

Desenvolvimento Ágil

Curso de Ciência da Computação
Disciplina de Engenharia de Software I
Prof. Humberto Torres Marques Neto

Agosto de 2018



Objetivo

- Apresentar e discutir os modelos de desenvolvimento ágil de software

Referências Bibliográficas

Básica:

PRESSMAN, Roger S. Engenharia de Software: uma abordagem profissional. 8 ed. McGraw-Hill, 2016.

Complementar:

SOMMERVILLE, Ian. Engenharia de software. 9. ed. São Paulo: Pearson Prentice Hall, 2011.

O que é “agilidade”?

(1/2)

- Eficácia (rápido e adaptativo) para resposta a mudanças
- Eficácia na comunicação entre os *stakeholders* (envolvidos)

O que é “agilidade”?

(2/2)

- Envolvimento do cliente na equipe de desenvolvimento
- Organização da equipe para entregar rapidamente incrementos de software

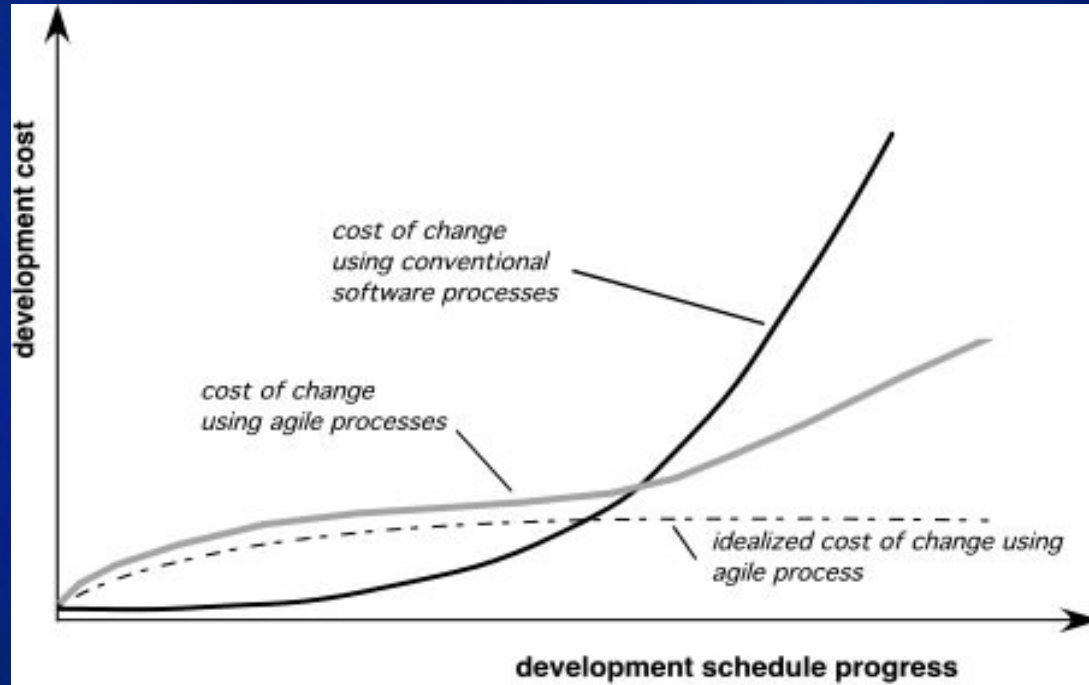
Produzindo...

- rapidamente, entrega incremental de software

Agilidade para...

*...produzir e entregar rapidamente
incrementos de software!*

Agilidade e o custo da mudança



PRESSMAN, 2011, p.83

O Manifesto do Desenvolvimento de Software Ágil

(Kent Beck et. al., 2001) (1/2)

“Desenvolvendo e ajudando outros a desenvolver software, estamos desvendando formas melhores de desenvolvimento. Por meio deste trabalho, passamos a valorizar:

O Manifesto do Desenvolvimento de Software Ágil

(Kent Beck et. al., 2001) (2/2)

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

Ou seja, embora haja valor nos itens à direita, valorizaremos os da esquerda mais ainda.”

Princípios da agilidade da *Agile Alliance*

1. A maior prioridade é satisfazer o cliente por meio de entrega adiantada e contínua de software valioso
2. Acolha bem os pedidos de alterações, mesmo atrasados no desenvolvimento; os processos ágeis se aproveitam das mudanças como uma vantagem competitiva na relação com o cliente
3. Entregue software em funcionamento frequentemente, de algumas semanas para alguns meses, dando preferências a intervalos mais curtos

Princípios da agilidade da *Agile Alliance*

4. O pessoal comercial e os desenvolvedores devem trabalhar em conjunto diariamente ao longo de todo o projeto
5. Construa projetos em torno de indivíduos motivados; dê a eles o ambiente e apoio necessários e confie neles para ter o trabalho feito
6. O método mais eficiente e efetivo de transmitir informações para e dentro de uma equipe de desenvolvimento é uma conversa aberta, de forma presencial

Princípios da agilidade da *Agile Alliance*

- 7. Software em funcionamento é a principal medida de progresso
- 8. Os processos ágeis promovem desenvolvimento sustentável; os proponentes, desenvolvedores e usuários devem estar capacitados para manter um ritmo constante indefinidamente
- 9. Atenção contínua para com a excelência técnica e para com bons projetos aumenta a agilidade

Princípios da agilidade da *Agile Alliance*

- 10.Simplicidade – a arte de maximizar o volume de trabalho não efetuado – é essencial
- 11.As melhores arquiteturas, requisitos e projetos emergem de equipes que se auto organizam
- 12.A intervalos regulares, a equipe se avalia para ver como tornar-se mais eficiente, então sintoniza e ajusta seu comportamento de acordo

A política do desenvolvimento ágil

- Grande discussão acerca das metodologias tradicionais versus metodologias ágeis
- Não se pode apontar melhor ou pior
- O ideal é aplicar o melhor de cada uma?

Fatores humanos (1/2)

- O processo se adapta às necessidades das pessoas e equipes específicas, e não o caminho inverso
- Deve haver traços chave entre as pessoas de uma equipe ágil e a equipe em si:

(...)

Fatores humanos (2/2)

- Competência
- Foco comum
- Colaboração
- Habilidade na tomada de decisão
- Habilidade de solução de problemas confusos
- Confiança mútua e respeito
- Auto-organização

Processos Ágeis

Processo Ágil

- Dirigido por cenários construídos pelos clientes com seus requisitos
- Planejamento tem vida curta
- Desenvolvimento iterativo com uma forte ênfase nas atividades de construção
- Entrega de múltiplos incrementos de software
- Adaptação às mudanças

Extreme Programming (XP)

- Também conhecido como Programação Extrema
- Foi proposto por Kent Beck em 2004
- Processo ágil mais utilizado

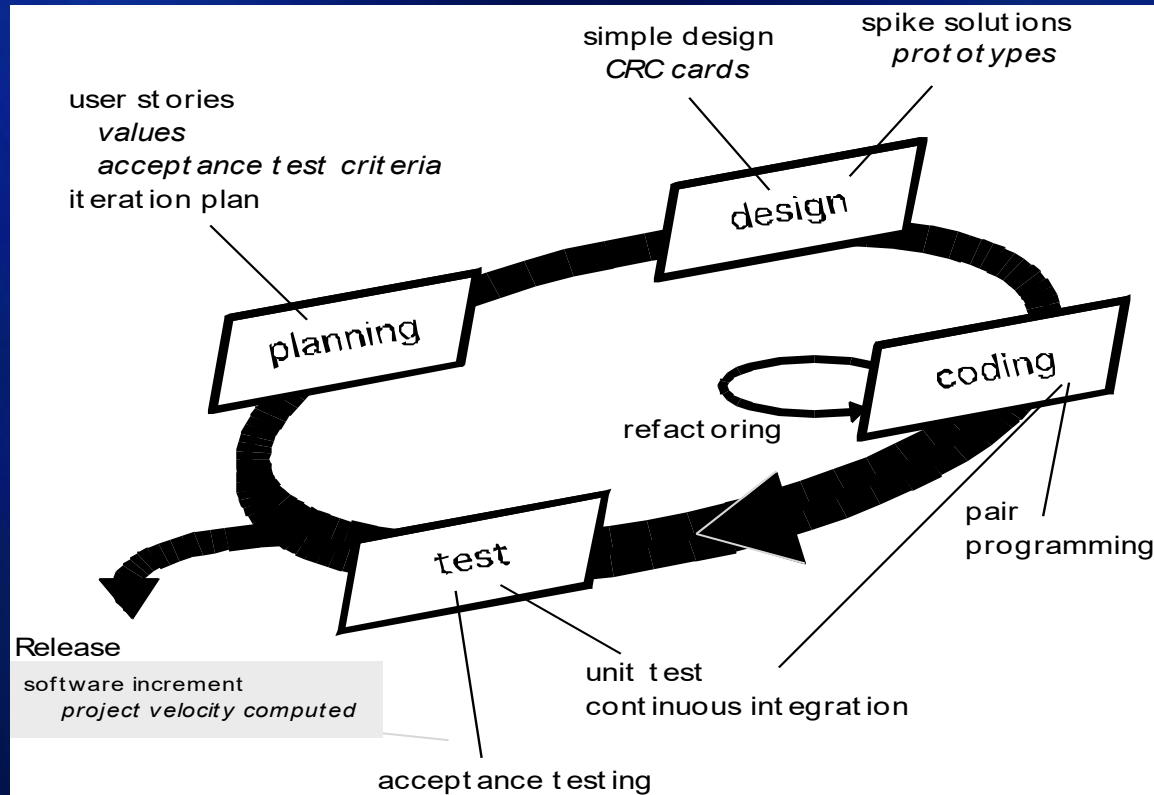
Valores básicos do XP (1/2)

- Comunicação
 - Relação estreita e informal
- Simplicidade
 - Fazer somente o necessário
- Feedback
 - Devem ser constantes, à medida do desenvolvimento, com testes e entregas

Valores básicos do XP (2/2)

- Coragem (ou disciplina)
 - Disciplina para pensar somente no presente, mas bem feito
- Respeito
 - Conforme as entregas vão acontecendo, o respeito pela equipe e pelo software vai aumentando

Extreme Programming (XP)



Processo XP: *Planejamento* (1/2)

- Comece com a criação das “histórias dos usuários”
- A equipe avalia as histórias e estima o custo
- As histórias são agrupadas em um incremento de software

Processo XP: *Planejamento* (2/2)

- Define-se uma data de entrega do incremento
- O primeiro incremento é utilizado como referência para acelerar os demais

Processo XP: *Desenho*

- Segue o princípio KIS (*keep it simple*)
- Encoraja o uso de cartões CRC
- Situações de dificuldade são resolvidas com prototipação
- Encoraja a *refatoração* – refinamento iterativo do desenho interno do programa

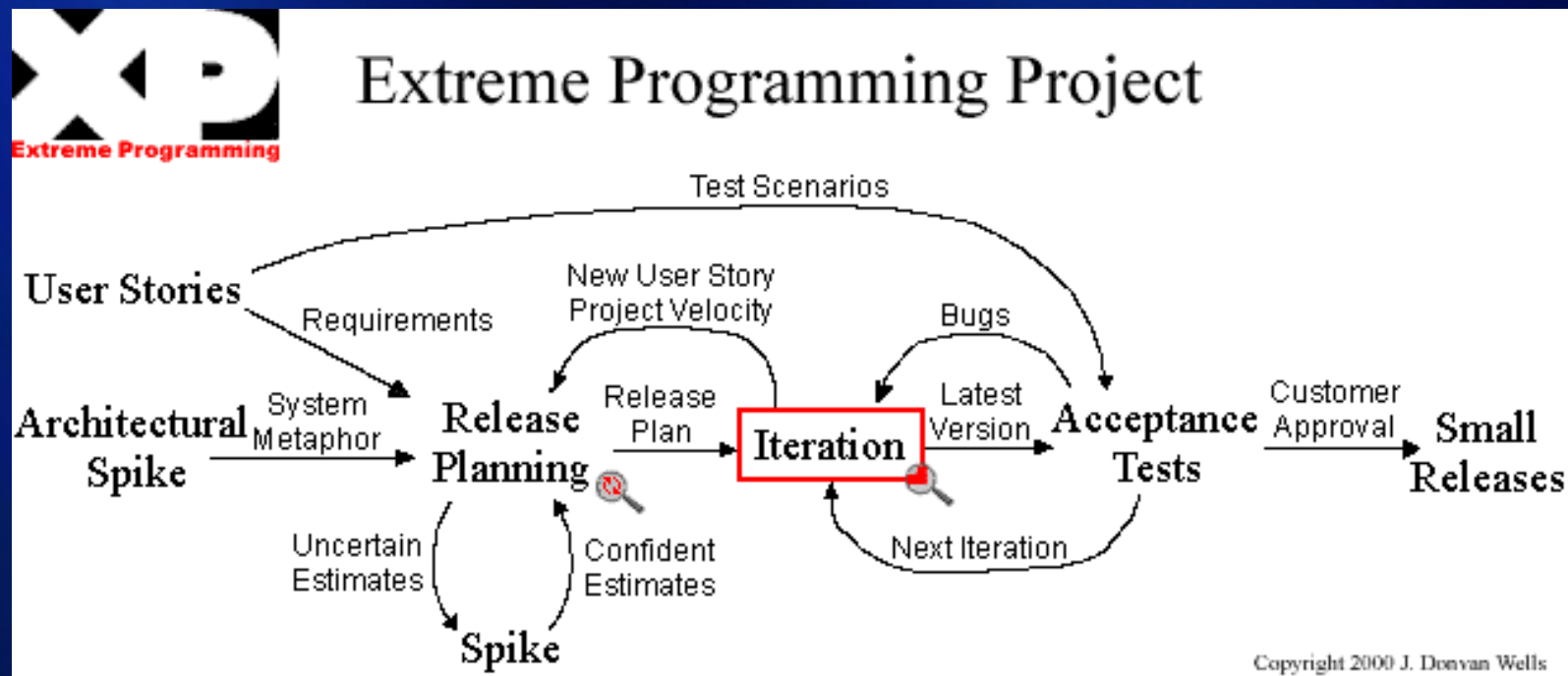
Processo XP: *Codificação*

- Recomenda a construção de testes de unidade antes do início da codificação
- Encoraja a programação em par

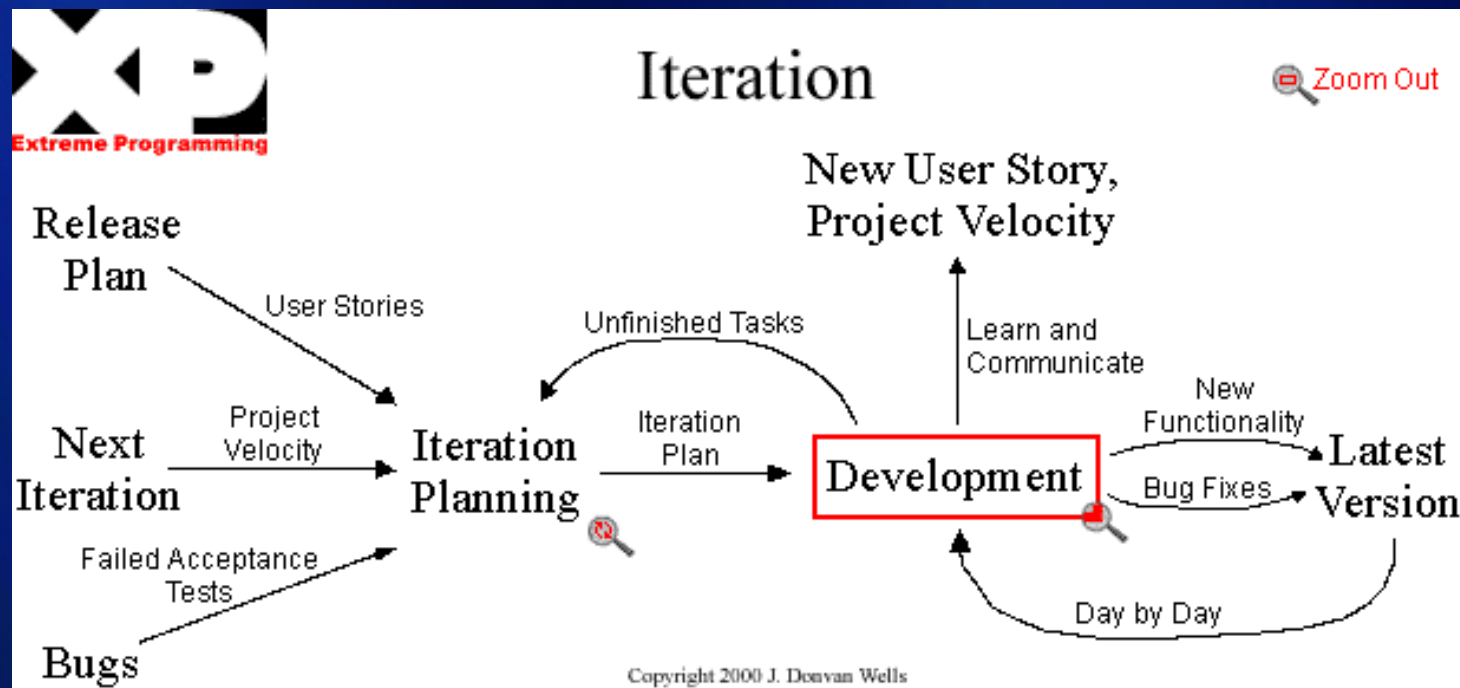
Processo XP: *Teste*

- Todos os testes de unidade são executados diariamente
- Testes de aceitação são definidos pelo cliente para avaliação de uma funcionalidade

Ciclo de Vida dos Projetos XP



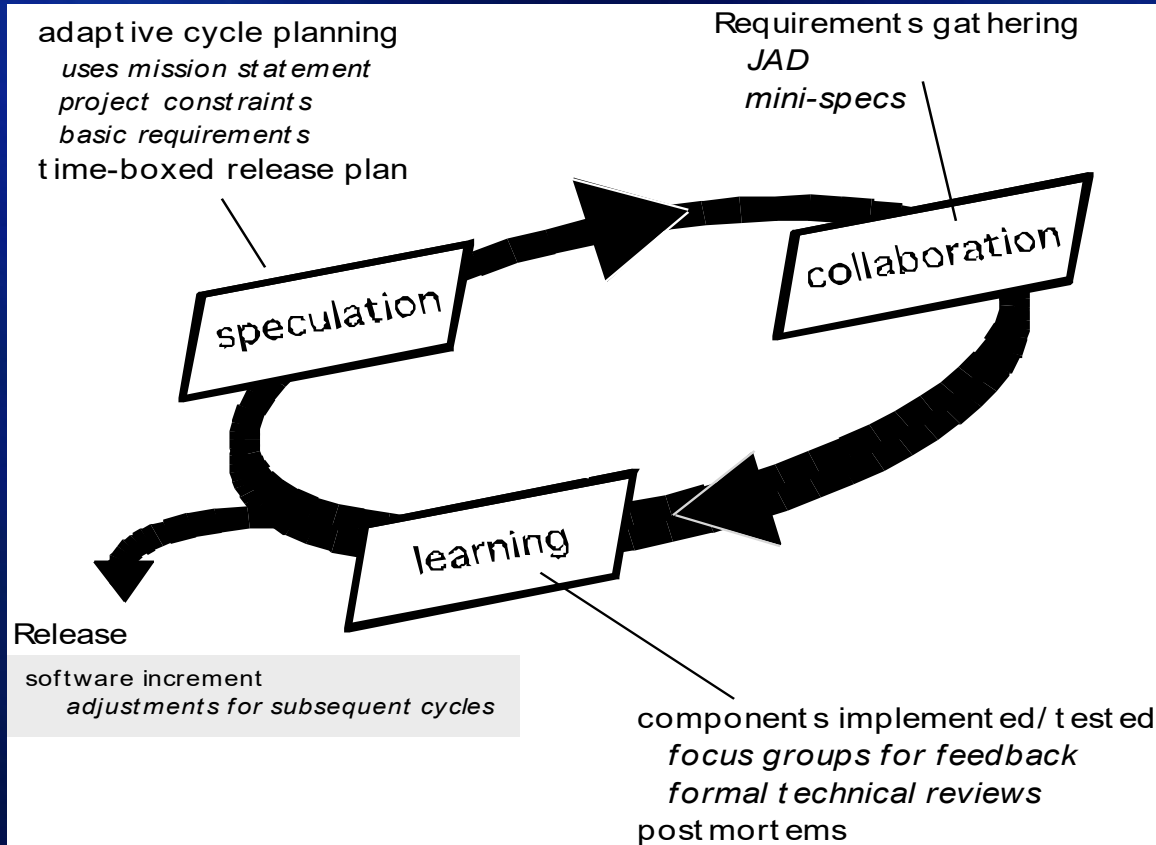
Ciclo de Vida de uma Iteração XP



Industrial XP (IXP)

- Evolução orgânica do XP
 - Avaliação imediata
 - Comunidade de projeto
 - Mapeamento de projeto
 - Gerenciamento orientado a testes
 - Retrospectivas
 - Aprendizagem contínua

Desenvolvimento de Software Adaptativo



Desenvolvimento de Software Adaptativo

- Proposto por Jim Highsmith em 2000
- Características do ASD (*Adaptive Software Development*)
 - Planejamento dirigido a missão
 - Foco no desenvolvimento baseado em componentes
 - Uso de “time-boxing”

Desenvolvimento de Software Adaptativo

- Mais características do ASD
 - Consideração explícita de riscos
 - Ênfase na colaboração para entendimento dos requisitos
 - Ênfase na “aprendizagem” durante o processo

Método Dinâmico de Desenvolvimento de Sistemas (1/3)

- Promovido pelo consórcio DSDM – *Dynamic Systems Development Method*, www.dsdm.org)
- Características do DSDM
 - Similar ao XP e ao ASD em muitos aspectos

Método Dinâmico de Desenvolvimento de Sistemas (2/3)

- Princípios:
 1. O envolvimento ativo do usuário é imperativo
 2. As equipes devem ter autonomia para tomada de decisão
 3. O foco é a entrega frequente de produtos
 4. Conveniência para o negócio é um critério essencial para aceitação da entrega

Método Dinâmico de Desenvolvimento de Sistemas (3/3)

- Princípios:
 5. Desenvolvimento iterativo e incremental é necessário para convergência de uma solução de negócio correta
 6. Todas as mudanças durante o desenvolvimento são reversíveis
 7. Requisitos são marcos de alto nível
 8. Testes são integrados ao longo do ciclo de vida

Crystal

- Proposto por Cockburn e Highsmith em 2002
- Características
 - Permite a adaptabilidade baseada nas características do problema
 - Ênfase na comunicação face a face
 - Sugere-se o uso de *workshops* de reflexão para revisão da forma de trabalho da equipe

Desenvolvimento direcionado por características

- Proposto por Peter Coad em 1999
- Características do FDD (*Feature Driven Development*)
 - Ênfase na definição de características
 - ✓ Uma característica (*feature*) é “uma função valorizada pelo cliente que pode ser implementada em duas semanas ou menos”

Desenvolvimento direcionado por características

- Mais características do FDD
 - Utiliza o seguinte *template*
<action> the <result> <by | for | of | to> a(n) <object>
 - A lista de características é criada e o planejamento por características é conduzido
 - O desenho se funde com a construção no FDD

Processo Unificado Ágil (AUP)

- Adota uma filosofia:
 - Serial para o que é amplo
 - Iterativa para o que é particular

Atividades do AUP

- Modelagem
- Implementação
- Teste
- Aplicação (entrega)
- Configuração e gerenciamento de projeto
- Gerenciamento do ambiente

Scrum

Declaração de Interdependência (DOI) da Gestão de Projeto Ágil (2005)

“Somos uma comunidade de líderes de projeto que tem sido altamente bem-sucedida em entregar resultados. Para alcançar tais resultados:”

- ***Aumentamos o retorno de investimento***, tornando o fluxo contínuo de valor o nosso foco;
- ***Entregamos resultados confiáveis***, engajando interações frequentes e propriedade compartilhada;

Declaração de Interdependência (DOI) da Gestão de Projeto Ágil (2005)

- ***Esperamos incertezas*** e gerenciamos levando-as em conta, por meio de interações, antecipação e adaptação;
- ***Promovemos criatividade e inovação*** reconhecendo que os indivíduos são a fonte última e criamos um ambiente em que eles fazem a diferença;

Declaração de Interdependência (DOI) da Gestão de Projeto Ágil (2005)

- ***Impulsionamos o desempenho*** por meio do compromisso do grupo em obter resultados e da responsabilidade compartilhada pela eficácia do grupo;
- ***Melhoramos a eficácia e a confiabilidade*** por meio de estratégias situacionais específicas, processos e práticas.”

Origem do SCRUM

TAKEUCHI, Hirotaka, NONAKA, Ikujiro. The new new product development game. *Harvard Business Review*, January 1986 Issue.

“... equipes de projeto são compostas de pequenas equipes multifuncionais, trabalhando Com sucesso rumo a um objetivo comum, que os autores compararam à formação Scrum do rugby.”

Origem do SCRUM



Origem do SCRUM

- A pedido do Object Management Group (OMG), Jeff Sutherland e Ken Schwaber, em 1995, publicaram:

Scrum and the perfect storm
(Scrum e a tempestade perfeita)

Como o SCRUM funciona

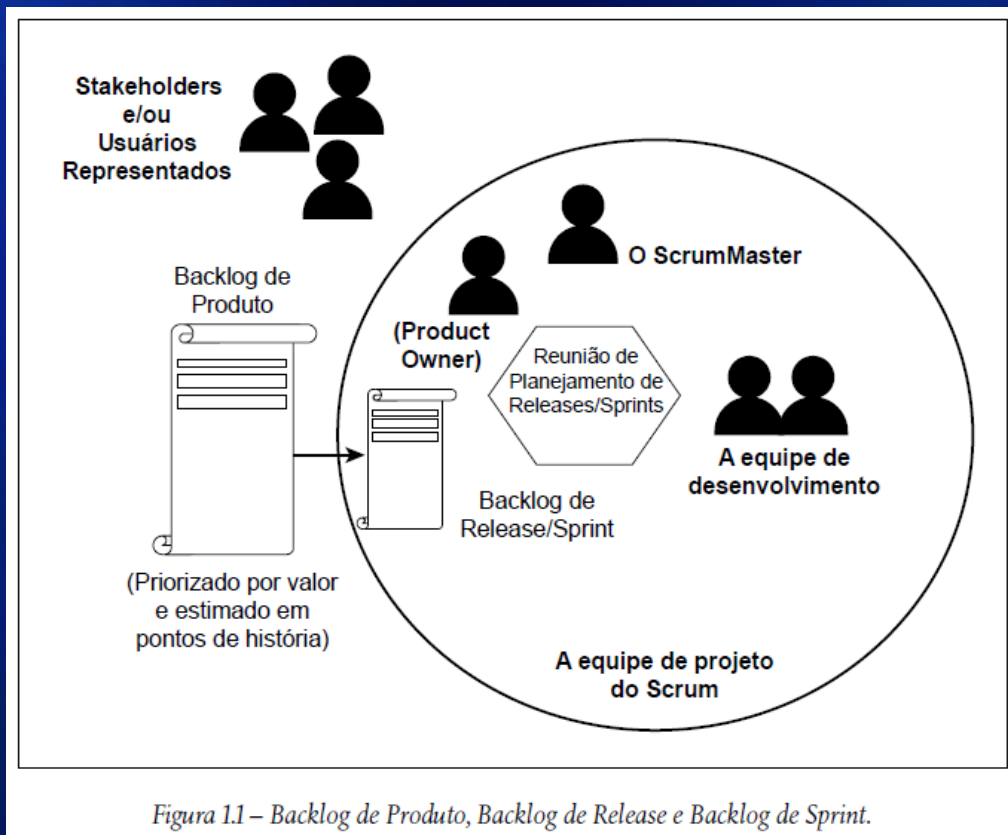


Figura 1.1 – Backlog de Produto, Backlog de Release e Backlog de Sprint.

Início de um projeto SCRUM

- Um projeto SCRUM é iniciado quando o *Product Owner*, responsável por obter informações dos *stakeholders*, ou usuários que os representem, elaboram uma lista de requisitos e criam um *Backlog* de Produto.

O que é um *backlog* de produto?

- Lista de requisitos priorizada, que pode incluir de tudo:
 - aspectos do negócio a tecnologias
 - questões técnicas
 - correções de bugs

Criação de um *backlog* de produto

- Os requisitos dos usuários costumam ser coletados como histórias de usuários curtas
- Workshop de um ou dois dias, anterior à reunião de Planejamento de Releases e à reunião de Planejamento de Sprints

Planejamento de Releases

- Socialização da agenda de entregas prováveis
- Duração: cerca de quatro horas para cada Sprint de quatro semanas

Planejamento de Sprints

- Duração: cerca de oito horas para cada Sprint de quatro semanas (ou quatro horas para Sprint de duas semanas)
- Geralmente é dividido em duas reuniões de quatro horas cada uma

Planejamento de Sprint: Parte I

- O *Product Owner* organiza, com a participação colaborativa da equipe, os requisitos (histórias dos usuários) e definem, também juntos, os objetivos
- Cria-se um Backlog de Sprint, que, geralmente, não pode ser modificado

"O quê" deve ser feito no Sprint?

Planejamento de Sprint: Parte II

- Identificação de tarefas com base nas histórias dos usuários
- Estimativa de tempo para os incrementos que serão entregues
- Construção de um Quadro de Tarefas (*Task Board*)

"Como" o Sprint será realizado?

Funcionamento do Sprint

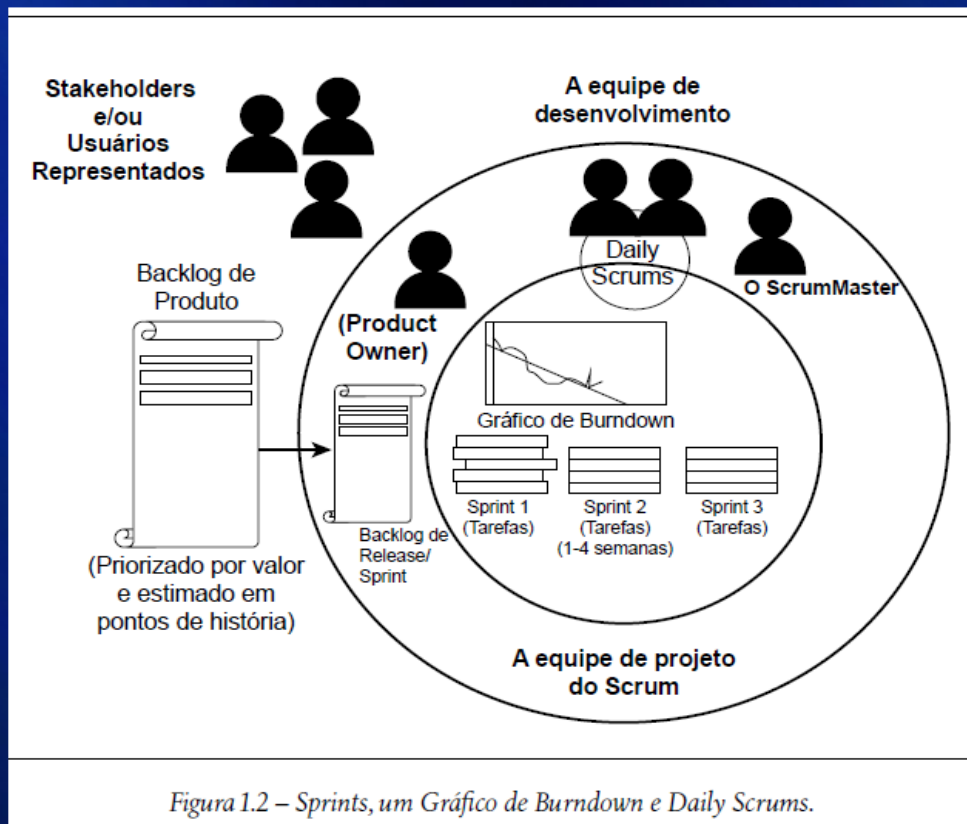


Figura 1.2 – Sprints, um Gráfico de Burndown e Daily Scrums.

Reuniões diárias

- *Daily Scrum* ou *Daily Standup*
- Duração: 15 minutos
- Pauta:
 - a) O foi feito desde a última reunião?
 - b) Quais foram os obstáculos?
 - c) O que será feito?
- Atualização do Gráfico de *Burndown*

Revisão do Sprint

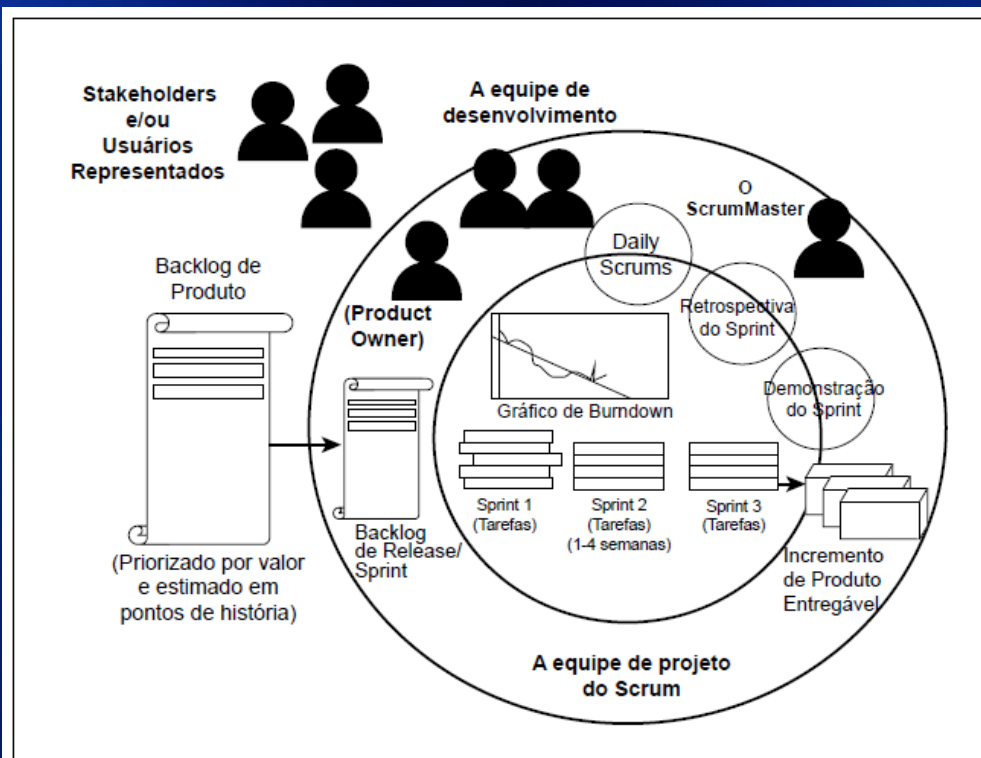


Figura 1.3 – Revisão do Sprint e Retrospectiva.

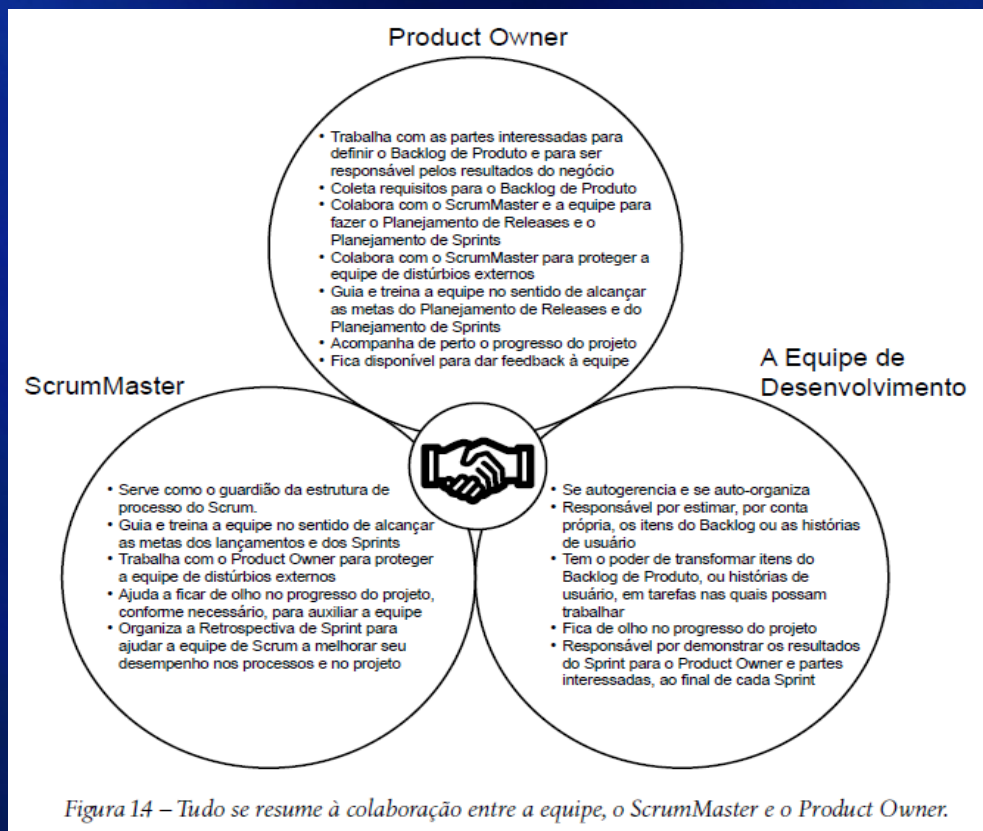
Reuniões de Revisão de Sprint

- Duração: cerca de uma hora para cada semana de Sprint
- Realizada com o *Product Owner* para:
 - a) Relatar o que foi feito para obter *feedback*
 - b) Apresentar e explicar o que não foi feito
 - c) Atualizar o *roadmap* do produto

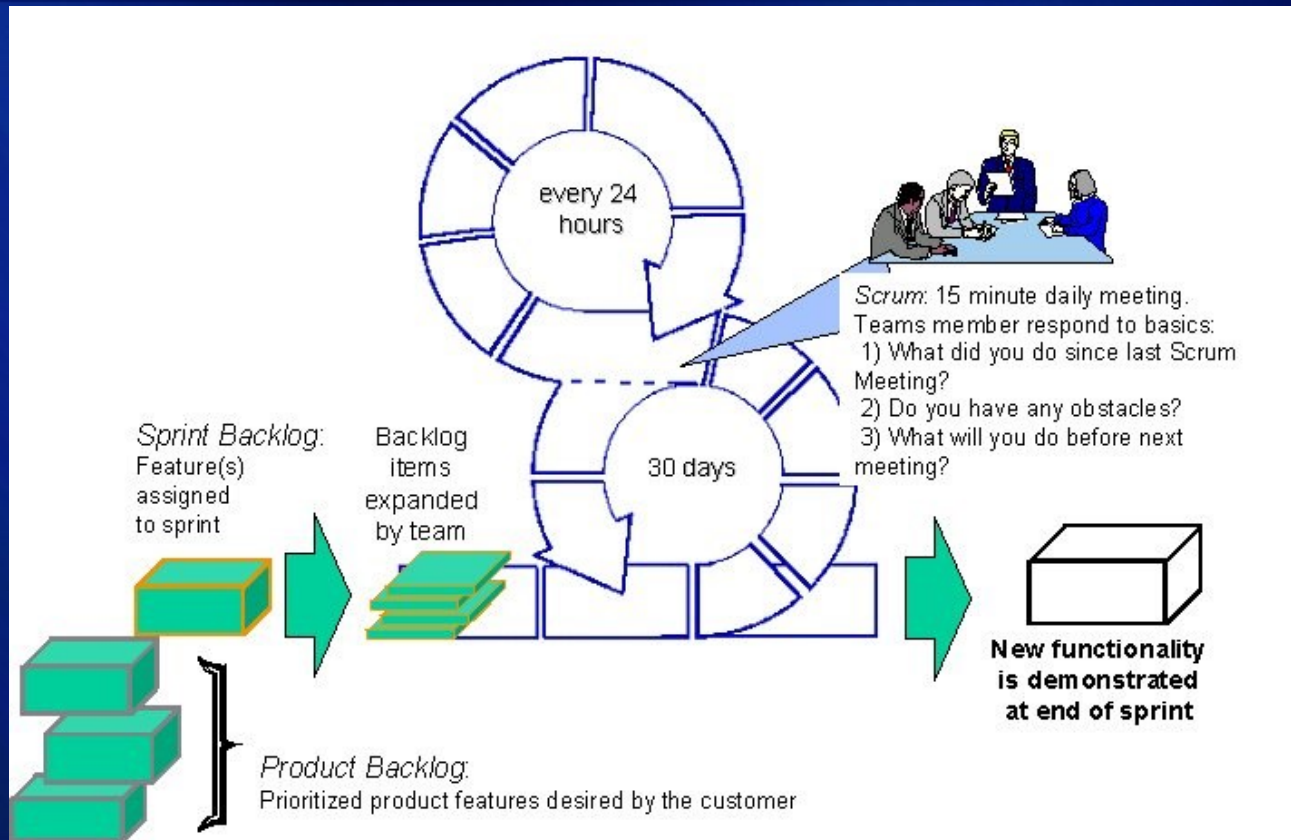
Reuniões de Retrospectiva

- Duração: cerca de uma hora para cada semana de Sprint
- O que funcionou e o que não funcionou durante o Sprint atual?
- O que pode melhorar no próximo Sprint?

Responsabilidades



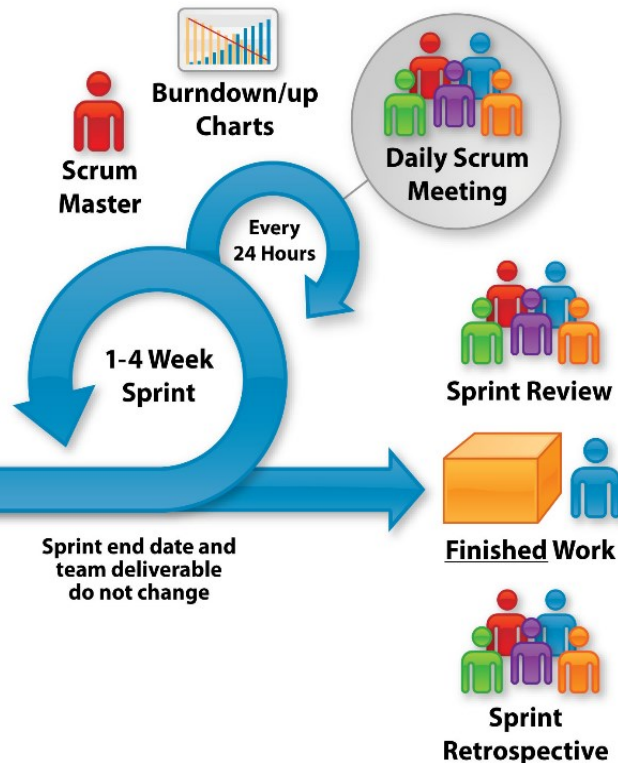
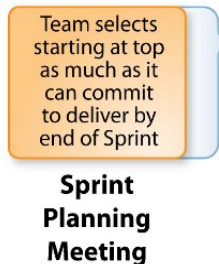
SCRUM Process Flow



SCRUM Process Flow

The Agile: Scrum Framework at a glance

Inputs from Executives,
Team, Stakeholders,
Customers, Users



Modelagem Ágil

Modelagem Ágil

- A Modelagem Ágil (MA), proposta por Scott Ambler, é uma metodologia baseada na prática para modelagem e documentação eficazes de sistemas baseado em software
- A MA não é um processo prescritivo
- A MA não significa menos modelagem

Por que modelar?

- Para entender qual software será construído
- Para melhorar a comunicação dentro da equipe
- Para facilitar a comunicação com o cliente

Objetivos da Modelagem Ágil

1. Definir e mostrar como colocar em prática um conjunto de valores, princípios e práticas relativas a uma modelagem eficaz e leve
2. Lidar com a questão de como aplicar técnicas de modelagem em projetos de software adotando uma perspectiva ágil

Objetivos da Modelagem Ágil

3. Discutir como você pode melhorar suas atividades de modelagem adotando uma perspectiva “quase ágil” para o desenvolvimento de software e equipes de projeto que adotaram uma instanciiação do Processo Unificado como o RUP ou o EUP (Enterprise Unified Process)

Modelagem Ágil e o Processo de Software

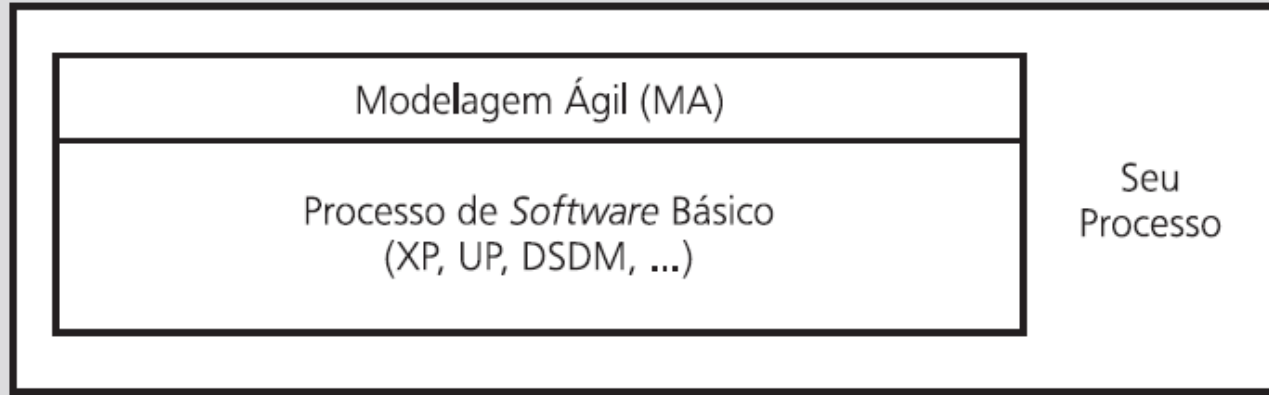


FIGURA 1.1 A MA aperfeiçoa outros processos de *software*.

Modeladores Ágeis (1/2)

1. Software é seu objetivo primário
2. Habilitar o próximo trabalho é sua meta secundária
3. Modelar deve ser feito com um propósito
4. Precisa reconhecer que necessita de múltiplos modelos

Modeladores Ágeis (2/2)

- 5. O conteúdo é mais importante que a forma
- 6. Precisa conhecer seus modelos
- 7. Todos podem aprender com todos
- 8. Deve trabalhar com os instintos das pessoas
- 9. Comunicação aberta e honesta

O que são Modelos Ágeis? (1/2)

1. Os modelos ágeis cumprem seu propósito
2. Os modelo ágeis são compreensíveis
3. Os modelos ágeis são suficientemente precisos
4. Os modelos ágeis são suficientemente consistente

O que são Modelos Ágeis? (2/2)

- 5. Os modelos ágeis são suficientemente detalhados
- 6. Os modelos ágeis proporcionam valor positivo
- 7. Os modelos ágeis são os mais simples possíveis

O que é (ou não é) Modelagem Ágil? (1/3)

1. A MA é uma atitude, não um processo prescritivo
2. A MA é um suplemento dos métodos preexistentes; não uma metodologia completa
3. A MA é complementar aos processos de modelagem

O que é (ou não é) Modelagem Ágil? (2/3)

- 5. A MA é uma maneira de trabalhar em conjunto de modo eficaz para alcançar os objetivos dos clientes do projeto
- 6. A MA é eficaz e trata de eficácia
- 7. A MA é algo que funciona na prática; não uma teoria acadêmica

O que é (ou não é) Modelagem Ágil? (3/3)

- 8. A MA não é uma “bala de prata”
- 9. A MA foi feita para o desenvolvedor médio, mas não é uma substituição de pessoas competentes
- 10. A MA não é um ataque à documentação
- 11. A MA não é um ataque às ferramentas CASE

Valores da Modelagem Ágil

- Comunicação
- Simplicidade
 - ✓ KISS
- Retorno
- Coragem
- Humildade

Princípios Básicos da Modelagem Ágil (1/2)

1. Software é seu objetivo principal
2. Possibilitar o próximo trabalho é o seu objetivo secundário
3. Diminua a carga de trabalho de criação e manutenção dos modelos
4. Adote a simplicidade
5. Encampe a mudança

Princípios Básicos da Modelagem Ágil (2/2)

6. Mude incrementalmente
7. Modele com um propósito
8. Tenha mais de um modelo
9. Trabalho de qualidade
10. Retorno rápido
11. Maximize o retorno que seus cliente obterão

Princípios Suplementares da MA

- O conteúdo é mais importante que a forma
- Todos podem aprender com todos
- Conheça seus modelos
- Adaptação ao local de prática da MA
- Comunicação aberta e honesta
- Trabalhe com o instinto das pessoas

Práticas Básicas de Modelagem Ágil

1. Modelagem Iterativa e Incremental

- Aplique o(s) artefato(s) correto(s)
- Crie diversos modelos em paralelo
- Itere em outro artefato
- Modele incrementalmente

Práticas Básicas de Modelagem Ágil

2. Trabalho em equipe

- Modele com outras pessoas
- Organize uma participação ativa dos clientes
- Promova a posse coletiva
- Mostre os modelos publicamente

Práticas Básicas de Modelagem Ágil

3. Simplicidade

- Crie conteúdo simples
- Mostre os modelos de modo simples
- Use as ferramentas mais simples

Práticas Básicas de Modelagem Ágil

4. Validação

- Considere a testabilidade
- Comprove com código

Práticas suplementares de Modelagem Ágil

1. Produtividade

- Aplique as convenções (standards) da modelagem
- Utilize os padrões (*patterns*) com moderações
- Reuse os recursos já existentes

Práticas suplementares de Modelagem Ágil

2. Documentação

- Descarte os modelos temporários
- Formalize os modelos de contrato
- Atualize apenas quando necessário

Práticas suplementares de Modelagem Ágil

3. Motivação

- Modele para entender
- Modele para comunicar