



UNIVERSIDADE ESTADUAL  
VALE DO ACARAÚ



GOVERNO DO  
ESTADO DO CEARÁ

*Secretaria da Ciência, Tecnologia  
e Educação Superior*



Núcleo de Educação a Distância  
Universidade Estadual Vale do Acaraú

# Engenharia de Software

Prof. Thales Damasceno de Andrade

# 2º Bloco

Engenharia de Software



## Capítulo 4

# Desenvolvimento Ágil

## Introdução



# 2º Bloco

## Engenharia de Software

### Melhorias

- Valorizar pessoas
- Interações
- Funcionamento do programa
- Cliente participativo
- Alterações

### Fundamentos

- Satisfação do cliente
- Trabalho em pequenos grupos
- Motivação
- Informalidade
- Simplicidade no desenvolvimento

# 2º Bloco

## Engenharia de Software

### EXTREME PROGRAMMING

- **Planejamento**

- Histórias (Prioridades): todas desenvolvidas ou alta prioridade antecipada ou alta complexidade antecipada.

- **Projeto**

- Determina classes relevantes
- Problema - > Protótipo
- Refabricação (Otimização interna)

- **Codificação**

- Antes testes unitários (Cria protótipos)
- Codificação em Pares
- Integração (Desenvolvedores ou equipe específica)
- Teste (Constantes; Pequenos erros X Grandes erros)



# 2º Bloco

## Engenharia de Software

### Método de Desenvolvimento Dinâmico de Sistemas

- **Protótipos** (Ambiente controlado)
  - Entrega 80% em 20% do total
- **Ciclo de vida**
  - Viabilidade (Requisito; Restrições; Utilizar ou não o método)
  - Negócio (Requisitos funcionais; Valor de negócio; Arquitetura; manutenção)
- **Ciclo Iterativo**
  - Funcional (Protótipo demonstrativo)
  - Projeto e construção (Rever Ciclo Funcional; Gera valor de negócio ao usuário)
  - Implementação (Protótipo operativo; Completo ou não; Modificações futuras)

# 2º Bloco

Engenharia de Software

## MANIFESTO DESENVOLVIMENTO ÁGIL

1. Garantir a satisfação do consumidor entregando rapidamente e continuamente softwares funcionais;
2. Softwares funcionais são entregues frequentemente (semanas, ao invés de meses);
3. Softwares funcionais são a principal medida de progresso do projeto;
4. Até mesmo mudanças tardias de escopo no projeto são bem-vindas;
5. Cooperação constante entre pessoas que entendem do 'negócio' e desenvolvedores;



# 2º Bloco

Engenharia de Software

## MANIFESTO DESENVOLVIMENTO ÁGIL

6. Projetos surgem através de indivíduos motivados, e que deve existir uma relação de confiança;
7. Design do software deve prezar pela excelência técnica;
8. Simplicidade;
9. Rápida adaptação às mudanças;
10. Indivíduos e interações mais do que processos e ferramentas;
11. *Software* funcional mais do que documentação extensa;
12. Colaboração com clientes mais do que negociação de contratos;
13. Responder a mudanças mais do que seguir um plano.

# 2º Bloco

## Engenharia de Software

### SCRUM

O scrum tem esse nome devido a uma jogada do rugby, uma jogada onde todos vão ao encontro da bola. Esse modelo ágil de processo foi desenvolvido por Jeff Sutherland e por sua equipe no início da década de 1990. Como todos os modelos ágeis ele é consistente com o manifesto ágil.

- Pequenas equipes de trabalho são organizadas de modo a maximizar a comunicação, minimizar a supervisão e maximizar o compartilhamento de conhecimento.



# 2º Bloco

## Engenharia de Software

- O processo precisa ser adaptável tanto à modificações técnicas quanto a modificações de negócios, para garantir o melhor produto possível.
- O trabalho de desenvolvimento e o pessoal é dividido em partições claras.
- Testes e documentações constantes são realizados a medida que o produto é construído.
- O processo scrum fornece a habilidade de declarar o produto pronto, quando for necessário.



# 2º Bloco

## Engenharia de Software

### SCRUM





# 2º Bloco

## Engenharia de Software

### SCRUM

O scrum enfatiza um conjunto de padrões de processos de software que se mostram efetivos.

- **Pendência:** Uma lista priorizada de requisitos ou características do projeto que fornecem valor de negócio para o cliente. (backlog).
- **Sprints:** Período de tempo de tipicamente trinta dias para que todas as pendências setadas para aquela sprint sejam prontas.
- **Reunião Scrum:** Reuniões curtas de apenas quinze minutos, onde o scrum master (líder da equipe) faz três perguntas.

# 2º Bloco

## Engenharia de Software

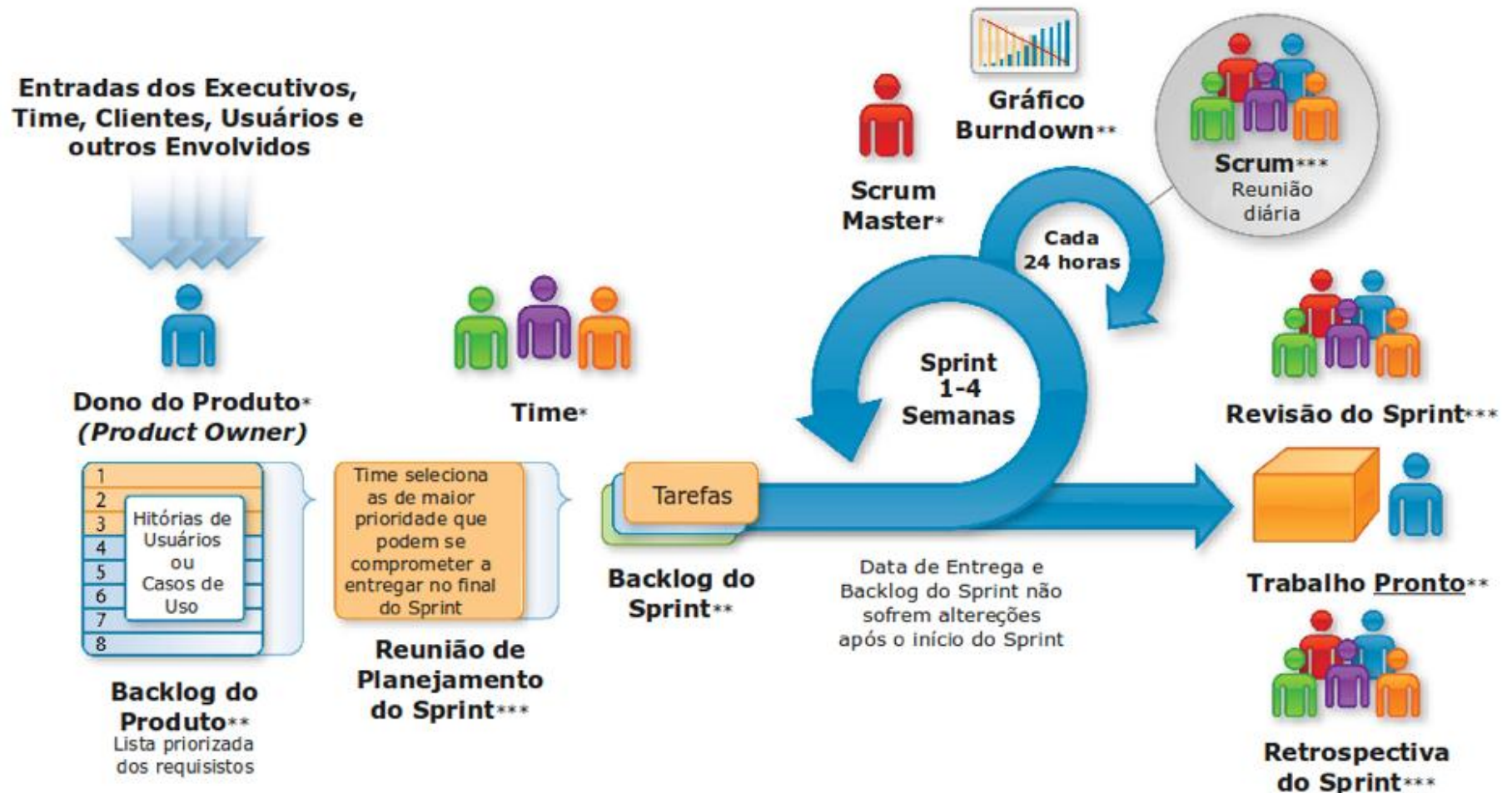
### SCRUM

1. O que você fez desde a última reunião de equipe?
  2. Que obstáculos você está encontrando?
  3. O que você planeja realizar até a próxima reunião de equipe?
- **Demos:** Entrega o implemento de software para o cliente de modo que a funcionalidades implementada possa ser demonstrada e avaliada pelo o cliente.



# 2º Bloco

## Engenharia de Software



# 2º Bloco

## Engenharia de Software

### CRYSTAL

A Família Crystal foi criada por Alistair Cockburn e Jim Highsmith, ela é caracterizada por ser um jogo cooperativo de invenção e comunicação de recursos limitados, com o principal objeto de entregar softwares úteis funcionando e com o objetivo secundário de preparar-se para o jogo seguinte. De acordo com a necessidade do projeto pode-se adotar algum tipo de método ágil:

- **Crystal Clear**
- **Crystal Yellow**
- **Crystal Orange**
- **and others**



# 2º Bloco

## Engenharia de Software

## FDD (Feature Driven Development) – Desenvolvimento Guiado por Características

- Originalmente criado por Peter Coad e seus colegas como um modelo prático para a engenharia de software orientada a objeto. Stephen Palmer e John Felsing melhoraram o FDD descrevendo o como um processo ágil e adaptativo que pode ser utilizado para projetos de softwares medianos e grandes.
- No contexto do FDD, uma característica é uma função valorizada pelo o cliente que pode ser implementada em duas semanas ou menos. A ênfase na definição das características fornecem os seguintes benefícios:
- Como pequenos blocos de funcionalidades, elas fazem com que o usuário possa descreve-las com maior facilidade e suas relações umas com as outras, permitindo também uma melhor revisão.

# 2º Bloco

## Engenharia de Software

- Podem ser agrupadas hierarquicamente de acordo com o negócio.
- Como características são um incremento de software passível de entrega do FDD, a equipe desenvolve características operacionais a cada duas semanas.
- Como características são pequenas, sua representação de projeto e de código são mais fáceis de inspecionar.
- Planejamento de projeto, cronogramação e monitoramento são guiados pela hierarquia.



# 2º Bloco

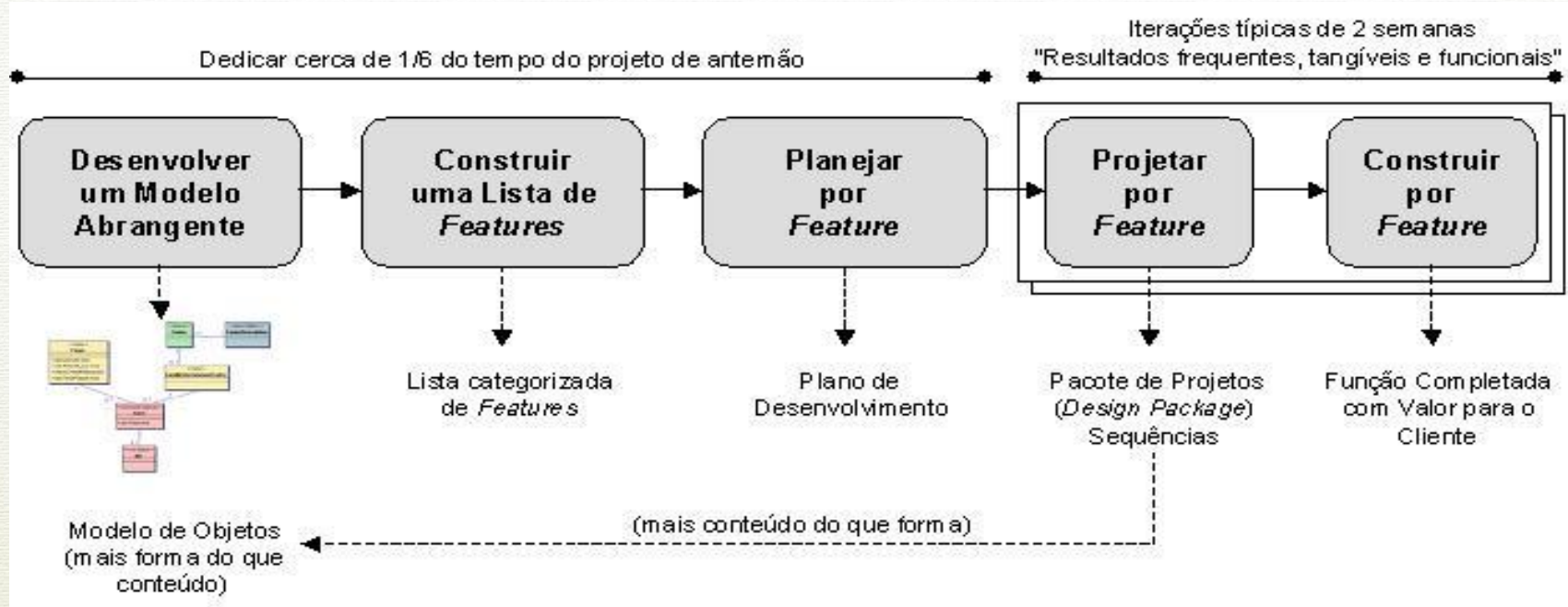
## Engenharia de Software

### FDD (Feature Driven Development) - Desenvolvimento Guiado por Características

Coad define o gabarito para uma característica como:

<ação> o <resultado> <por | para | de | a> um <objeto>.

Ex: Adicionar um produto a um carrinho de compras.



# 2º Bloco

Engenharia de Software

## Modelagem Ágil (Agile Modeling - AM)

Há situações em que os engenheiros necessitam construir um projeto grande. Para facilitar o entendimento desse projeto é feito uma modelagem do escopo e complexibilidade, tais que todos os participantes devem entender melhor o que será realizado, o problema deve ser particionado entre o pessoal que irá resolvê-lo e que a qualidade possa ser avaliada a cada passo. O que caracteriza o AM é:



# 2º Bloco

## Engenharia de Software

## Modelagem Ágil (Agile Modeling - AM)

- **Modelar Com Finalidade:** O desenvolvedor que usa AM deve ter uma meta definida como por ex: Ajudar o cliente a entender algum aspecto do software, antes de criar o modelo.
- **Usar Modelos Multiplos:** A AM sugere que, para fornecer a visão necessária, cada modelo apresente um aspecto diferente do sistema, e que somente modelos que ofereçam valores altos sejam usados.
- **Viajar Leve:** A medida em que o trabalho de engenharia de software prossegue, conserve apenas aqueles modelos de valores de longo prazo, o resto você se livra.

# 2º Bloco

## Engenharia de Software

### Modelagem Ágil (Agile Modeling - AM)

- O conteúdo é mais importante que a apresentação, pois um modelo que apresenta notação defeituosa, mas que fornece um conteúdo valioso é muito mais vantajoso.
- Conhecer os modelos e as ferramentas que você usa para criá-los, pois entendendo seus pontos fortes e fracos vai te ajudar no processo.
- Adaptar Localmente, significa que a abordagem de modelagem deve adaptar-se as necessidades da equipe ágil.



# 2º Bloco

Engenharia de Software

## FONTE:

[http://www.balagan.org.uk/work/agile\\_comparison.htm](http://www.balagan.org.uk/work/agile_comparison.htm)

<http://www.versionone.com/Agile101/Agile-Development-Methodologies-Scrum-Kanban-Lean-XP/wikipedia>