

Metodologias Ágeis

Mauro Henrique Lima de Boni

Sumário

- Entendendo o processo de desenvolvimento de *software*
- Por que os softwares falham ?
- Porque os softwares devem mudar ?
- Metodologias Ágeis
- SCRUM

Entendendo o processo de desenvolvimento de *software*

Atividades básicas

- Especificação
- Projeto e Implementação
- Testes
- Evolução

Principais Processos

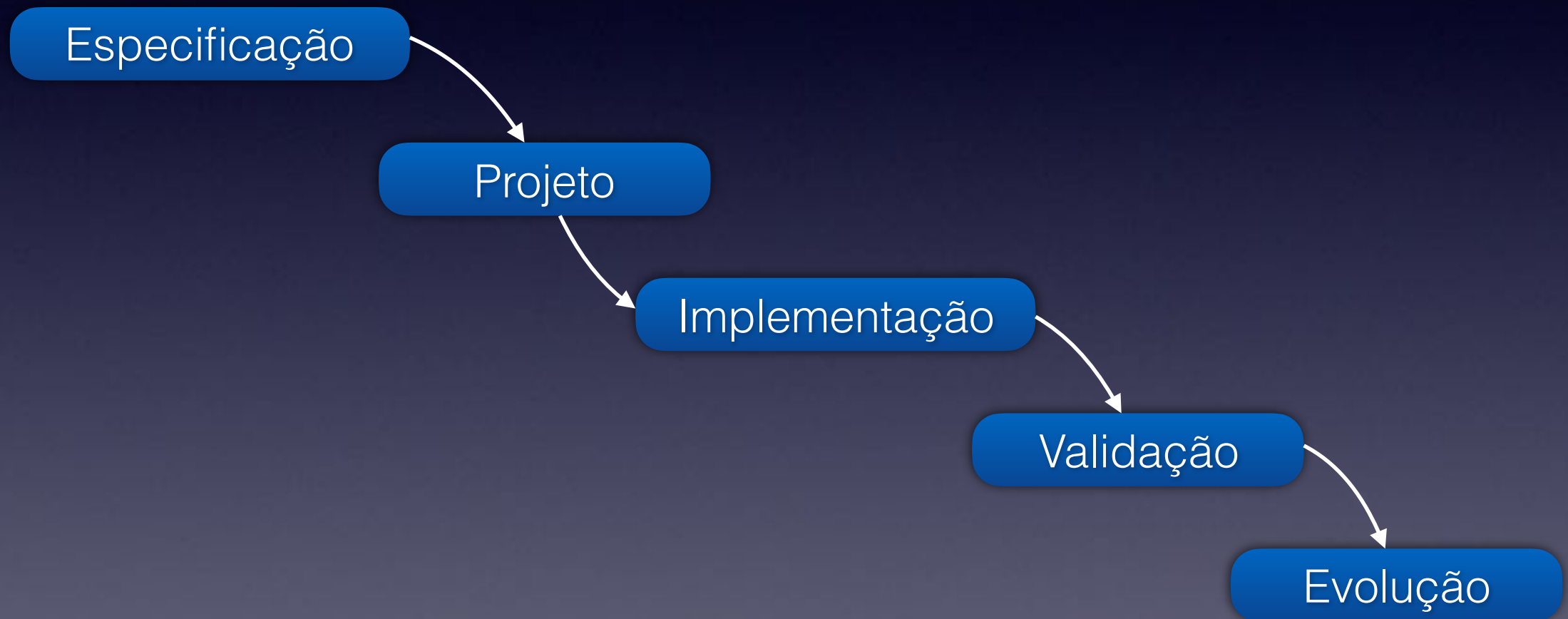
Modelo cascata (*waterfall model*)

- Modelo dirigido a planos. Possui fases separadas de especificação e implementação.

Desenvolvimento incremental

- Especificação, implementação e validação são intercaladas. Pode ser tanto dirigida a planos quanto ágil.

Modelo Cascata

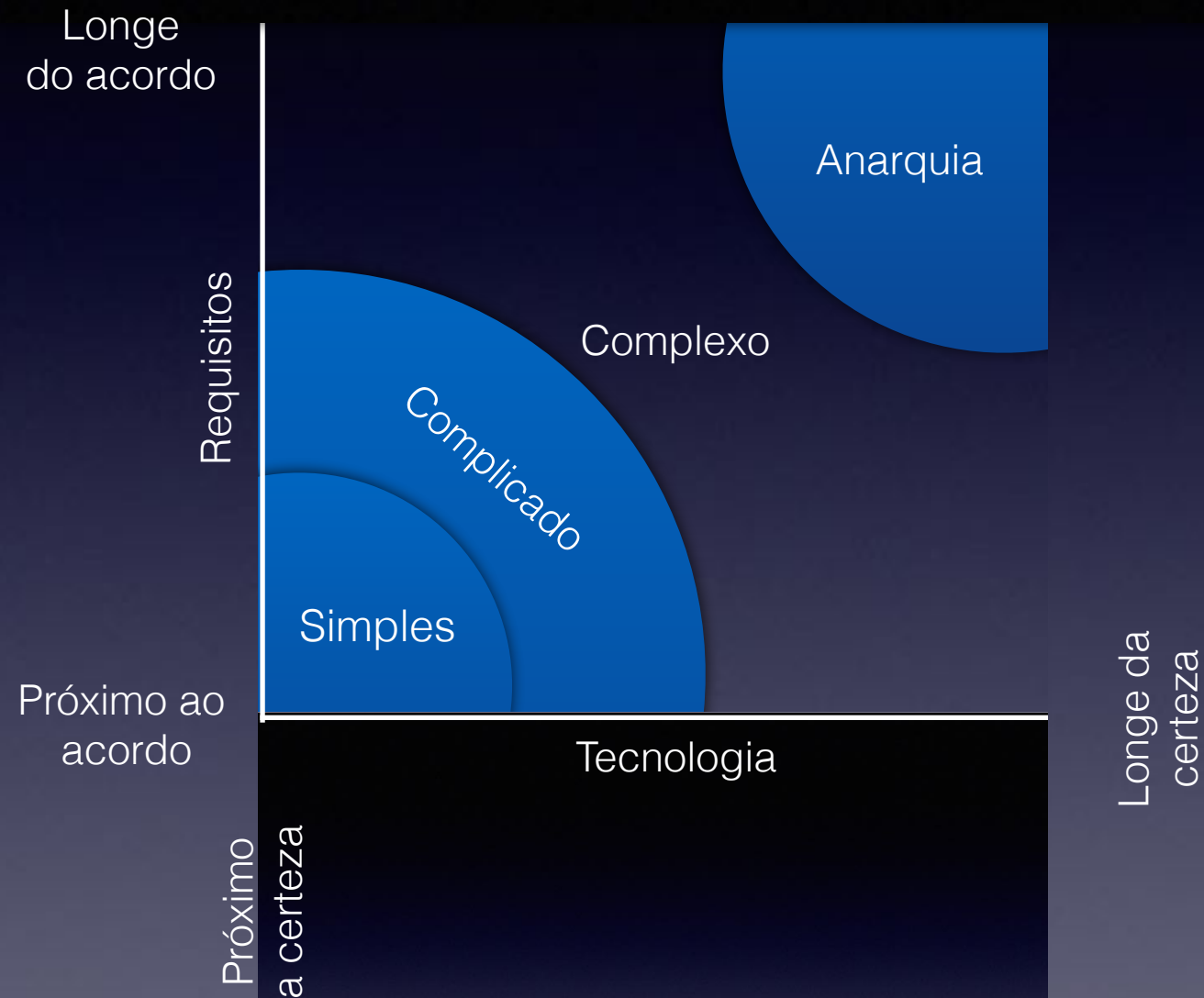


Problemas



Por que os softwares
falham ?

Por que os softwares falham ?



Fonte: Strategic Management and Organizational Dynamics by Ralph Stacey in Agile Software Development with Scrum by Ken Schwaber and Mike Beedle.

Por que os softwares falham ?

O caso do foguete ARIANE 5

- OS sistemas referenciais inerciais (SRI), deixaram de funcionar por volta de 36,7 segundos após a decolagem.
- A anomalia interna de software do SRI ocorreu durante a execução de uma conversão de dados de um número de 64 bits em ponto flutuante para um inteiro de 16 bits com sinal. O valor do número em ponto flutuante era maior do que poderia ser representado pelo inteiro de 16 bits com sinal.



Por que os softwares
sofrem mudanças ?

Por que os softwares sofrem mudanças ?

- Mudanças nos negócios geram novos requisitos para o sistema.
- Novas tecnologias permitem maneiras de melhorar a implementação.
- Mudanças nas plataformas fazem com que as aplicações mudem.
- Mudanças levam ao retrabalho, então os custos das mudanças incluem os valores desse retrabalho e também aqueles relacionados com a implementação de novas funcionalidades.

METODOLOGIAS ÁGEIS

Manifesto ágil

Indivíduos e interações

Software que funciona

Colaboração do cliente

Resposta à mudanças

ao
invés
de

Processos e ferramentas

Documentação
abrangente

Negociação de contrato

Seguir um plano

XP

DSM

LEAN

EDD

FDD

TDD

ATDD

SCRUM

CRYSTAL

DDD

MDE

SCRUM

SCRUM



SCRUM

- Scrum é um processo ágil que permite manter o foco na entrega do maior valor de negócio, no menor tempo possível.
- Isto permite a rápida e contínua inspeção do software em produção (em intervalos de duas a quatro semanas).
- As necessidades do negócio é que determinam as prioridades do desenvolvimento de um sistema. As equipes se auto-organizam para definir a melhor maneira de entregar as funcionalidades de maior prioridade.
- Entre cada duas a quatro semanas todos podem ver o real software em produção, decidindo se o mesmo deve ser liberado ou continuar a ser aprimorado por mais um “Sprint”.

SCRUM

- Microsoft
- Yahoo
- Google
- Electronic Arts
- IBM
- Lockheed Martin
- Philips
- Siemens
- Nokia
- Capital One
- BBC
- Intuit
- Nielsen Media
- First American Real Estate
- BMC Software
- Ipswitch
- John Deere
- Lexis Nexis
- Sabre
- Salesforce.com
- Time Warner
- Turner Broadcasting
- Océ
- gobo.com

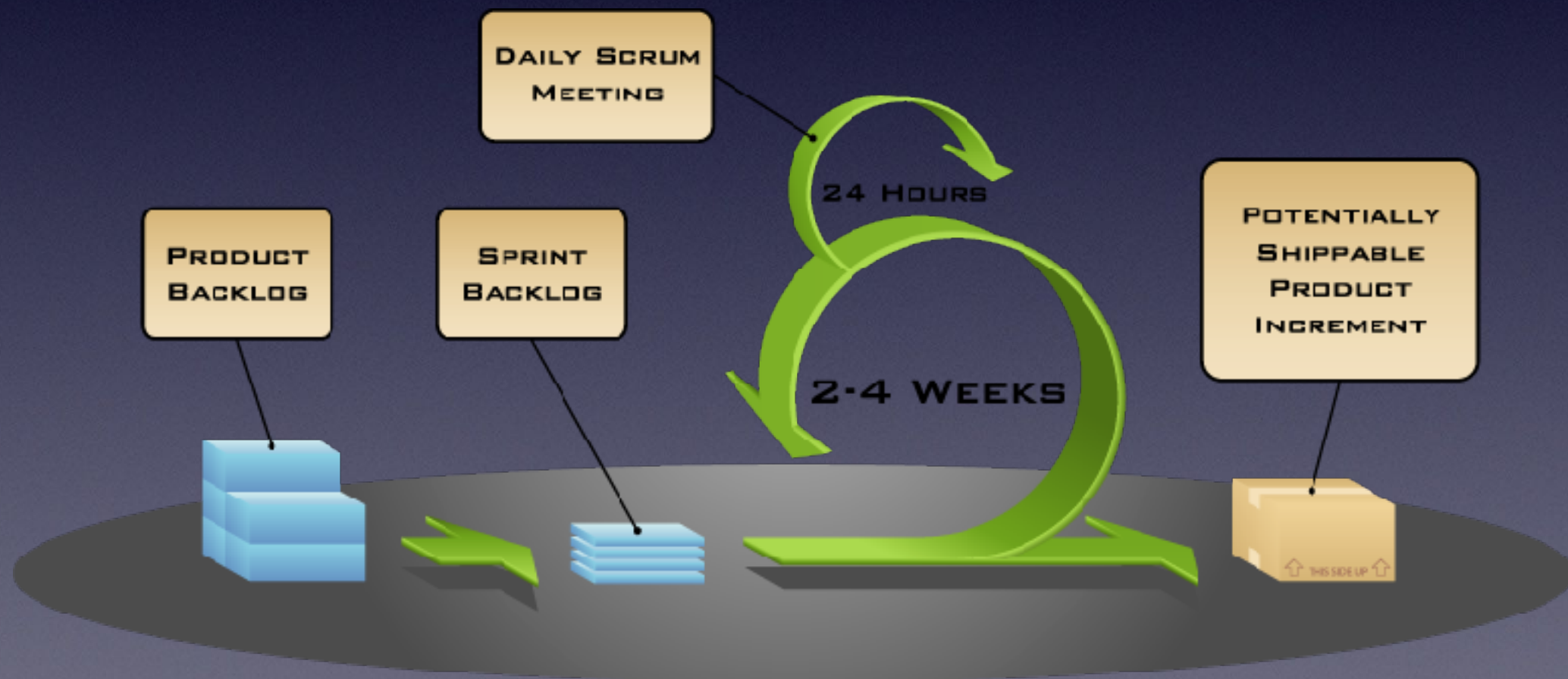
SCRUM

- Software comercial
- Desenvolvimento interno
- Desenvolvimento contratado (terceirização)
- Projetos de preço fixo
- Aplicações Financeiras
- Aplicações certificadas pela ISO 9001
- Sistemas embarcados
- Sistemas disponíveis 24x7
- Desenvolvimento por hackers solitários
- Video games
- Sistemas para suporte à vida
- Sistemas para controle de satélites
- Websites
- Software para handhelds
- Telefones celulares
- Aplicações para redes
- Aplicações de ISV (Independent Software Vendors)

SCRUM

- Equipes que se auto-organizam
- O produto evolui em uma série de “Sprints” mensais
- Os requerimentos são listados em um “Product Backlog”
- Não há prática de engenharia prescrita (o Scrum adequa-se a todas)
- Usa regras generativas na criação de um ambiente ágil para a entrega de projetos

SCRUM



SCRUM

- Projetos Scrum progridem em uma série de “*sprints*”
- Ocorre em um período de duas a quatro semanas
- Um período constante leva a um melhor “ritmo”
- O produto é projetado, codificado e testado durante o sprint

SCRUM

Especificação

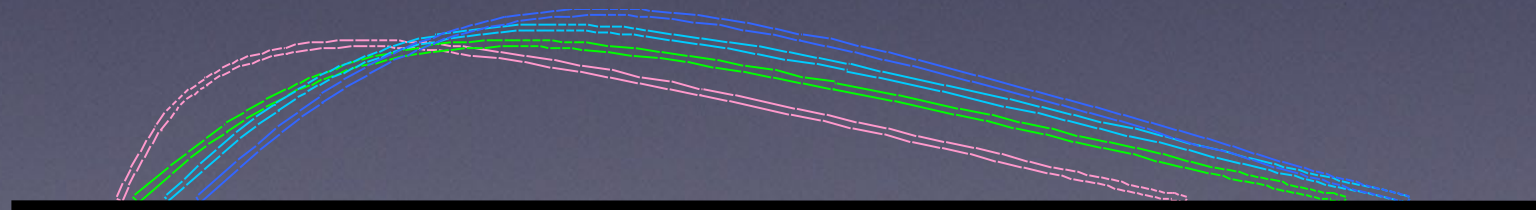
Projeto

Código

Teste

Ao invés de completar uma coisa por vez...

... equipes Scrum fazem um pouco de cada coisa, todo o tempo.



SCRUM

Papéis

- Dono do produto
- ScrumMaster
- Equipe

Cerimônias

- Planejamento
- Revisão
- Retrospectiva
- Reunião diária

Artefatos

- Product backlog*
- Sprint backlog*
- Burndown charts*

Papéis do SCRUM

Dono do produto

- Define as funcionalidades do produto
- Decide datas de lançamento e conteúdo
- Responsável pela rentabilidade (ROI)
- Prioriza funcionalidades de acordo com o valor de mercado
- Ajusta funcionalidades e prioridades
- Aceita ou rejeita o resultado dos trabalhos

Scrum Master

- Representa a gerência para o projeto
- Responsável pela aplicação dos valores e práticas do Scrum
- Remove obstáculos
- Garante a plena funcionalidade e produtividade da equipe
- Garante a colaboração entre os diversos papéis e funções
- Escudo para interferências externas

Equipe

- Não existe uma divisão funcional através de papéis tradicionais, tais como programador, designer, analista de testes ou arquiteto.
- Todos trabalham juntos para completar o conjunto de trabalho com o qual se comprometeram conjuntamente para um Sprint.

Cerimônias do SCRUM

Planejamento do *Sprint*

- A equipe seleciona itens do *Product Backlog* com os quais compromete-se a concluir
- O *Sprint Backlog* é criado
- Tarefas identificadas e estimadas (1 a 16 horas)
- De forma colaborativa, não apenas feito pelo *ScrumMaster*

Reunião diária

- Duração
 - 15 minutos
- Todos em pé!
- Não é para a solução de problemas
 - Todo mundo é convidado
 - Apenas os membros da equipe, ScrumMaster, dono do produto podem falar
- Ajuda a evitar reuniões adicionais desnecessárias



Revisão do Sprint

- A equipe apresenta os resultados obtidos durante o Sprint
- Tipicamente, demonstração de novas funcionalidades ou sua arquitetura
- Informal
 - 2 horas de preparação
 - Sem slides
- Todo o time participa
- O mundo é convidado

Retrospectiva

- Periodicamente, observe o que funciona e o que não funciona
- Tipicamente de 15 a 30 minutos
- Feita após cada Sprint
- Toda a equipe participa
 - ScrumMaster
 - Dono do produto
 - Membros da equipe
 - Clientes e outros

Inicia, Pára, Continua

A equipe discute o que gostaria de:

Iniciar a fazer

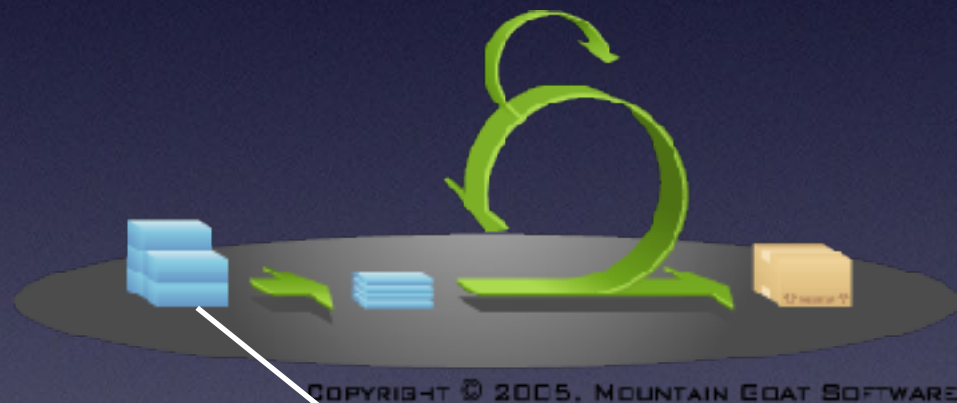
Parar de fazer

Continuar fazendo

Esta é uma das
várias maneiras de
se conduzir uma
retrospectiva do
Sprint

Artefatos

Product Backlog



Este é o Product Backlog

- Contem os requisitos
- Uma lista de todo o trabalho desejado no projeto
- Idealmente, na forma em que cada item tenha seu peso de acordo com a vontade do cliente ou usuários
- Priorizado pelo dono do produto
- Repriorizado no início de cada Sprint

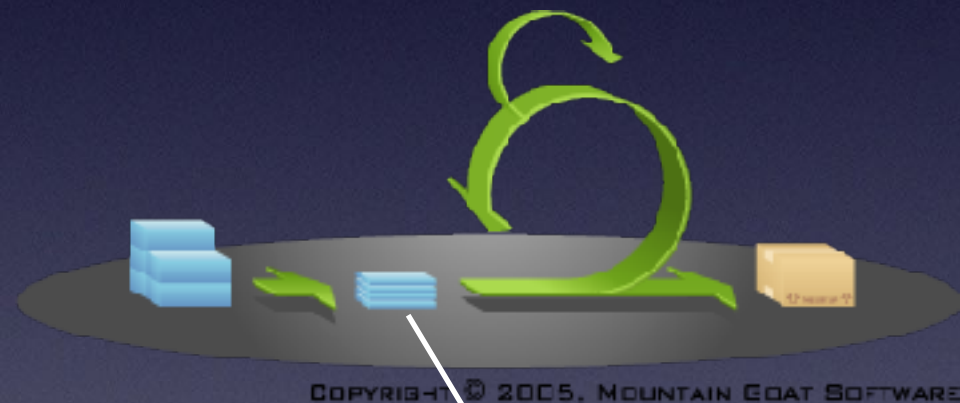
Exemplo

Item do Backlog	Estimativa
Permitir que o usuário faça uma reserva	3
Permitir que o usuário cancele a reserva	5
Permitir a troca de datas da reserva	3
Permitir que empregados do hotel gerem relatórios de lucratividade	8
Melhorar manipulação de erros	8
...	30
...	50

User stories / Estórias do usuário

- User Stories, são similares aos casos de uso, mas focam nos objetivos do usuário e como o sistema alcança esses objetivos.
- Como um <papel/ator>, eu quero/desejo <ação> para <funcionalidade>
- Como um cliente eu quero ver os filmes disponíveis no catalogo para reservar um aluguel para a data X .

Sprint Backlog



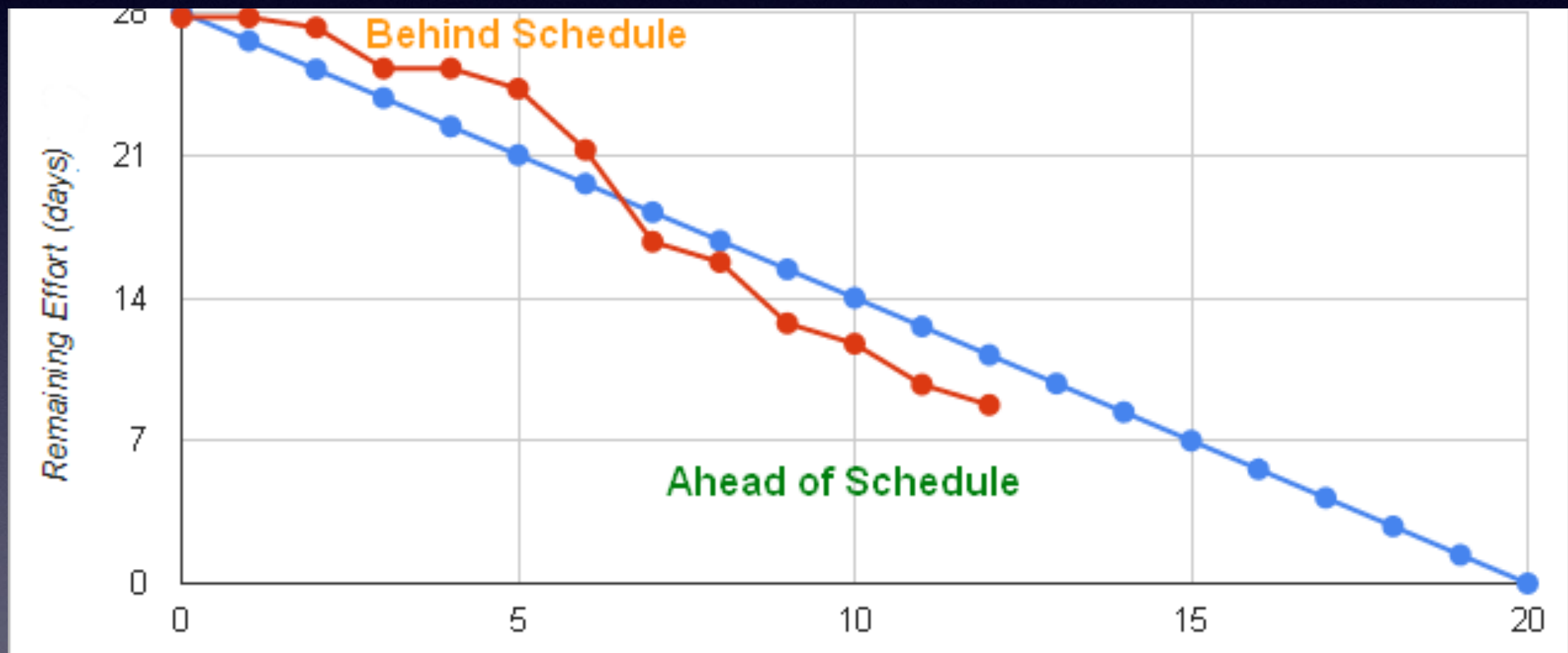
Este é o Sprint Backlog

- Cada indivíduo escolhe o trabalho que fará
- Trabalhos nunca são atribuídos
- Atualização diária da estimativa do trabalho restante
- Qualquer membro da equipe pode adicionar, apagar ou mudar tarefas
- O trabalho aparece a partir do Sprint
- Se uma tarefa não é clara, defina-a como um item com uma quantidade maior de tempo e subdivida-a depois
- Atualize as coisas a serem feitas na medida em que se tornam mais conhecidas

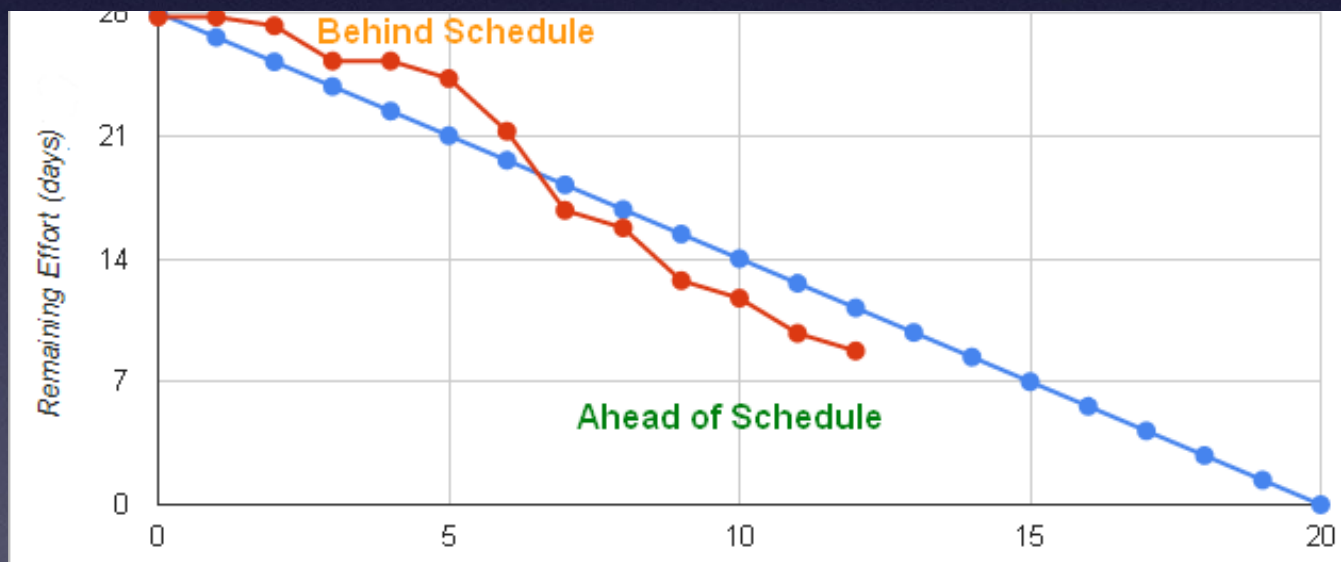
Sprint backlog

Tarefas	Seg	Ter	Qua	Qui	Sex
Codificar interface de usuário	8	4	8		
Codificar regra de negócio	16	12	10	4	
Testar	8	16	16	11	8
Escrever help online	12				
Escrever a classe foo	8	8	8	8	8
Adicionar log de erros			8	4	

Burndown Charts



Burndown Charts



- Representa diariamente o progresso do trabalho em desenvolvimento. Ou seja, após cada dia de trabalho o gráfico apresenta a porção de trabalho finalizada em comparação com o trabalho total planejado.
- O gráfico mostra a quantidade de trabalho que falta ser feito no eixo vertical (y) versus o tempo no eixo horizontal (x).