

SCRUM

Dr. rer. nat. Christiane Gresse von Wangenheim, PMP



Objetivo de aprendizagem desta aula

Ao final desta aula, você deverá ser capaz de:

- ☐ Conhecer um método ágil para gerência de projetos.
- ☐ Explicar o processo de gerência de projetos seguindo SCRUM.
- ☐ Aplicar o SCRUM.



Manifesto ágil

Indivíduos e interações	acima de	Processos e ferramentas
Software funcionando	acima de	Documentação abrangente
Colaboração com o cliente	acima de	Negociação de contratos
Responder a mudanças	acima de	Seguir um plano

SCRUM no Rugby

Um Scrum (jogadores amontoados) é um time em Rugby. Todos no time agem em conjunto para mover a bola para baixo no campo.



Gerência de projetos: SCRUM

- ❑ É um *framework* para gerência de projeto.
- ❑ SCRUM concentra-se na organização em:
 - ❑ construção de produtos de sucesso.
 - ❑ entrega de recursos úteis em intervalos regulares.
 - ❑ esperando requisitos, arquitetura e alterações no projeto.



SCRUM

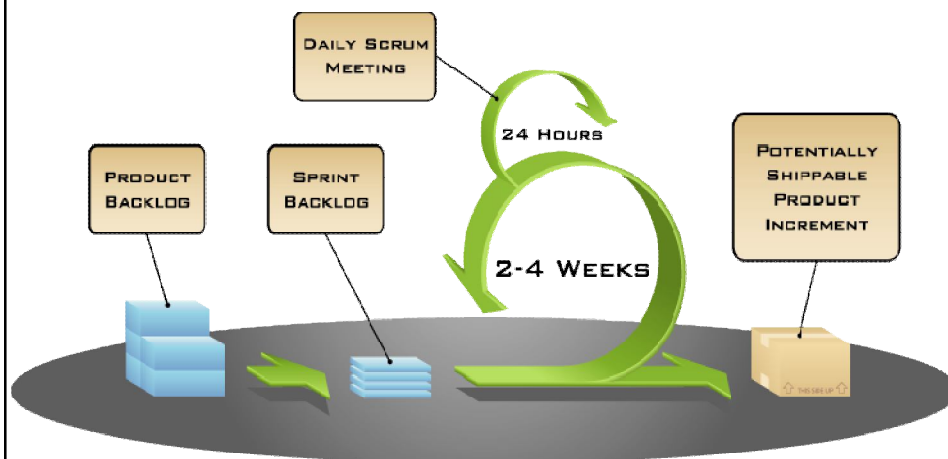
Scrum é um **método de gerência de projetos de software** que pode ser aplicado a outros métodos ágeis como XP e que define como **controlamos um projeto**, NÃO como o produto é desenvolvido.

David James Anderson

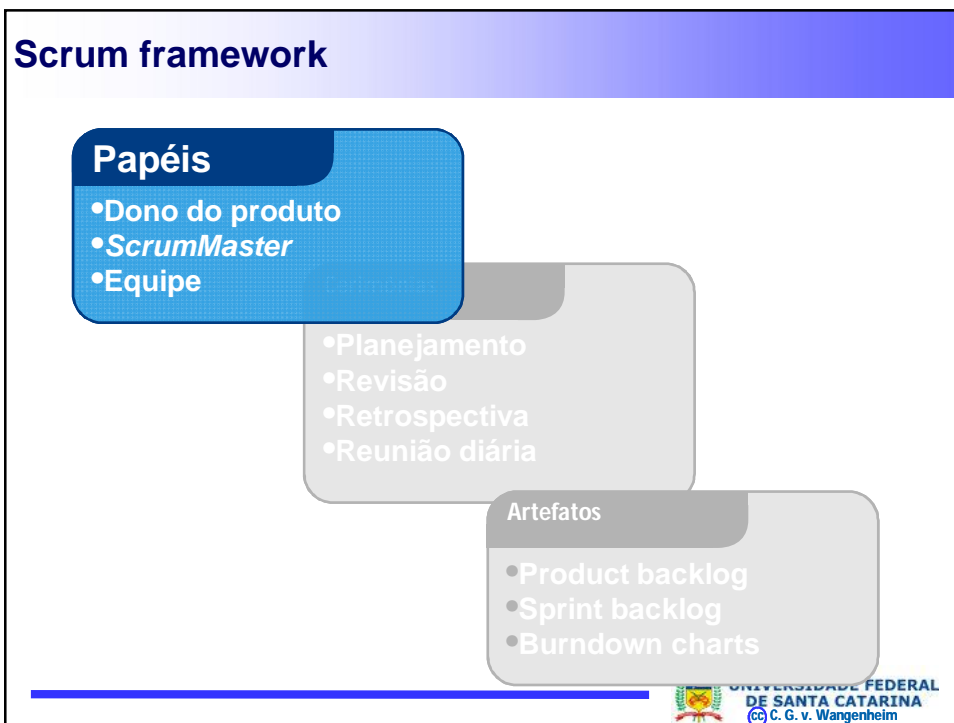
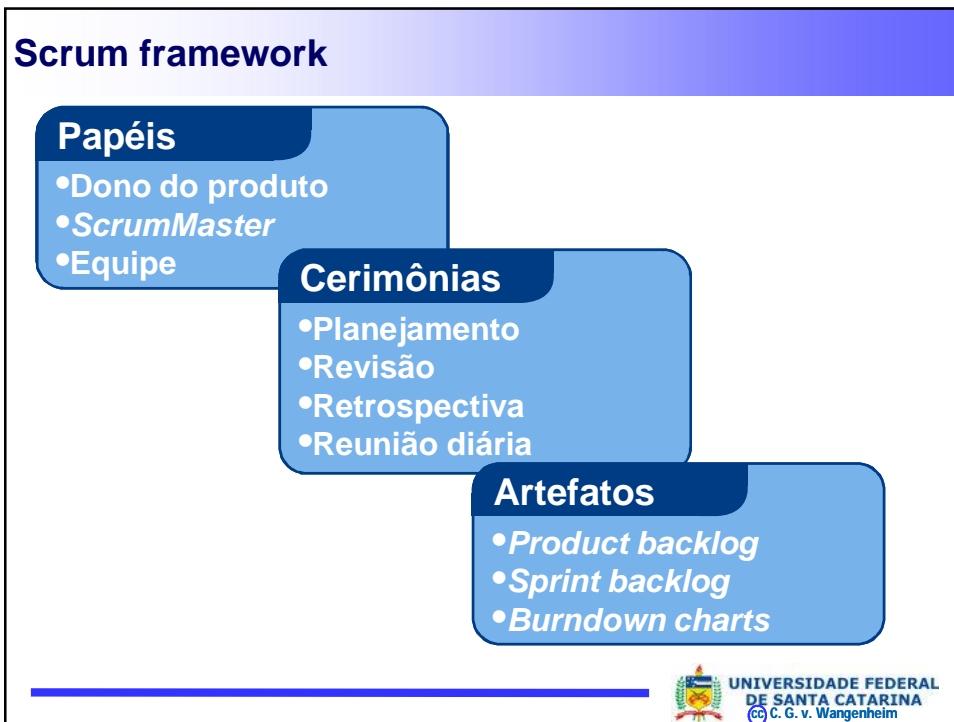
Características

- ❑ Um dos “métodos ágeis”.
- ❑ Equipes pequenas (<10 pessoas).
- ❑ O produto evolui em uma série de *sprints* com duração de 2 a 4 semanas.
- ❑ Gerando incrementos visíveis e úteis.
- ❑ Requisitos são capturados como itens de uma lista de *product backlog*.
- ❑ Não define práticas de desenvolvimento de software.

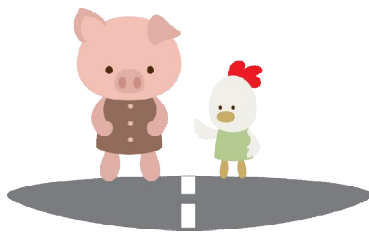
Visão geral



COPYRIGHT © 2005, MOUNTAIN G3AT SOFTWARE



Uma galinha e um porco estão....



UNIVERSIDADE FEDERAL
DE SANTA CATARINA
C. G. v. Wangenheim

A história ...



By Clark & Vizdos

© 2006 implementingscrum.com



UNIVERSIDADE FEDERAL
DE SANTA CATARINA
C. G. v. Wangenheim

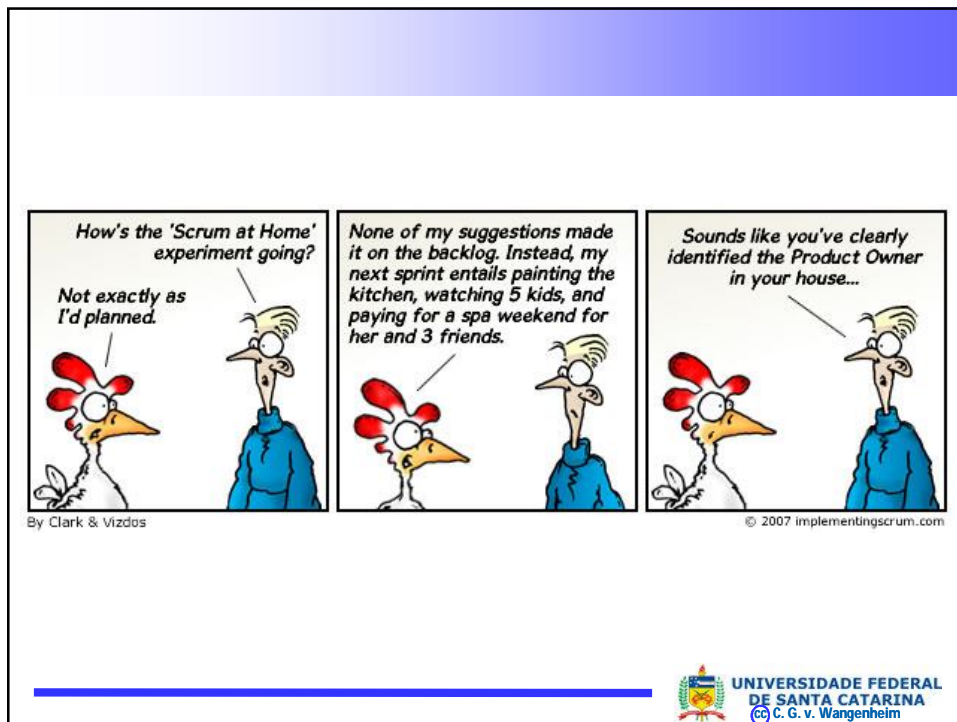
E no SCRUM ...

- ☐ Quem são os porcos?
 - ☐ Dono do produto (*Product owner*)
 - ☐ *Scrum Master*
 - ☐ Equipe
- ☐ Quem são as galinhas?
 - ☐ Clientes
 - ☐ Vendedores
 - ☐ Gerentes sênior

Dono do produto (*Product owner*)

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





Scrum master



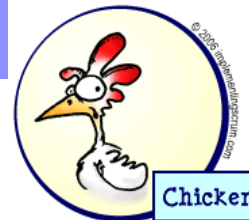
- ❑ Representa a gerência para o projeto.
- ❑ Responsável pela aplicação dos valores e práticas do Scrum.
- ❑ Remove obstáculos.
- ❑ Garante a plena funcionalidade e produtividade da equipe.
- ❑ Garante a colaboração entre os diversos papéis e funções.
- ❑ Escudo para interferências externas.

Equipe



- ☐ Entre 5 e 9 pessoas.
- ☐ Multi-funcional
 - ☐ Programadores, testadores, desenvolvedores de interfaces, etc.
- ☐ Tempo integral
 - ☐ Raras exceções (Ex.: Administrador de banco de dados)
- ☐ Auto-organizável
 - ☐ Idealmente, sem títulos
- ☐ Trocas só na mudança de *sprints*.

Outros *stakeholders*



- ☐ Papéis de galinhas não fazem parte do processo Scrum, mas devem ser considerados.
- ☐ *Stakeholders* (clientes, fornecedores):
 - ☐ para quem o software está sendo construído.
 - ☐ financiam o projeto.
 - ☐ estão envolvidos diretamente somente durante as revisões de *sprints*.
- ☐ Gerência sênior: providencia o ambiente para as organizações de desenvolvimento do produto.

Scrum framework

Papéis

- Dono do produto
- ScrumMaster
- Equipe

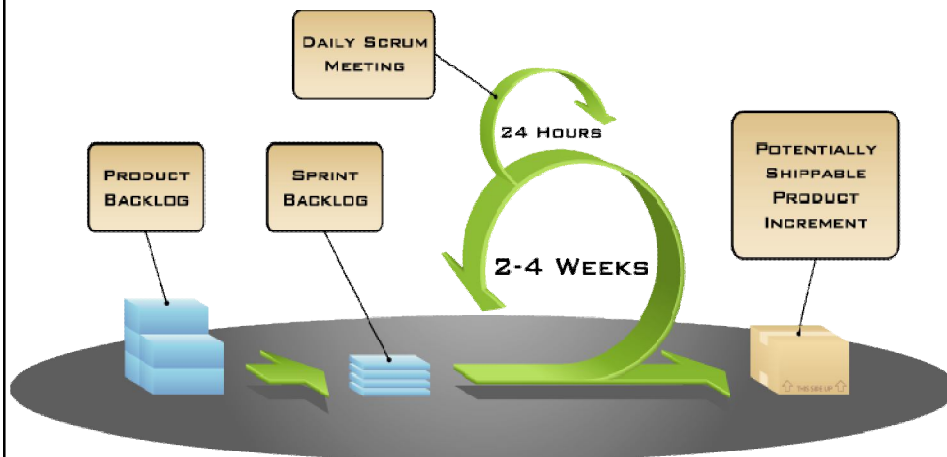
Cerimônias

- Planejamento
- Revisão
- Retrospectiva
- Reunião diária

Artefatos

- Product backlog
- Sprint backlog
- Burndown charts

Visão geral



COPYRIGHT © 2005, MOUNTAIN G3AT SOFTWARE

Product backlog

- ❑ Lista de todas as funcionalidade desejada no produto.
- ❑ Representa os requisitos alto nível (*features*) do projeto.
 - ❑ Podem ser representadas em várias formas: *features*, histórias de usuário, ...
- ❑ Responsável: *product owner*
 - ❑ Qualquer um pode contribuir itens para *product backlog*.
 - ❑ Controlada pelo *product owner*.
- ❑ Lista priorizada de trabalho a ser executado em um produto.
 - ❑ Priorizado pelo *product owner*.
 - ❑ Re-priorizada no início de cada *sprint*.

“Biggest risk in a project is building the wrong product.”

Mike Cohn



Product backlog

- ❑ Consiste em:
 - ❑ Requisitos
 - ❑ Prioridade de cada requisito
 - ❑ Estimativa de esforço (tamanho - não duração) de cada requisito

Estórias de usuário (user stories)

Estória de usuários é uma definição de alto nível de requisitos, contendo apenas informação suficiente para que os desenvolvedores possam produzir uma estimativa razoável do esforço para implementá-la.

- ❑ Histórias devem ser suficientemente pequenas para serem feitas por 2-3 membros da equipe em 3-4 dias.

Como um planejador de férias, eu quero ver fotos dos hotéis para poder ter uma ideia melhor das instalações.

Exemplo: *product backlog*

Requirement / User Story
As a Clerk, I can reserve a room for a guest.
As a Clerk, I can generate a bill for the guest's stay.
As a Manager, I can make unlimited adjustments to a customer's bill.
As a Clerk, I can cancel a reservation.
As a Clerk, I can change a reservation.
As a Guest, I can make a reservation.
As a Clerk, I can check-in a guest.
As a Clerk, I can check-out a guest.
As a Guest, I can assign a charge to my room.
As a Clerk, I can adjust a customer's bill.
As a Clerk, I can enter a work order for a customer's room.
As a Housekeeper, I can obtain a list of rooms to service.
As a Clerk, I can move a guest to a different room.
As a Guest, I can purchase an in-room movie.
As a Guest, I can request a copy of my current charges.
As a Manager, I can schedule staff work-hours and work-days.
As a Housekeeper, I can inform the system of problems found in a room.



UNIVERSIDADE FEDERAL
DE SANTA CATARINA
C. G. v. Wangenheim

Exercício



- ☐ Exercício em pequenos grupos.
- ☐ Instruções
 - ☐ Voltando ao projeto da pizzaria. Mas agora, vocês gerenciarão o projeto de forma ágil.
 - ☐ Definem o *product backlog* para este projeto.

Exemplos:

As a Clerk, I can reserve a room for a guest.
As a Clerk, I can generate a bill for the guest's stay.
As a Manager, I can make unlimited adjustments to a customer's bill.

- ☐ Tempo: 10 minutos



UNIVERSIDADE FEDERAL
DE SANTA CATARINA
C. G. v. Wangenheim

Exercício



- ❑ O seu tio, que é dono de um pizzaria, ligou para você com uma proposta de projeto. Atualmente ele já oferece a entrega em domicilio via ligações telefônicas. Para ampliar o seu negocio, ele quer possibilitar que os seus clientes, via Internet, possam encomendar pizzas no site da sua pizzaria. Estas informações então serão processadas por seus dois atendentes, que precisarão ser treinados, visto que atualmente tem pouco conhecimento de TI. Além disso, ele também quer um módulo para dispositivos móveis deste sistema a ser acessado pelos iPhones dos entregadores, por qual eles possam verificar detalhes da entrega (endereço, valor total, etc.). O seu tio pretende lançar o sistema já na festa de pizza deste ano, no dia 1 de Julho. Ele conseguiu reservar \$ 15.000,00 para o projeto. Você falou com os seus dois melhores amigos no curso de computação e eles estão interessados em participar.



Priorizar os itens do *product backlog*

- ❑ A priorização de trabalho é essencial para o sucesso do projeto.
 - ❑ Priorizadas de acordo com o **valor de negócio** dado pelo cliente ou usuário.
 - ❑ Depende do critério pelo qual o projeto será avaliado como um sucesso ou um fracasso.
- ❑ Como fazer?
 - ❑ *Product owner* identifica o conjunto de atributos de itens de *backlog* a serem considerados para priorização.
 - ❑ ROI, tempo de retorno, necessidade de usuário para ter esta funcionalidade, etc.
 - ❑ Atribuir um valor a estes atributos.
 - ❑ Classificar a lista com base nos valores desses atributos.



Exemplo: product backlog

Product Backlog	
Priority	Requirement / User Story
Very High	As a Clerk, I can reserve a room for a guest.
	As a Clerk, I can generate a bill for the guest's stay.
	As a Manager, I can make unlimited adjustments to a customer's bill.
High	As a Clerk, I can cancel a reservation.
	As a Clerk, I can change a reservation.
	As a Guest, I can make a reservation.
	As a Clerk, I can check-in a guest.
	As a Clerk, I can check-out a guest.
	As a Guest, I can assign a charge to my room.
Medium	As a Clerk, I can adjust a customer's bill.
	As a Clerk, I can enter a work order for a customer's room.
	As a Housekeeper, I can obtain a list of rooms to service.
Low	As a Clerk, I can move a guest to a different room.
	As a Guest, I can purchase an in-room movie.
	As a Guest, I can request a copy of my current charges.
	As a Manager, I can schedule staff work-hours and work-days.
	As a Housekeeper, I can inform the system of problems found in a room.



UNIVERSIDADE FEDERAL
DE SANTA CATARINA
C. G. v. Wangenheim

Exercício



- ☐ Exercício em pequenos grupos.
- ☐ Instruções:
 - ☐ Priorizam os itens do *product backlog* para este projeto.

- ☐ Tempo: 5 minutos



UNIVERSIDADE FEDERAL
DE SANTA CATARINA
C. G. v. Wangenheim

Estimar o tamanho dos itens do *product backlog*

❑ Unidades de estimativas

- ❑ **Dias ideais:** dias de desenvolvimento ideal de esforço ininterruptos, concentrados na tarefa e planejamento produtivo.

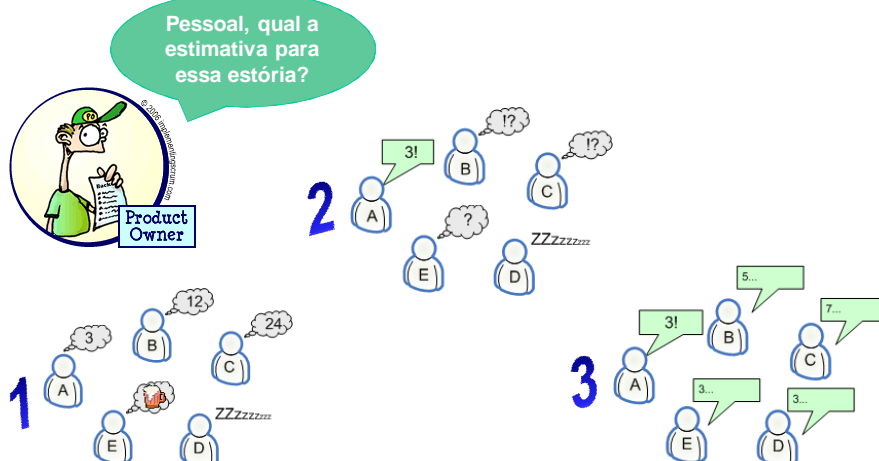
- ❑ Exemplo: tempo ideal de um filme geralmente é 120 minutos. Considerando os intervalos de propaganda o tempo total é muito maior.

- ❑ **Pontos de história de usuários:** estimativas **sem dimensões**, mas unidades **proporcionais**.

- ❑ Uma história de usuário de 10 pontos é esperado de levar duas vezes mais tempo do que uma história de usuário de 5 pontos.

❑ Usado no planejamento da *sprint* e releases.

Estimando o *product backlog* usando *planning poker*



Exemplo: *product backlog*

Product Backlog		
Priority	Requirement / User Story	Effort (in Story Points)
Very High	As a Clerk, I can reserve a room for a guest.	5
	As a Clerk, I can generate a bill for the guest's stay.	2
	As a Manager, I can make unlimited adjustments to a customer's bill.	8
High	As a Clerk, I can cancel a reservation.	2
	As a Clerk, I can change a reservation.	5
	As a Guest, I can make a reservation.	5
	As a Clerk, I can check-in a guest.	5
	As a Clerk, I can check-out a guest.	3
	As a Guest, I can assign a charge to my room.	3
Medium	As a Clerk, I can adjust a customer's bill.	3
	As a Clerk, I can enter a work order for a customer's room.	3
	As a Housekeeper, I can obtain a list of rooms to service.	2
Low	As a Clerk, I can move a guest to a different room.	3
	As a Guest, I can purchase an in-room movie.	2
	As a Guest, I can request a copy of my current charges.	3
	As a Manager, I can schedule staff work-hours and work-days.	5
	As a Housekeeper, I can inform the system of problems found in a room.	3



UNIVERSIDADE FEDERAL
DE SANTA CATARINA
C. G. v. Wangenheim

Exercício



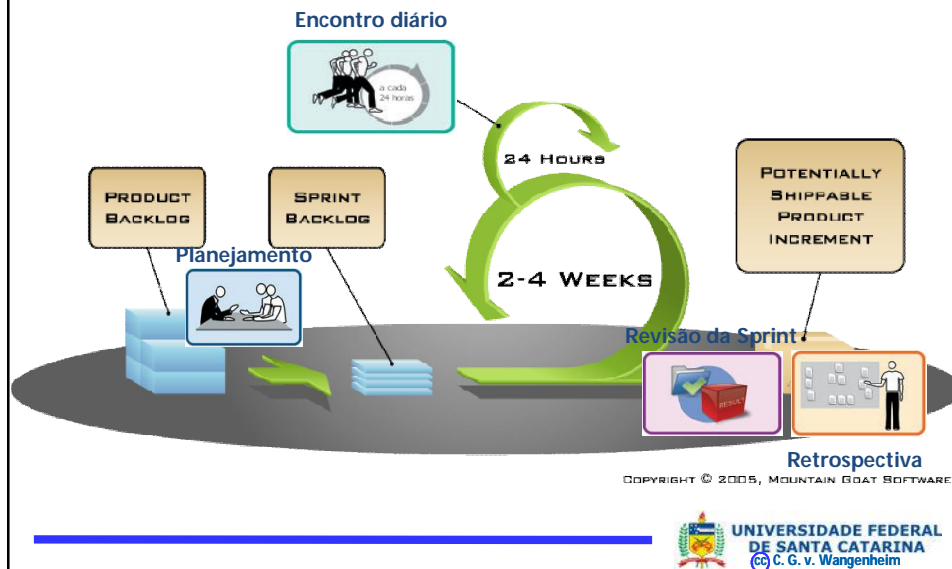
- ❑ Exercício em pequenos grupos.
- ❑ Instruções:
 - ❑ Estimem o tamanho dos itens do *product backlog* para este projeto.

- ❑ Tempo: 5 minutos



UNIVERSIDADE FEDERAL
DE SANTA CATARINA
C. G. v. Wangenheim

Visão geral



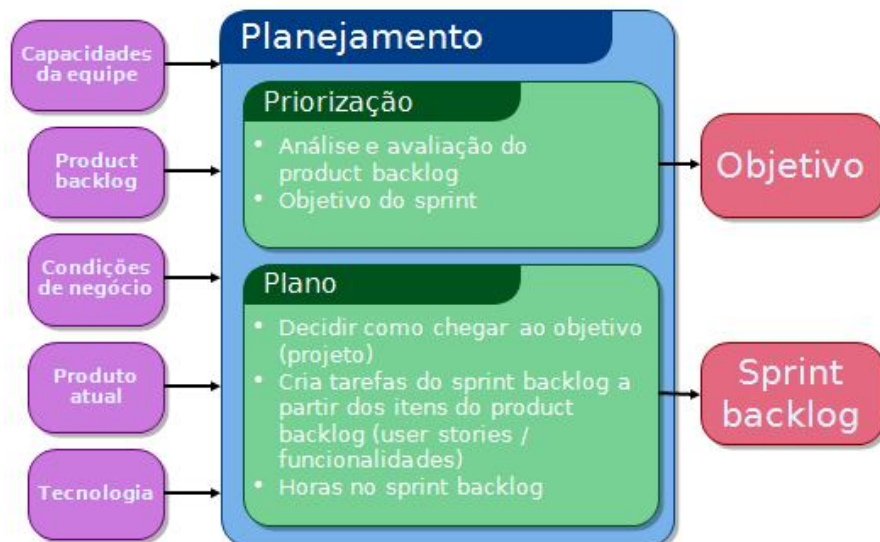
O conceito de *Timebox*

- ❑ **Timebox:** um período em que algo é realizado.
- ❑ **Sprint:** Desenvolvimento de iterações baseado em *timeboxes*.
 - ❑ Definir a data final de iteração, nenhuma alteração permitida.
 - ❑ Se as solicitações não podem ser cumpridos no prazo do *timebox*:
 - ❑ Coloque as solicitações com baixa prioridade de volta na lista de desejos.
 - ❑ Nunca mova um prazo.
 - ❑ Nunca peça aos desenvolvedores para trabalhar mais para cumprir um prazo.
- ⇒ Prazos não podem ser excedidos – ao invés disto, partes do trabalho previsto são eliminadas.

Sprint

- ❑ **Sprint:** um período fixo de (2 a) 4 semanas para desenvolver um produto entregável de valor para o cliente.
 - ❑ Aplicando o conceito de *timebox*.
- ❑ A **entrega de valor** é o objetivo da *sprint* e deverá esta bem definido e concordado com o cliente.
- ❑ *Sprint* inclui: projeto, codificação, testes e documentação.
- ❑ Uma vez que uma *sprint* começou, somente a equipe pode adicionar ou remover itens no *sprint backlog*.
- ❑ O término anormal da *sprint* é feito quando o objetivo da *sprint* já não faz mais sentido.

Planejamento de *sprint*



Reunião de planejamento da *sprint*

- ❑ A reunião de planejamento da *sprint* deve ocorrer dentro de 8 horas com duas partes de 4 horas.
- ❑ Entrada: *Product Backlog*
- ❑ Primeira etapa:
 - ❑ Seleção dos itens do *Product Backlog*, que a equipe se compromete em torná-los incrementos potencialmente implementáveis.
 - ❑ Decisão final é do *Product Owner*
- ❑ Segunda etapa:
 - ❑ Ocorre imediatamente após a primeira.
 - ❑ Como fazer?
 - ❑ Expandir estórias em tarefas.
 - ❑ Estimar as tarefas: até a estimativa do número de horas disponíveis para esta iteração é alcançado.
- ❑ Resultado: *Sprint Backlog*.



Objetivos da *sprint*

- ❑ Objetivo da *sprint*: breve declaração do foco do trabalho durante a *sprint*.
 - ❑ Exemplo: Serviços financeiros: Criar suporte para indicadores de desempenho em tempo real.
- ❑ Resultado da negociação entre o *product owner* e a equipe de desenvolvimento.
- ❑ Objetivos são definidos de maneira que permite no final da *sprint* avaliar de forma visível se foram alcançados ou não.



Planejamento de Sprint

- ❑ Equipe seleciona itens do *Product Backlog* com os quais se compromete a concluir.
- ❑ *Sprint Backlog* é criado.
 - ❑ Tarefas identificadas e estimadas (1 a 16 horas).
 - ❑ De forma colaborativa, não apenas pelo *Scrum Master*.
- ❑ Planejamento alto-nível.

Como um planejador de férias, eu quero ver fotos dos hotéis para poder ter uma idéia melhor das instalações.

Prioridade 4 [10 Story Points]

Modelagem (8 horas)
Codificar interface (4)
Escrever textos (4)
Codificar a classe xx (6)
Atualizar testes de performance (4)



UNIVERSIDADE FEDERAL
DE SANTA CATARINA
C. G. v. Wangenheim

Sprint backlog

Lista o trabalho do *product backlog* a ser concluído na *sprint* atual.

Story As a Clerk, I can generate a bill for the guest's stay.

Tasks
Code read-only access to customer charges in database
Code methods to accumulate individual charges
Code bill formatter for 3rd party report writer
Code the user interface
Write end-user Help

Story As a Manager, I can make unlimited adjustments to a guest's bill

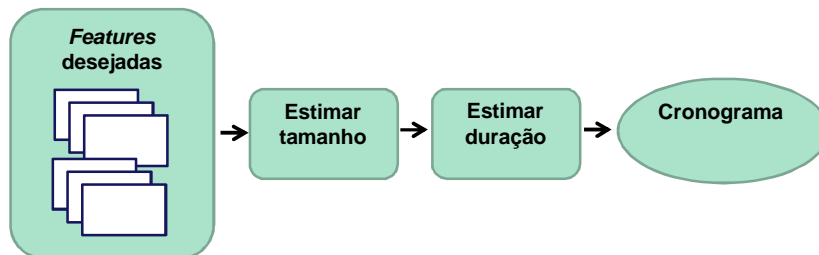
Tasks
Code AdjustmentController class
Code <<façade>> class for charges on guest bill
Code the user interface
Write end-user Help



UNIVERSIDADE FEDERAL
DE SANTA CATARINA
C. G. v. Wangenheim

Estimar tarefas

- ❑ Cada tarefa é estimado em tempo (horas/dia).
- ❑ Idealmente, as tarefas devem ter uma estimativa de 1 a 16 horas.



- ❑ Se o tempo total de tarefas excede as pessoas-horas na *sprint*, tarefas precisam ser excluídas da *sprint*.

Exemplo: *Sprint backlog*

Story As a Clerk, I can generate a bill for the guest's stay.

Estimates in hours

Tasks
Code read-only access to customer charges in database
Code methods to accumulate individual charges
Code bill formatter for 3rd party report writer
Code the user interface
Write end-user Help

4
8
8
8
4

Story As a Manager, I can make unlimited adjustments to a guest's bill

Estimates in hours

Tasks
Code AdjustmentController class
Code <<façade>> class for charges on guest bill
Code the user interface
Write end-user Help

4
8
16
8

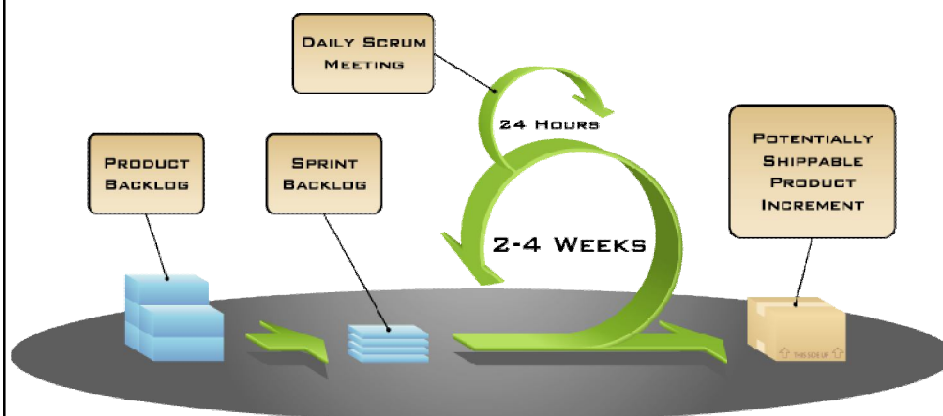
Exercício



- ❑ Exercício em pequenos grupos.
- ❑ Instruções:
 - ❑ Identifiquem o objetivo da Sprint 1 (1 mês) do projeto da pizzeria.
 - ❑ Identifiquem as tarefas dos itens relevantes do product backlog referente ao objetivo da Sprint.
 - ❑ Estimem o tempo das tarefas.
- ❑ Tempo: 10 minutos



Visão geral



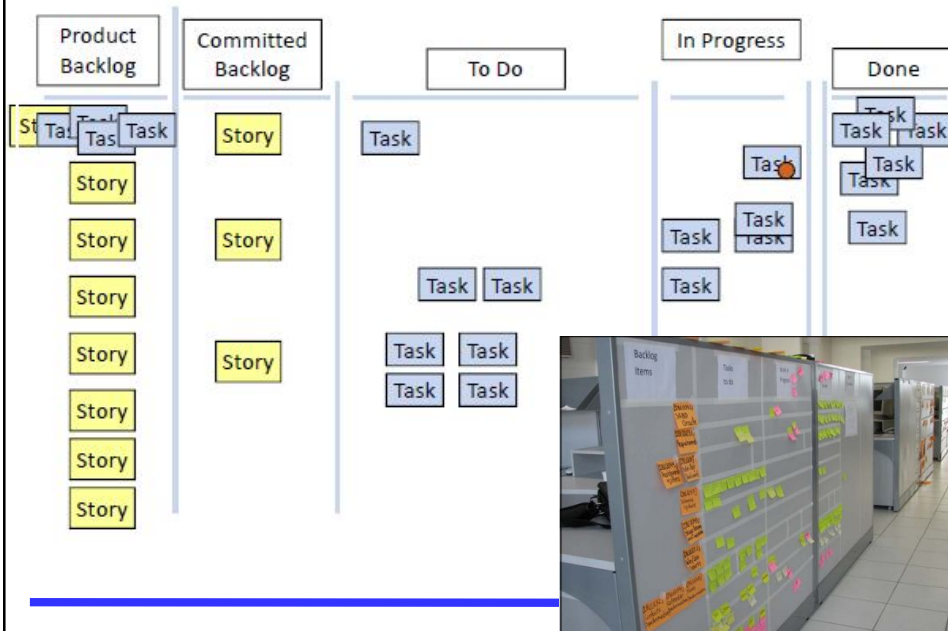
COPYRIGHT © 2005, MOUNTAIN GOAT SOFTWARE



Começar a *Sprint*

- ❑ Preparar o quadro de tarefas (*task board*, *backlog status board*, quadro de *kanban*).
- ❑ Preparar o gráfico de *burndown*.
- ❑ Fazer o *kick-off*.

Quadro de tarefas



Exemplo: Quadro de tarefas (kick-off) DIA 0

Sprint Backlog*	Em Execução	Concluído	BurnDown
Cadastro de Categoria de Apartamentos			
Cadastro de Apartamentos			
Cadastro de Clientes			

UNIVERSIDADE FEDERAL DE SANTA CATARINA
C. G. v. Wangenheim

Gráfico de *burndown* da *sprint*

- Usado para monitorar o progresso e previsão.

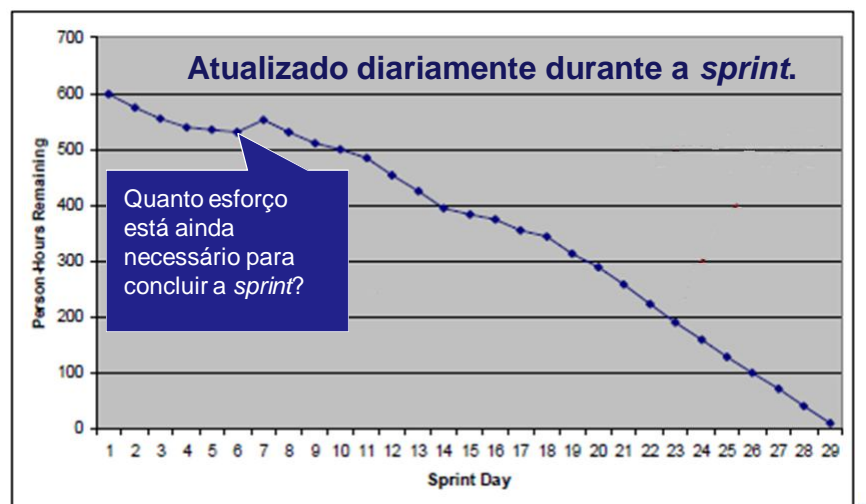
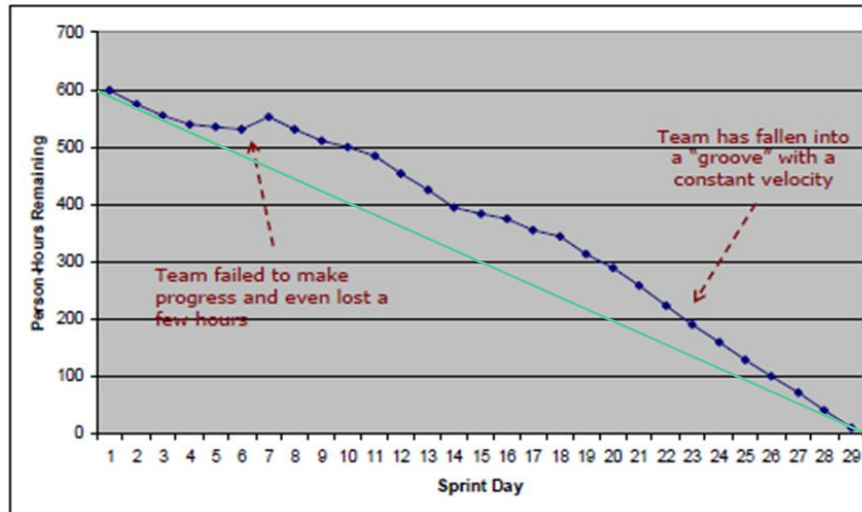


Gráfico de *burndown* da *sprint*

- ❑ Medir e monitorar o progresso com o esforço restante vs. tempo restante.



UNIVERSIDADE FEDERAL
DE SANTA CATARINA
C. G. v. Wangenheim

Gerenciando o *sprint backlog*

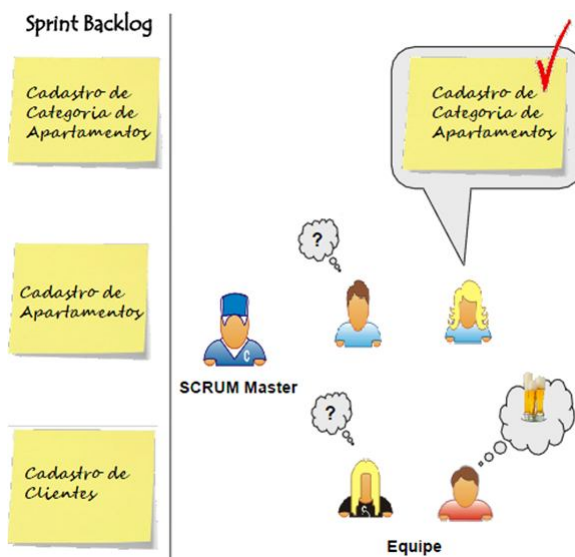
- ❑ Cada indivíduo escolhe o trabalho que fará.
 - ❑ Trabalhos nunca são atribuídos.
- ❑ Atualização diária da estimativa do trabalho restante.
- ❑ Qualquer membro da equipe pode adicionar, apagar ou mudar tarefas.



UNIVERSIDADE FEDERAL
DE SANTA CATARINA
C. G. v. Wangenheim

KICK-OFF

Exemplo: reunião de kick-off DIA 0



Exemplo: Quadro de tarefas após *kick-off*

Sprint Backlog	Em Execução	Concluído	BurnDown
<div>Cadastro de Apartamentos</div> <div>Cadastro de Clientes</div>	<div>Cadastro de Categoria de Apartamentos</div>		



UNIVERSIDADE FEDERAL
DE SANTA CATARINA
C. G. v. Wangenheim

Final do 1. Dia



UNIVERSIDADE FEDERAL
DE SANTA CATARINA
C. G. v. Wangenheim

Reunião diário do Scrum

- ❑ Diariamente 15-minutos de reunião.
- ❑ Equipe fica **em pé** num círculo de frente para o outro.
- ❑ Participantes:
 - ❑ Todo mundo é convidado.
 - ❑ Apenas os membros da equipe, *ScrumMaster* e *Product Owner* podem falar.
- ❑ Ajuda a evitar reuniões adicionais desnecessárias:
 - ❑ Melhora a comunicação.
 - ❑ Estimula a um desenvolvimento baseado em equipe, rápido, intenso e cooperativo.



Cada um responde 3 perguntas

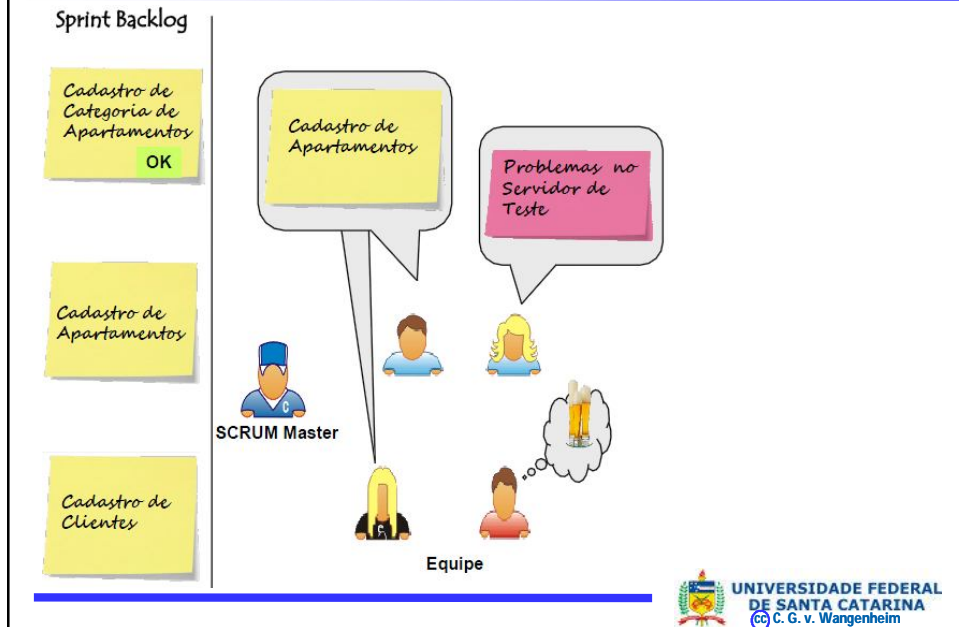
1
O que fizeste ontem?

2
O que vais fazer hoje?

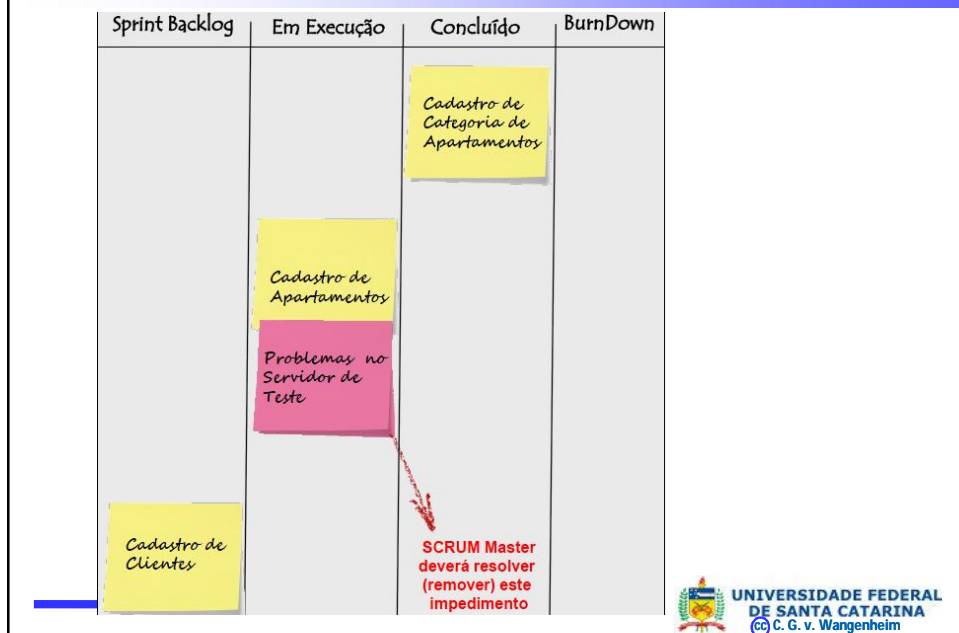
3
Há algum obstáculo?

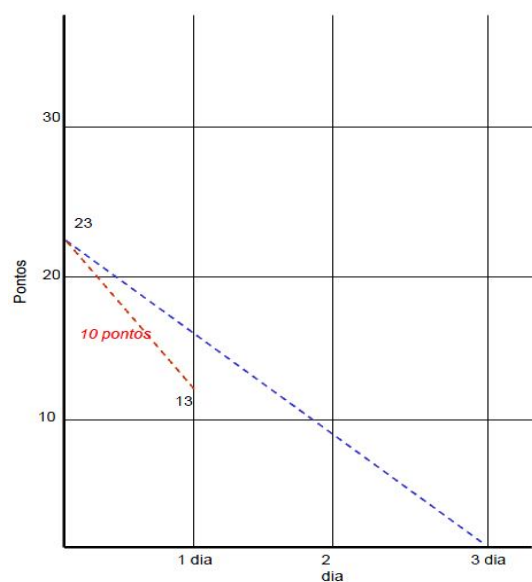
As respostas NÃO são um “relatório” para o *Scrum Master* -
são COMPROMISSOS perante os pares.

Exemplo: primeira reunião diária DIA 1



Exemplo: quadro de tarefas após a reunião diária



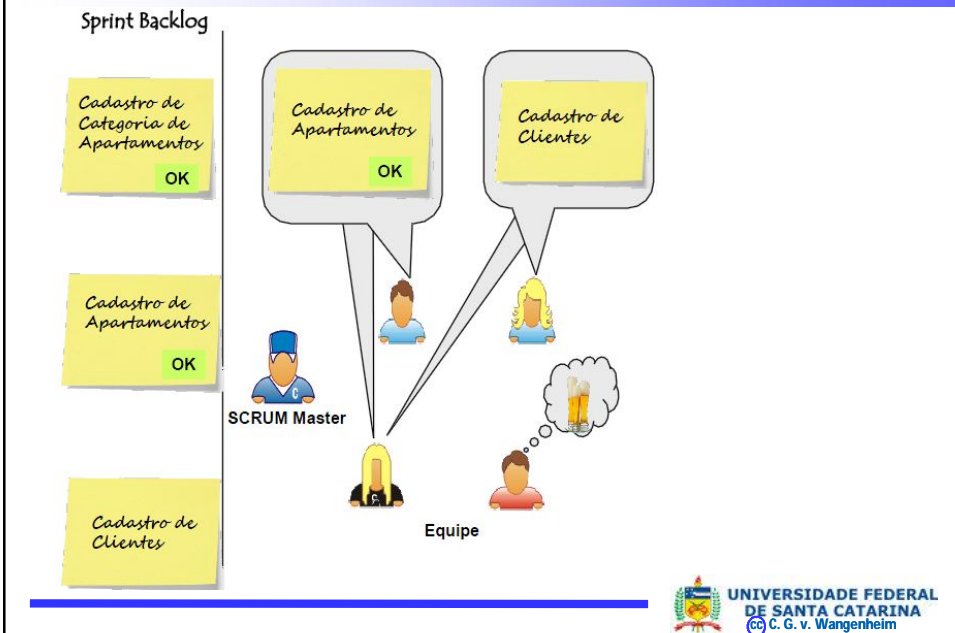
Exemplo: *Burndown* no final do dia 1

*Cadastro de
Categoria de
Apartamentos*

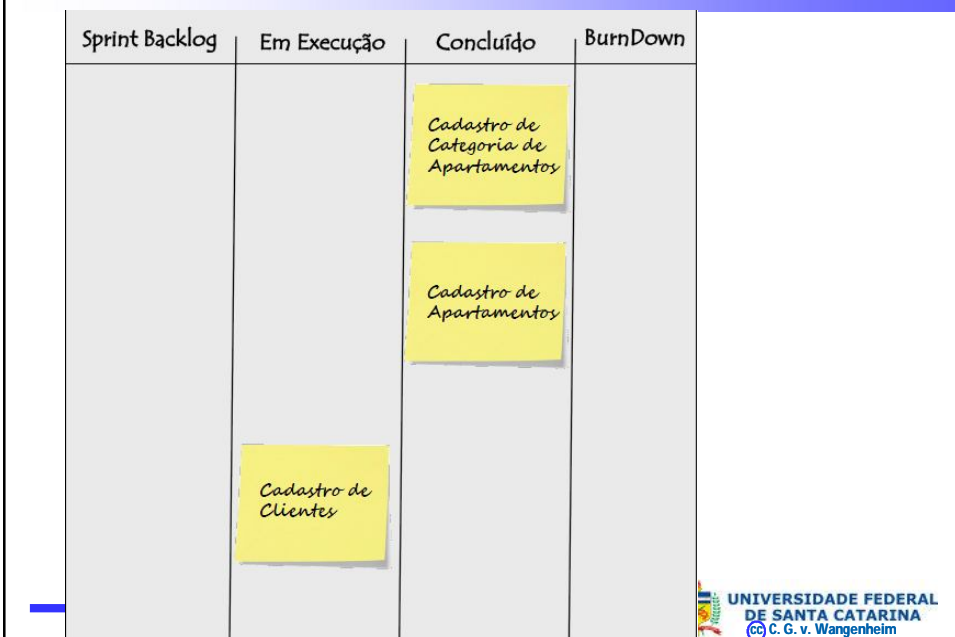
OK

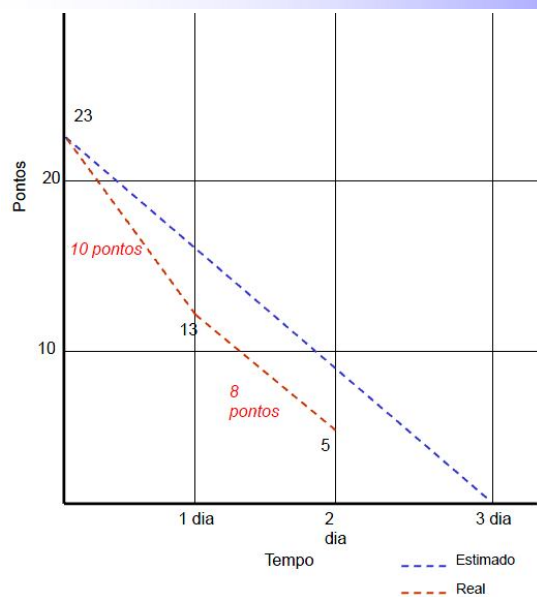
DIA 2

Exemplo: reunião diária no DIA 2



Exemplo: Quadro de tarefas após da reunião diária



Exemplo: Burndown DIA 2

Cadastro de
Categoria de
Apartamentos

OK

Cadastro de
Apartamentos

OK

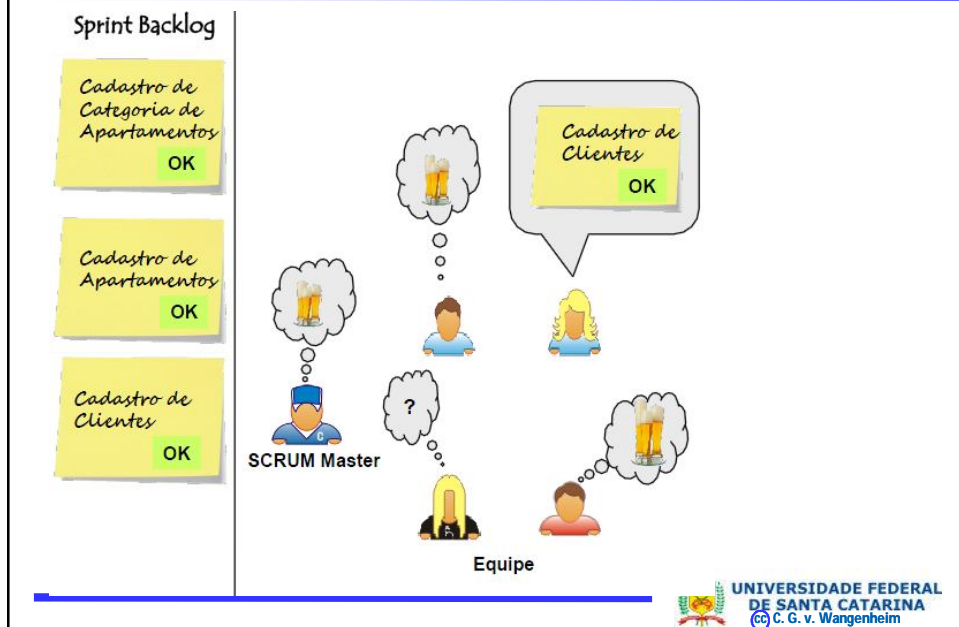


UNIVERSIDADE FEDERAL
DE SANTA CATARINA
C. G. v. Wangenheim

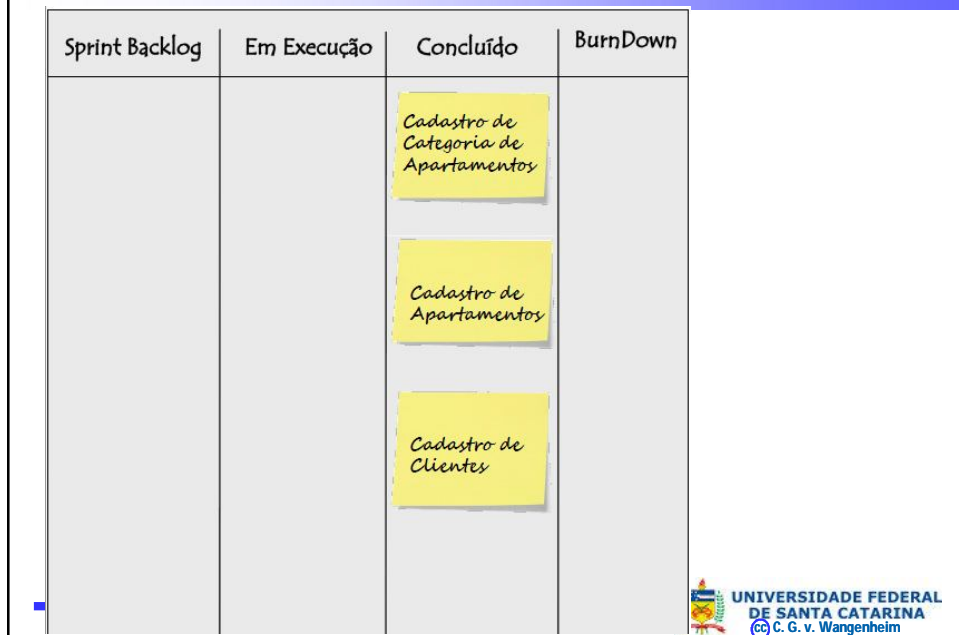
DIA 3

UNIVERSIDADE FEDERAL
DE SANTA CATARINA
C. G. v. Wangenheim

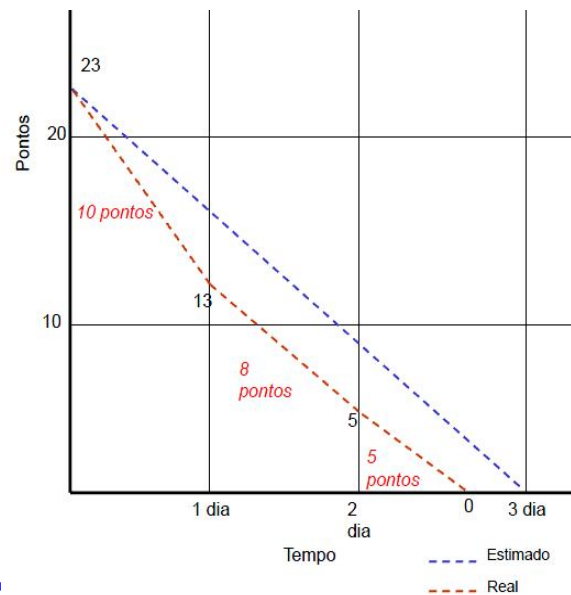
Exemplo: reunião diária DIA 3



Exemplo: Quadro de tarefas DIA 3



Exemplo: Burndown DIA 3



Cadastro de
Categoria de
Apartamentos
OK

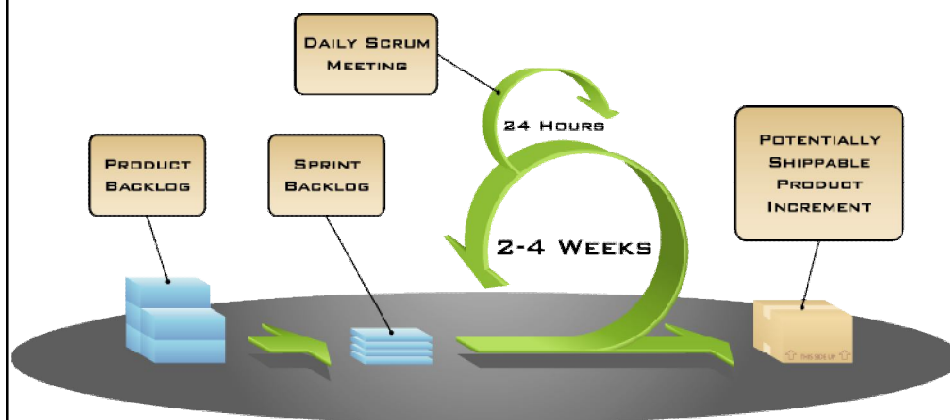
Cadastro de
Apartamentos
OK

Cadastro de
Clientes
OK



UNIVERSIDADE FEDERAL
DE SANTA CATARINA
C. G. v. Wangenheim

Visão geral



COPYRIGHT © 2005, MOUNTAIN BOAT SOFTWARE



UNIVERSIDADE FEDERAL
DE SANTA CATARINA
C. G. v. Wangenheim

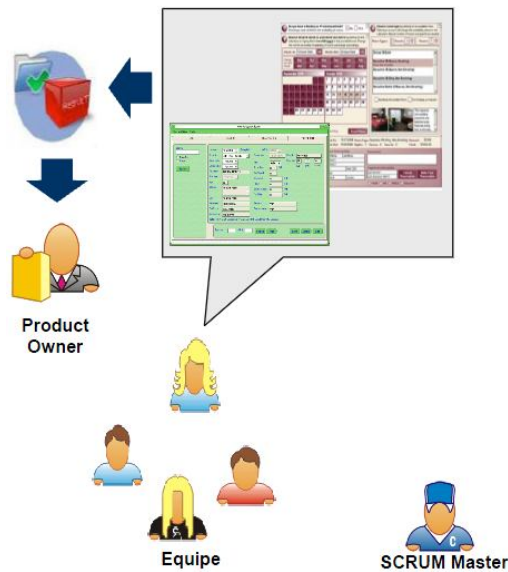
Reunião de revisão da Sprint (*Sprint Review*)

- ❑ Reunião de no máximo 4 horas sob responsabilidade do *Scrum Master*.
- ❑ Objetivos:
 - ❑ Validar o objetivo da iteração.
 - ❑ Validar finalização das histórias.
- ❑ Objetivo do *sprint* e *product backlog* são comparados com os resultados reais da *sprint*.
- ❑ Razões para discrepâncias são discutidas.

Formato de uma reunião de revisão de *sprint*

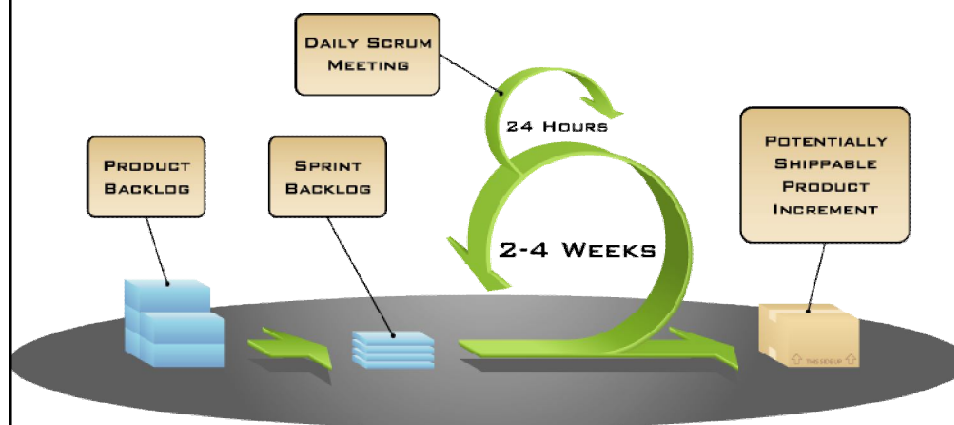
- ❑ *Scrum Master* dá uma visão geral da *sprint*.
- ❑ Equipe apresenta os resultados obtidos durante a *sprint*.
 - ❑ Tipicamente, demonstra novas funcionalidades ou sua arquitetura.
 - ❑ Informal
 - ❑ max. de 1-2 horas de preparação.
 - ❑ sem slides.
- ❑ No final da reunião:
 - ❑ Cada *stakeholder* fala suas impressões e sugere mudanças com suas respectivas prioridades.
 - ❑ Possíveis modificações no *Product Backlog* são discutidas entre o *Product Owner* e a equipe.
 - ❑ *Scrum Master* anuncia a data e o local da próxima reunião de revisão da *sprint* ao *Product Owner* e a todos *stakeholders*.

Exemplo: reunião de revisão da *sprint*



UNIVERSIDADE FEDERAL
DE SANTA CATARINA
C. G. v. Wangenheim

Visão geral



COPYRIGHT © 2005, MOUNTAIN GOAT SOFTWARE

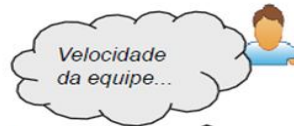


UNIVERSIDADE FEDERAL
DE SANTA CATARINA
C. G. v. Wangenheim


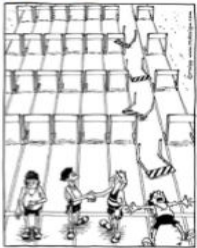

Retrospectiva da *sprint* (Sprint Retrospective)

- ❑ Feito após cada *sprint*.
- ❑ Reunião de não mais do que 3 horas.
- ❑ Objetivo:
 - ❑ Observar o que funcionou e o que não funcionou.
 - ❑ Avaliar e aprender com a experiência da iteração visando aumentar a produtividade.
 - ❑ Criar uma base de conhecimento de lições aprendidas.
- ❑ Nesta reunião participam: Equipe, *ScrumMaster* e, opcionalmente, *Product Owner*.
- ❑ Os membros da equipe devem responder a duas questões:
 - ❑ O que aconteceu de bom durante a última *sprint*?
 - ❑ O que pode ser melhorado para a próxima *sprint*?
- ❑ *ScrumMaster* escreve as respostas e prioriza na ordem que deseja discutir as potenciais melhorias.

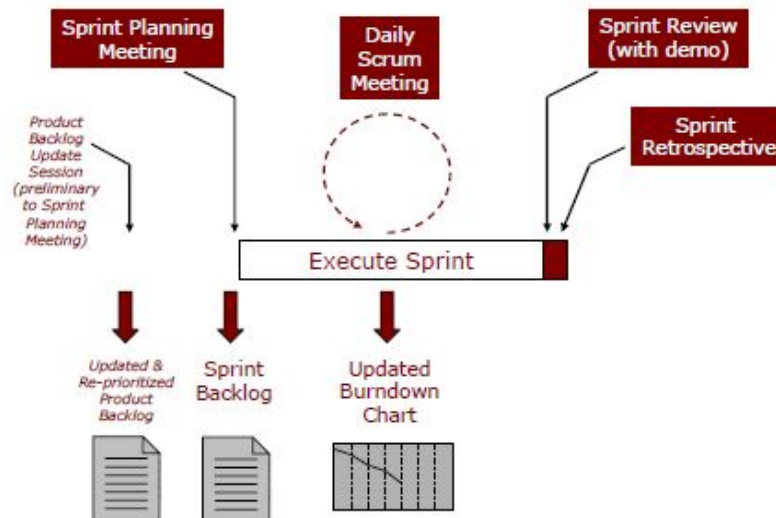
Exemplo: Retrospectiva da *sprint*



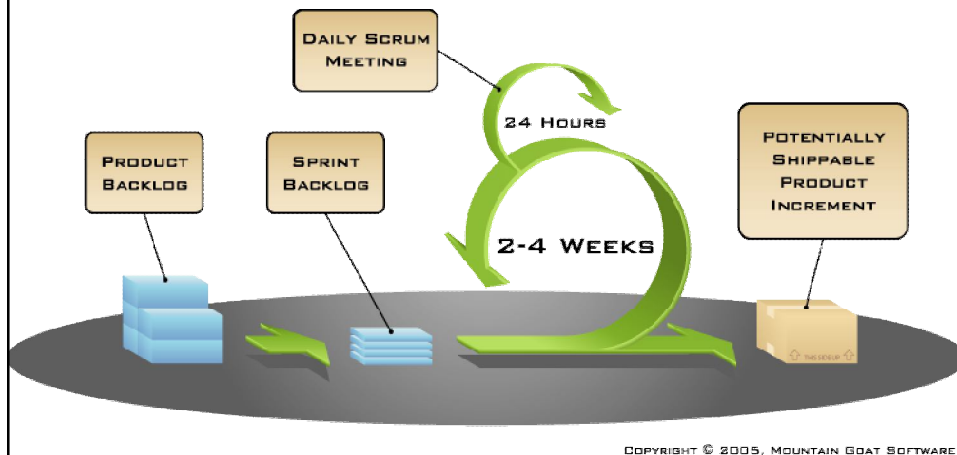
Equipe

Exemplo	OK	Pontos de Atenção	O Que Deve Ser Melhorado 
	<p>Cadastro de Categoria de Apartamentos</p> <p>Cadastro de Apartamentos</p> <p>Cadastro de Clientes</p>	<p>Velocidade da equipe</p>  <p>Será necessário mais atenção na hora de estimar as histórias</p>	 <p>Atitude: Para uma equipe (time) SCRUM funcionar será necessário mudança de atitude, caso contrário isto poderá afetar o desempenho da equipe</p> <p>Impedimentos:</p> <p>Problemas no Servidor de Teste</p> <p>Planejamento: Prestar atenção na hora do planejamento da Sprint, para identificar se todos os recursos necessário estão disponíveis</p>

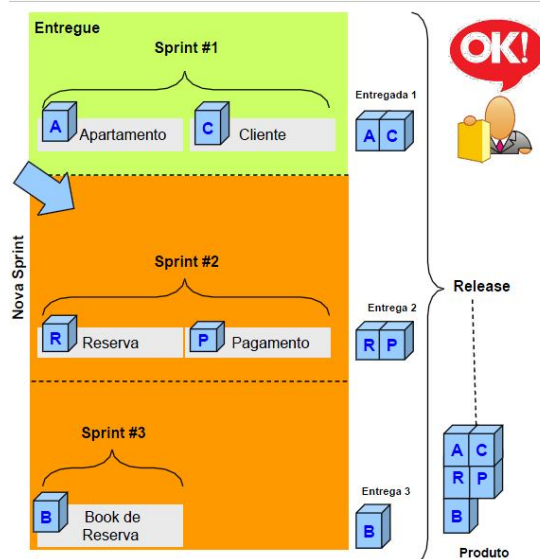
Cerimônias no Scrum



Visão geral

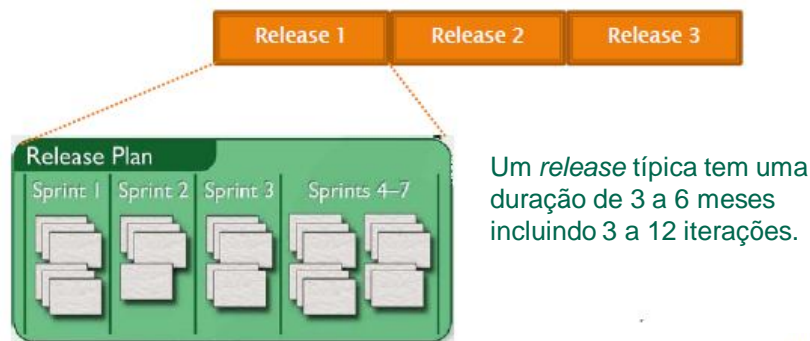


Nova sprint



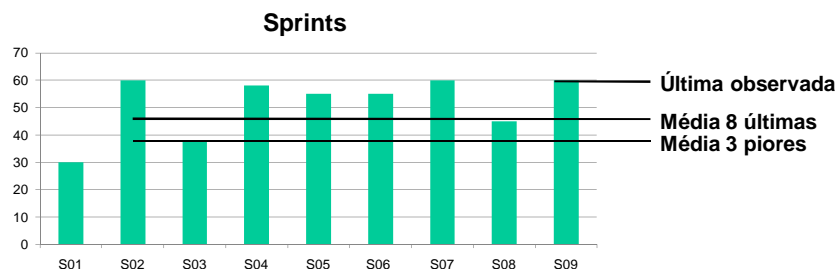
Planejamento de *releases*

- ❑ Processo de criação de um plano alto nível, que cobre um período maior do que uma iteração.
- ❑ **Release Planning**: $\text{Total de pontos do backlog} / \text{velocidade} = \sim \text{no. de sprints}$.
- ❑ **Velocidade da equipe**: Quanto a equipe pode realizar em uma *sprint*.



Velocidade da equipe

- ❑ A velocidade da equipe é a unidade de medida de produtividade de cada *sprint*.
- ❑ Velocidade pode alterar entre *sprints*.
- ❑ Ao longo do projeto a velocidade tende a ter um valor estável.

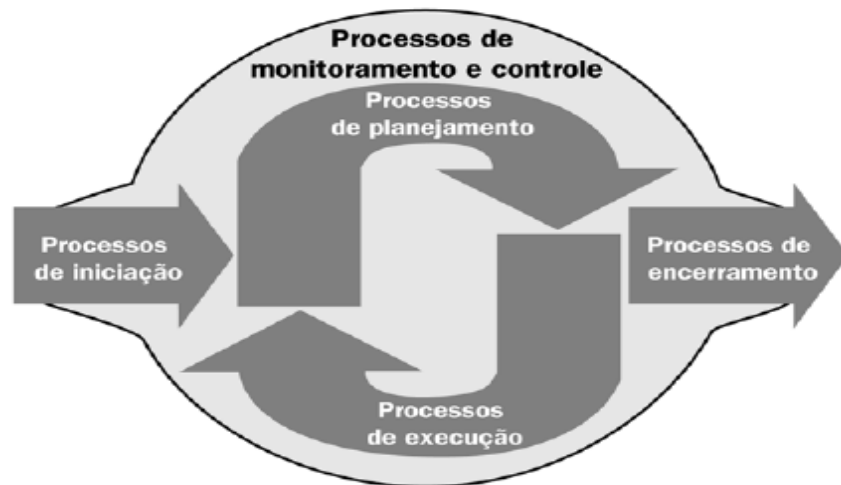


Como estimar velocidade?

- ☐ Use valores históricos: com base em projetos similares realizados no passado.
- ☐ Com base na velocidade observada durante a(s) iterações anterior(es) neste projeto.
 - ☐ Exemplo: Se a equipe completou 3 iterações com velocidades de 12, 15 e 16 ⇒ velocidade estimada está na faixa de 12 - 16.
- ☐ Faça uma previsão.

SCRUM vs. Métodos tradicionais

Grupos de processos de gerenciamento de projetos



Algumas diferenças chaves

Scrum	PMBOK
Equipes são auto-gerenciáveis.	Equipes são gerenciadas.
<i>Product Owner</i> prioriza características - Equipe define o trabalho.	Gerente de projeto define e prioriza o trabalho.
<i>Scrum Master</i> é responsável pela remoção dos obstáculos para o progresso.	Gerente de projeto é responsável pela gerência do projeto.
Equipe é responsável pelo sucesso ou fracasso do projeto.	O gerente de projeto é responsável pelo sucesso ou fracasso do projeto.
Status é comunicado publicamente por postar o <i>sprint backlog</i> , <i>burndown</i> , e demonstrações na revisão de <i>sprint</i> .	Monitoramento & controle de desempenho.
Poucos artefatos de planejamento.	Muitas entradas e saídas.
Gerenciamento de riscos implícita.	Gerência de riscos explícita.

Vantagens do SCRUM

- ☐ Desenvolvimento de software focado no valor de negócio - reduz o risco de ROI.
- ☐ Permite que os desenvolvedores focam em entregar funcionalidade usável ao cliente.
- ☐ Melhora produtividade pela implementação de uma estrutura que dá autoridade as equipes.
- ☐ Capacidade de responder a exigências imprevisíveis e mudanças.

Quando não usar SCRUM?

- ☐ O ambiente não é flexível.
- ☐ A cultura corporativa não é propícia a este ambiente de desenvolvimento.
- ☐ Nenhum apoio à gerência.
- ☐ Nenhum treinamento formal disponível.
- ☐ (Equipes de desenvolvedores são maiores de 10 pessoas e/ou distribuídos em vários locais.)

E como se tornar um SCRUM Master?



- ❑ Atender treinamento formal.
 - ❑ Previsto: prova de Certified ScrumMaster (CSM).
- ❑ Associar à *Scrum Alliance*.
- ❑ Renovação anual:
 - ❑ Renovar associação (não há necessidade de re-certificação).

<http://www.scrumalliance.org/training>



PMI Agile Certified Practitioner (PMI-ACP)

- ❑ Certificação oferecida pelo PMI a partir de Agosto 2011

PMI Agile Certification Eligibility Requirements	
Requirement	Description
Educational Level	Secondary degree (high school or equivalent) or higher
General Project Management Experience	2,000 hours working on project teams. These hours must be earned within the last 5 years. Note: for those holding a PMP® credential, PMI has already verified that the candidate has exceeded these requirements. Thus, a PMP will be accepted to fulfill these requirements.
Agile Project Management Experience	1,500 hours working on Agile project teams. These hours are in addition to the 2,000 hours required in general project management experience. These hours must be earned within the last 2 years.
Agile Project Management Training	21 contact hours; hours must be earned in Agile project management topics
Examination	Tests knowledge of Agile fundamentals
Maintenance	30 PDUs/3 CEUs every 3 years in Agile project management Note: these hours would count toward fulfilling PMP requirements, if the individual holds both certifications

<http://www.pmi.org/en/Certification/New-PMI-Agile-Certification.aspx>



Leia mais sobre este tema ...



Ken Schwaber. Agile Project Management with Scrum. Microsoft Press, 2004.

Mike Cohn. Agile Estimating and Planning. Prentice Hall, 2005.

Mike Cohn. User Stories Applied for Agile Development. Addison-Wesley Professional, 2004.

Outras referências:

Henrik Kniberg. Scrum and XP from the Trenches. InfoQ, 2007.

<http://www.infoq.com/minibooks/scrum-xp-from-the-trenches>

[http://en.wikipedia.org/wiki/Scrum_\(management\)](http://en.wikipedia.org/wiki/Scrum_(management))

Rildo F Santos. SCRUM experience.

An Introduction to SCRUM. MountainGoat Software. Slides.

Mais um jogo para aprender SCRUM:

<http://www.mountaingoatsoftware.com/products/scrum-game>



UNIVERSIDADE FEDERAL
DE SANTA CATARINA
C. G. v. Wangenheim



Atribuição-Uso Não-Comercial-Compartilhamento pela Licença 2.5 Brasil

Você pode:

- copiar, distribuir, exibir e executar a obra
- criar obras derivadas

Sob as seguintes condições:

Atribuição — Você deve dar crédito ao autor original, da forma especificada pelo autor ou licenciante.

Uso Não-Comercial — Você não pode utilizar esta obra com finalidades comerciais.

Compartilhamento pela mesma Licença — Se você alterar, transformar, ou criar outra obra com base nesta, você somente poderá distribuir a obra resultante sob uma licença idêntica a esta.

Para ver uma cópia desta licença, visite <http://creativecommons.org/licenses/by-nc-sa/2.5/br/> ou mande uma carta para Creative Commons, 171 Second Street, Suite 300, San Francisco, California, 94105, USA.



UNIVERSIDADE FEDERAL
DE SANTA CATARINA
C. G. v. Wangenheim