é “uma função importante para o cliente que pode ser implementada em duas semanas ou menos”.
- **Planejar a construção por característica:** Cada atividade de negócio é atribuída a um chefe de programação e cada classe é atribuída a um programador.
- **Projetar por característica:** O chefe de programação realiza o projeto – diagramas de seqüência - e refina o modelo de objetos.
- **Construir por característica:** Implementa classes e métodos; Inspeção do código; Teste de unidade; Realização do *build*.

# Práticas do FDD

**Modelagem dos Objetos  
de Domínio**

**Inspeções**

**Modelagem dos Objetos  
de Domínio**

**Construções Regulares**

**Equipes de Características**

**Administração de  
Configuração**

**Propriedade Individual  
da Classe**

**Relatório dos resultados**

# Outros Métodos Ágeis

- DSDM (Dynamic System Development Method), criado por Arie van Bennekum
- Crystal Methods, criados por Alistair Cockburn.
- LD (Lean Development), criado por Bob Charette.



# Agile Modeling (Modelagem Ágil)



# Modelagem Ágil

- Criado por Scott W. Ambler em 2002.
- Foco na geração de modelos e documentação eficazes.
- Deve ser utilizada em conjunto com outros métodos de desenvolvimento.
- Não trata de nenhuma técnica de modelagem específica.



# Práticas da Modelagem Ágil

## **Iterativa e Incremental**

Aplicar o(s) artefato(s) correto(s)  
Criar diversos modelos em paralelo  
Iterar em outro artefato  
Modelagem incremental

## **Trabalho em Equipe**

Modelagem em conjunto  
Participação ativa do cliente  
Posse coletiva  
Visualização dos modelos

## **Motivação**

Modelar para entender  
Modelar para comunicar

## **Simplicidade**

Criação de conteúdo simples  
Apresentação de modo simples  
Utilização de ferramentas simples

## **Validação**

Considerar a testabilidade  
Comprovar com código

## **Documentação**

Descartar modelos temporários  
Formalização dos modelos de contrato  
Atualizações somente quando necessário

## **Produtividade**

Aplicar as convenções de modelagem  
Utilizar padrões com moderação  
Reuso dos recursos já existentes

# Modelagem Ágil

## → Sessões Ágeis

- Duração das sessões varia de 10 min a 3 dias.
- Após cada sessão devem ser feitas validações.
- Foco na criação de modelos relacionados às fases mais importantes do desenvolvimento, como requisitos, análise e projeto.

## → Documentação Ágil

- Só deverá ser criada se é realmente necessária.
- O benefício de ter documentação deve ser maior do que o custo de criá-la e mantê-la.
- Cada sistema tem suas necessidades de documentação.