



# SCRUM Experience

@tecnologia.com.br

Rildo F Santos

[rildosan@etecnologia.com.br](mailto:rildosan@etecnologia.com.br)

[rildo.santos@companyweb.com.br](mailto:rildo.santos@companyweb.com.br)

Twitter: <http://twitter.com/rildosan>  
Blog: <http://rildosan.blogspot.com/>



Tenho mais de 10.000 horas de experiência em Gestão de Negócios, Governança e Engenharia de Software.

Sou formado em Administração de Empresas, Pós-Graduado Didática do Ensino Superior e Mestre em Engenharia de Software pela Universidade Mackenzie.

Atuo em Gestão de Negócio (Inovação, Planejamento e Estratégia, Inteligência de Negócio, Gestão por Processo, e GRC - Governance, Risk and Compliance) e em projetos de Engenharia de Software utilizando métodos Agile (SCRUM, Lean, XP e FDD). Sou **Agile Coach**.

Fui instrutor de Tecnologia de Orientação a Objetos, UML e Linguagem Java na Sun Microsystems e da IBM.

Conheço Arquitetura de Software, SOA (Arquitetura Orientado a Serviço), RUP/UP - Processo Unificado, Business Intelligence, Gestão de Risco de TI entre outras tecnologias.

Sou professor de curso de MBA da Fiap e fui professor de pós-graduação da Fasp e IBTA.

Tenho vivência na implementação de Governança de TI e Gerenciamento de Serviços de TI, Conhecimento dos principais frameworks e padrões: ITIL, Cobit, ISO 27001 e ISO 15999;

Desempenhei diversos papéis como: Estrategista de Negócio, Gerente de Negócio, Gerente de Projeto, Arquiteto de Software, Projetista de Software e Analista de Sistema em diversos projetos em empresas como: Bradesco, Editora Abril, Scopus, Porto Seguro, Certagy, Secretária da Fazenda SP, Sonagol (Angola), Honda, Dix-Amico, Bank Tokyo-Mitsubishi, Vivo, Hospital das Clínicas, Aços Villares, Novabase do Brasil, Polícia Militar do Estado de São Paulo entre outras.

Possui as certificações: CSM - Certified SCRUM Master, CSPO - Certified SCRUM Product Owner, SUN Java Certified Instrutor, ITIL Foundation e Instrutor Oficial de Cobit Foundation e Cobit Games;

É membro: IIBA-International Institute of Business Analysis (Canada)

Twitter: <http://twitter.com/rildosan>

Blog: <http://rildosan.blogspot.com/>

Trabalho com desenvolvimento de software a quase 15 anos entre idas e vindas.

Como todo desenvolvedor tenho preferência por uma linguagem de programação, no meu caso é o Java.

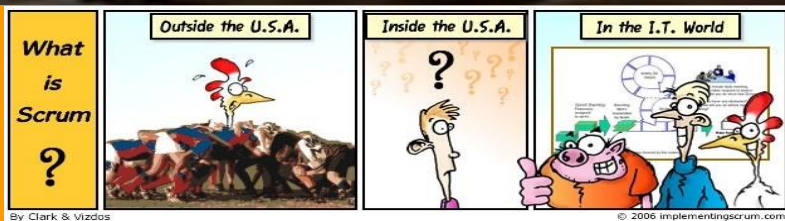
Confesso que já utilizei diversos frameworks, padrões e modelos para o construir software, contudo, os Métodos Ágeis me chamaram a atenção para um detalhe, que hoje julgo extremamente importante...

A essência dos métodos ágeis são as pessoas e **não o processo**, o respeito e a disciplina são outros aspectos que eu considero grandiosos.

Nos últimos dois anos estou trabalhando com SCRUM, o qual me fascina pela simplicidade e o foco em resultado.

Escrevi este tutorial para compartilhar minha experiência com as práticas SCRUM para as pessoas que estão tendo o primeiro contato com os Métodos Ágeis e necessitam de guia ou uma referência para entender como SCRUM funciona na prática.

Atualmente estou tentando empregar práticas do SCRUM em outras áreas de negócio (processo de negócio), este é meu desafio.



# Play SCRUM



# SCRUM:

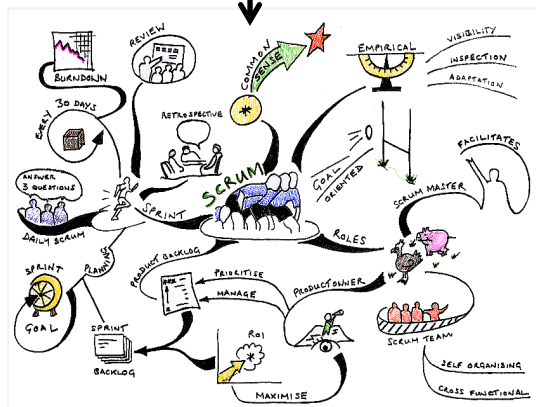
## As origens

The New, New  
Product  
Development  
Game

Iterative,  
Incremental  
Development

TimeBoxes

SmallTalk  
Engineering Tools



## O que é SCRUM ?

SCRUM é um processo iterativo e incremental para desenvolvimento de qualquer produto ou gerenciamento de qualquer trabalho...

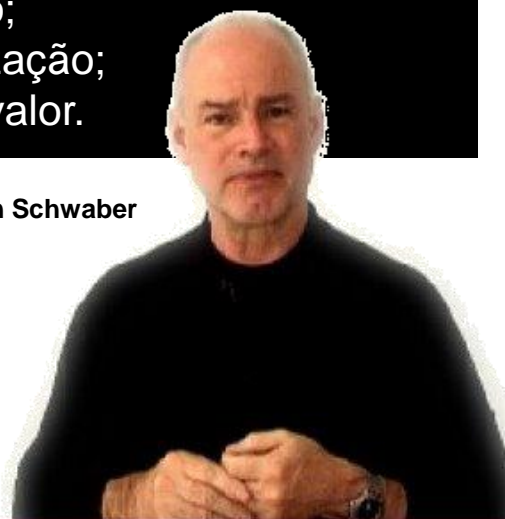
## SRUM:

É Um Processo empírico de gerenciamento e controle;  
Faz a inspeção e adaptação em loops de feedback;  
Faz entrega de valor ao cliente em até 30 dias;  
“Escalável” para suportar grandes projetos;  
Compatível com CMM3 e ISO9001;  
Extremamente simples, mas muito resistente...

## Valores do Scrum:

- Transparência;
- Integridade: assim que perceber algo, faça algo;
- Ser empírico;
- Auto-organização;
- Entrega de valor.

Ken Schwaber



# Manifesto Ágil:

## Manifesto para o desenvolvimento ágil de software

Estamos descobrindo maneiras melhores de desenvolver software fazendo-o nós mesmos e ajudando outros a fazê-lo. Através deste trabalho, passamos a valorizar:

**Indivíduos e interação entre eles** mais que processos e ferramentas

**Software em funcionamento** mais que documentação abrangente

**Colaboração com o cliente** mais que negociação de contratos

**Responder a mudanças** mais que seguir um plano

Ou seja, mesmo havendo valor nos itens à direita, valorizamos mais os itens à esquerda.

Kent Beck

Mike Beedle

Arie van Bennekum

Alistair Cockburn

Ward Cunningham

Martin Fowler

James Grenning

Jim Highsmith

Andrew Hunt

Ron Jeffries

Jon Kern

Brian Marick

Robert C. Martin

Steve Mellor

Ken Schwaber

Jeff Sutherland

Dave Thomas

## Princípios por trás do Manifesto Ágil:

A prioridade é **satisfazer o cliente**, entregando o mais rápido possível e de forma contínua software que tenha valor;

**Requisitos mutantes são bem vindos**, mesmo no final do desenvolvimento. Os processos ágeis podem ser usados a favor de mudanças que tragam vantagem competitiva para o cliente;

**É importante entregar software funcionando frequentemente**, mensalmente, quinzenalmente ou, se possível, toda semana;

**Clientes e desenvolvedores devem trabalhar juntos** diariamente num projeto;

**Projetos devem ser feitos por indivíduos motivados**. Os indivíduos precisam da confiança de que seu trabalho será realizado. Eles devem ter suas necessidades atendidas e trabalhar num ambiente adequado;

**Conversa face-a-face é SEMPRE a melhor forma de comunicação**;

**Software funcionando é a primeira medida de progresso**;

**O processo ágil torna o desenvolvimento sustentável**. Patrocinadores, desenvolvedores e usuários devem manter a paz indefinidamente;

**Atenção constante à excelência técnica** e bom design aumenta a agilidade;

A chave é **SIMPLICIDADE**: a arte de minimizar a quantidade de trabalho desnecessário;

**As melhores arquiteturas, requisitos e design surgem de equipes auto-organizadas**;

**Em intervalos regulares, a equipe reflete como se tornar mais eficiente. Então ajusta seu comportamento para atingir esse objetivo.**



# Como ser Ágil:

## Manifesto para o desenvolvimento ágil de software

Estamos descobrindo maneiras melhores de desenvolver software fazendo-o nós mesmos e ajudando outros a fazê-lo. Através deste trabalho, passamos a valorizar:

**Indivíduos e interação entre eles** mais que processos e ferramentas

**Software em funcionamento** mais que documentação abrangente

**Colaboração com o cliente** mais que negociação de contratos

**Responder a mudanças** mais que seguir um plano

Ou seja, mesmo havendo valor nos itens à direita, valorizamos mais os itens à esquerda.

## Como ser ágil ?

Para “ser ágil” é preciso colocar em prática os valores e os princípios ágeis (*aqueles descritos no Manifesto Ágil*).

## Quando utilizar os Métodos os Ágeis ?

Antes de responder esta questão façamos uma outra pergunta:

**Se os Métodos Ágeis ajudará a melhorar os resultados (ou sucesso) dos projetos de desenvolvimento de software ?**

Se a resposta for **SIM**, não perca tempo, arregace as mangas e vá a luta



☺ *Lembre-se que o bom processo de desenvolvimento de software é aquele que permite que a equipe tenha sucesso*

# Iterativo e Incremental:

## Incremental

Entrega 1



Entrega 2



Entrega 3

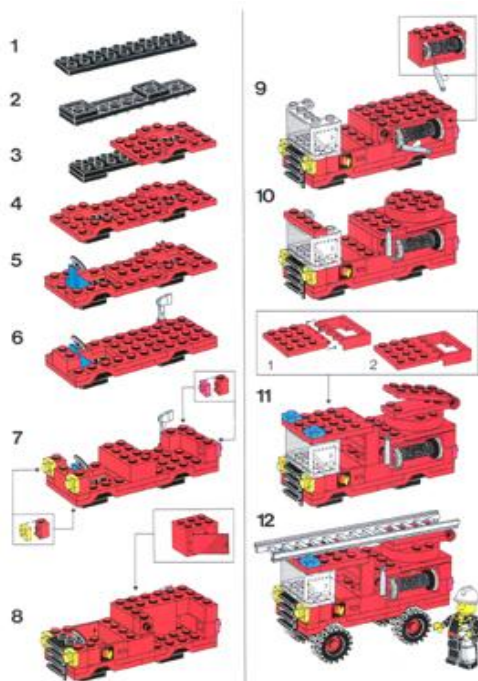


## Iterativo



Devido a complexidade, tamanho, mudanças de requisitos, urgência e necessidade de demonstrar valor mais rápido, fica quase inconcebível desenvolver software utilizando o modelo cascata, ou seja desenvolver todo o software de uma única vez.

Desenvolvimento **Iterativo e incremental** é uma **estratégia de planejamento** (que segue a linha: **dividir para conquistar** 😊), onde o software é construído em partes, ou seja, em ciclos (iterações), a cada iteração é feito um novo incremento (parte do software funcional) até completar o software.





# TimeBox e Sprint

## O que é Timebox ?

É um conceito diz que a quantidade de tempo (horas ou dias) é imutável, ou seja, a quantidade de horas não poderá aumentar. Assim, evita-se atraso no prazo de entrega e facilita o planejamento.

Entretanto, quanto se erra a estimativa de tempo (leia-se: horas ou dias) de uma Sprint (leia-se: iteração), neste caso é recomendável reduzir o escopo da Sprint, **desde que não afete a meta da Sprint** (isto é discutido um mais a frente) ao invés de aumentar a quantidade de horas/dias.

**Timebox** = Um prazo ou tempo (dias/horas, por exemplo) bem definido e imutável.

## O que é uma Sprint ?

É uma iteração (que pode ser parte de uma release) que deve ser realizada entre 2 a 4 semanas, no qual a equipe do projeto deverá produzir um entregável de valor para o cliente (lembre-se dos Princípios do Manifesto Ágil).

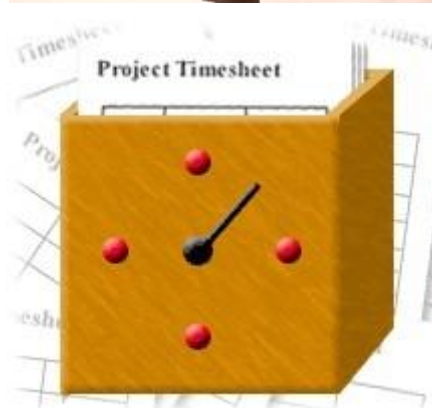
A **entrega de valor é a meta da Sprint** que deverá esta bem definida e combinada com o cliente, antes do começo da execução da Sprint.

O conceito de **Timebox** é aplicado a **Sprint**.

O conceito de timebox é aplicado as cerimônias (reuniões) do Scrum. Todas as reuniões são Timeboxed:

- **Reunião de Planejamento da Sprint (8 horas)**
- **Reunião Diária (15 minutos)**
- **Reunião de Revisão da Sprint (4 horas\*)**
- **Reunião de Retrospectiva da Sprint (3 horas\*)**

*Nota: \* A quantidade de horas pode variar de acordo com a necessidade (por exemplo, apresentação do que será entregue ao cliente) ou aquilo que será discutido/debatido, neste caso a Retrospectiva ela poderá variar entre 1 a 3 horas*



# Não existe Bala de Prata:



## SCRUM não é a Bala de Prata:

*Veja Lei F. Brooks,  
Não existe bala de prata*

O SCRUM não é a solução completa para os problemas de produtividade, complexidade, custo, prazo e qualidade do processo de desenvolvimento de software.

**“Não existe solução mágica para problemas complexos”**

### Contudo, você pode utilizar o SCRUM:

- Para desenvolvimento de software complexos onde os requisitos mudam rapidamente e constantemente;
- Para gerenciar e controlar desenvolvimento de trabalho;
- Para preservar as praticas de engenharia de software existentes e que já são conhecidas;
- Para tornar sua equipe auto-gerenciável e multifuncional;
- Para implementar o conceito iterativo e incremental no desenvolvimento de software e/ou produtos;
- Para facilitar a identificação de causa raiz e a remoção de qualquer coisa que esteja impedindo o desenvolvimento e/ou entrega de software/produtos;
- Para ser caminho da (alta) produtividade;
- Para valorizar os indivíduos (as pessoas).

# Algumas empresas que estão usando SCRUM:

Google™



B B C

amazon.com

YAHOO!

SIEMENS



Microsoft

Borland

NOKIA  
Connecting People

PHILIPS  
nielsen  
.....



Canon

THE DOCUMENT COMPANY  
XEROX®



salesforce.com ~~SOFTWARE~~  
Success On Demand.™



Quais empresas estão  
utilizando o  
SCRUM?

Algumas empresas  
brasileiras



Powerlogic  
Tecnologia Orientada ao Cliente

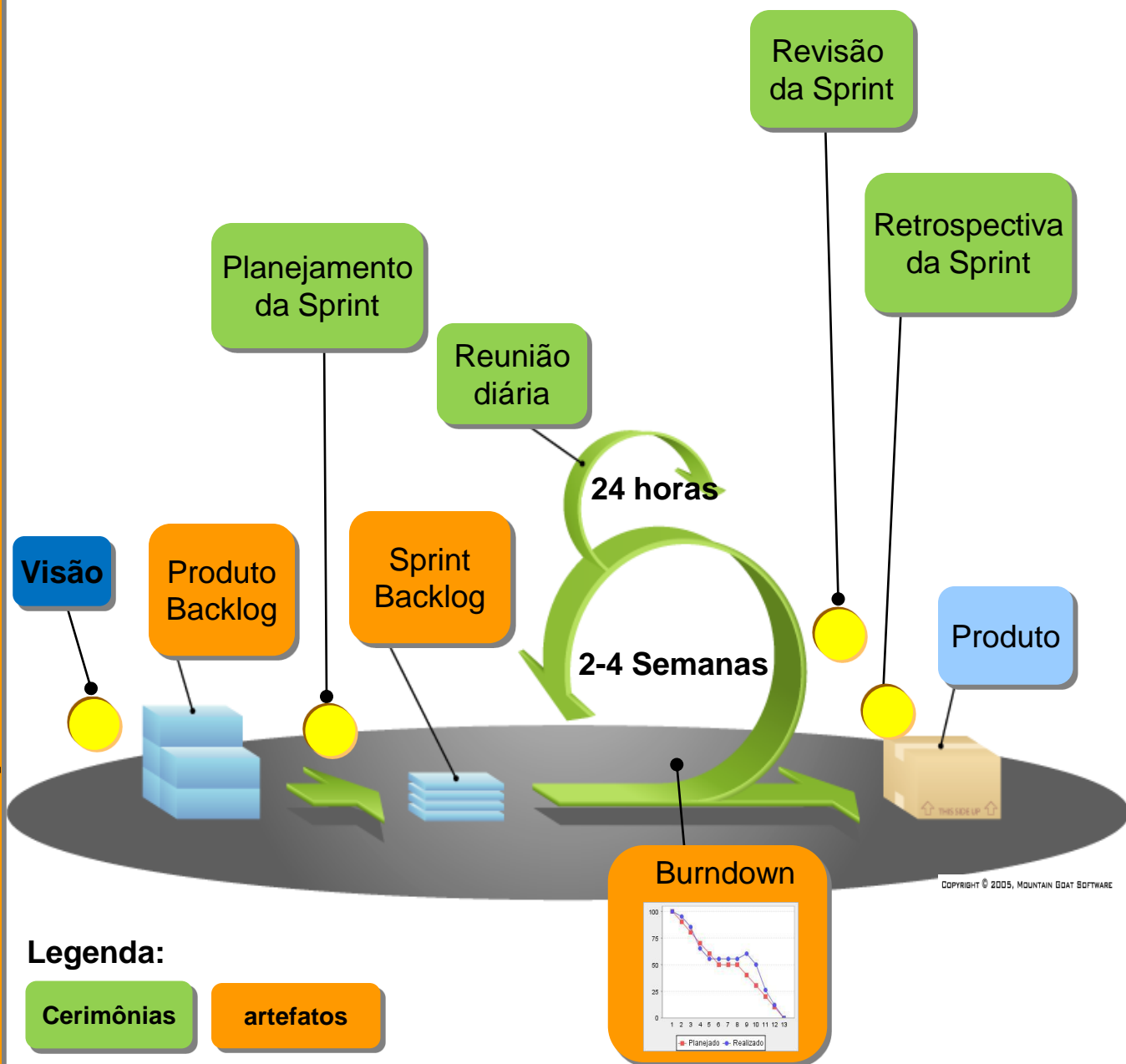
LOCAWEB

Ci&T  
CMMI5 and beyond

C.E.S.A.R



# A ALMA do SCRUM:



## Papéis

- Product Owner (PO)
- ScrumMaster (SM)
- Equipe Scrum

## Cerimônias

- Planejamento da Sprint
- Reunião Diária
- Revisão da Sprint
- Retrospectiva da Sprint

## Artefatos

- Product Backlog
- Sprint Backlog
- Burndown (gráfico)

# Papéis SCRUM:

O SCRUM tem somente três papéis: Product Owner (PO), SCRUM Master (SM) e a equipe SCRUM.



**Product Owner**, responsável por:

- Definir a Visão do Produto
- Elaborar e manter o Product Backlog;
- Definir a prioridade e ROI;
- Representar o cliente;
- Aceitar ou rejeitar os entregáveis.



**SCRUM Master** é responsável por:

- Ser um líder (servidor);
- Remover impedimentos;
- Proteger a equipe;
- Ajudar o PO (*com Product Backlog*);
- Ser o facilitador da equipe;
- Garantir as práticas SCRUM.



**Equipe SCRUM** é responsável por:

- Fazer estimativa;
  - Definir as tarefas;
  - Desenvolver o produto;
  - Garantir a qualidade do produto;
  - Apresentar o produto ao cliente
- Equipe: auto-gerenciável e multifuncional*

# A Equipe e Comprometimento:



By Clark & Vizdos

© 2006 implementingscrum.com

## Envolvidos

**Stakeholders**  
(clientes e usuários finais)



## Comprometidos



Equipe



Product Owner



SCRUM Master

A equipe Scrum é formada por pessoas “comprometidas” em realizar as tarefas da Sprint Backlog. As pessoas da equipe deverão possuir habilidades suficientes para desenvolver, testar, criar/desenhar interfaces gráficas e etc, ou seja, tudo que é realmente necessário para entregar o software funcionando.

O **tamanho da equipe** é algo muito importante, pois, o SCRUM recomenda que o tamanho da equipe seja entre 6 a 9 pessoas. Entretanto, podemos ter equipe menores com 4 pessoas, por exemplo.

Acertar o tamanho da equipe é um **Fator Crítico de Sucesso**. Às vezes é necessário reduzir o tamanho de uma equipe para aumentar a produtividade. *(Ooops, isto parece antagônico, mas geralmente as equipes menores são aquelas que produzem os melhores resultados)*

A equipe também possui diversas responsabilidades, entre elas a de participar das **cerimônias** (que são reuniões que acontecem em momentos distintos da Sprint).



# Cerimônias:

## Reunião de Planejamento da Sprint (8 horas)

Participantes: PO, Equipe e SCRUM Master

Esta reunião é primeira reunião, seu objetivo é fazer o planejamento da Sprint. Ela é dividida em duas partes. Na primeira parte o PO definirá prioridade, seleção dos itens do backlog e a meta da Sprint.

Na segunda parte a equipe definirá a Sprint Backlog (quais são as tarefas necessárias para cumprir a meta).



## Reunião Diária (15 minutos)

Participante: Equipe e SCRUM Master

Nesta reunião somente membros da equipe devem participar. A duração dela é de 15 minutos. As pessoas fazem a reunião de pé. O objetivo desta reunião é fazer que as pessoas respondam a **3 questões**:

- O que eu fiz ontem ?
- O que irei fazer hoje ?
- Encontrei algum impedimento ?



## Revisão da Sprint (4 horas\*)

Participantes: PO, Equipe e SCRUM Master e Convidados\*

Esta reunião acontece no final da Sprint, outras pessoas podem ser convidadas (se necessário).

O objetivo da reunião é apresentar o que a equipe fez durante a Sprint e fazer a entrega do produto (software funcionando) para o PO. *(Geralmente é apresentado uma demo do software).*

Ela deverá ser feita em um auditório ou em uma sala de reunião.



## Retrospectiva da Sprint (3 horas\*)

Participantes: Equipe e SCRUM Master

Esta reunião acontece logo após a Revisão da Sprint.

O objetivo dela é avaliar o que deu certo e que deu errado durante a Sprint e fazer os ajustes possíveis para a próxima Sprint, ou seja, o ciclo de melhoria contínua.



*Nota: \* A quantidade de horas pode variar de acordo com a necessidade (por exemplo, apresentação daquilo que será entregue ao cliente) neste caso a Retrospectiva da Sprint poderá variar entre 1 a 3 horas*

# Artefato: Product Backlog

**Product Backlog** é uma lista contendo todas as funcionalidades desejadas para um produto.

## Exemplo de Product Backlog: Sistema de Reserva On-Line

Nível de Prioridade	Categoria	Descrição do Item
A	Reserva	Os clientes poderão fazer reserva de apartamento
A	Reserva	Os clientes poderão cancelar a reserva
A	Reserva	Os clientes poderão fazer alterações de data da reserva
A	Reserva	Os cliente poderão fazer consulta de reservas
M	Reserva	Criação do Book de Reserva
A	Pagamento	O meio de pagamento da reserva será cartão de crédito
B	Apartamento	Os apartamentos deverão ser cadastros
B	Apartamento	Os apartamentos são classificados por categoria
A	Cliente	Precisamos registrar os dados dos clientes

Níveis de Prioridade: A- alto, M-médio e B-baixo



**Product Owner (PO), é responsável por elaborar e manter Product Backlog atualizado, bem como priorizar seus itens.**



### Product Owner

# Estória do Usuário (User Story):

**O que é uma estória do usuário (user story) ?**  
É uma pequena descrição, que detalha um item do Product Backlog.

## Para que serve a Estória:

Uma estória ajuda no entendimento e também é, utilizada como lembrete para as atividades de planejamento. Ela também permite fazer a estimativa de velocidade da equipe.

Usualmente a estimativa é feita em pontos (story points) ou horas/dias (dias ideais).

## Como escrever uma boa estória:

Conversa sobre a estória, entre os desenvolvedores e os clientes, de modo a detalhar o item e esclarecer todas as dúvidas sobre o que deve ser feito.



## Exemplos de Estórias de Usuário:

	<b>Título: Pagamento com Cartão de Crédito</b>	<b>Prioridade: 1-Alta</b>
	<i>Os clientes (hospedes) poderão fazer o pagamento da estadia e dos serviços com Cartão de Crédito.</i>	
	<i>Os cartão aceitos são: Visa, Master e Amex.</i>	

	<b>Título: Exibir preço do produto</b>	<b>Prioridade: 3-Baixa</b>
	<i>Quando um cliente "passar" um produto pelo leitor do scanner e o código de barra (código do produto) for válido o sistema deverá buscar o preço do produto e exibi-lo na tela do scanner</i>	

	<b>Título: Autenticação de usuário</b>	<b>Prioridade: 2-Média</b>
	<i>O usuário deverá fornecer uma identificação válida (e-mail e senha), o sistema deverá autenticá-lo de acordo com "User Profile". Se autenticação for positiva o usuário deverá receber a autorização de acesso.</i>	
	<i>Caso contrário receberá uma mensagem de identificação inválida</i>	



# Estimativa\* e o Planning Poker:

Para fazer estimativa de velocidade da equipe, antes é preciso o escrever as histórias de usuário.

O **Planning Poker** é a “prática” que ajuda na **estimativa** de uma **estória** ou de uma **tarefa**.



Geralmente o Planning Poker usa uma escala de pontos não linear, que pode ser baseada no Fibonacci: (1,2,3,5,8,13,...) + 20, 40, 100 ou em outra escala.

## Jogando o Planning Poker:

Antes de começar o jogo, ou seja, definir os pontos para cada histórias, é importante definir um **valor de referência**. Exemplo: Identificar a história que pode ser atribuído o menor valor (**dois pontos**, por exemplo), esta história será utilizada como referência para pontuação das demais histórias.

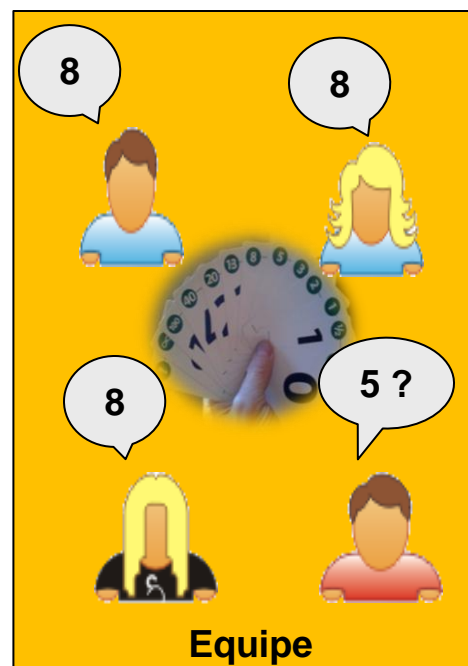
Pessoal, qual estimativa para essa estória...



Product Owner



Equipe



Equipe

Na reunião de **Planejamento da Sprint**, a equipe joga o **Planning Poker** e define as estimativas.

### Nota 1 – Estimativa\*

Para fazer a estimativa, você deve levar em consideração diversos aspectos além da codificação, como por exemplo: testes de aceitação, teste unitários, preparação de ambiente e entre outras coisas que são necessárias e importantes (mesmo que de baixo valor agregado) para que você entregue : software funcionando.

# Artefato: Sprint Backlog

O **Sprint Backlog** é uma lista de tarefas que equipe se compromete a fazer durante a **Sprint**. A **Sprint Backlog** é elaborada na segunda parte da reunião de Planejamento da Sprint.

Para atingir a meta da Sprint a equipe deverá fazer as tarefas da Sprint Backlog.

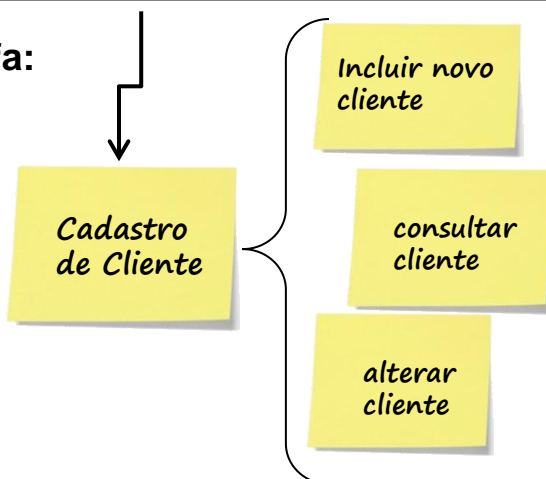
## Selected Product Backlog (item selecionado do Product Backlog)

Departamento	Os apartamentos são classificados por categoria	5
Cliente	Precisamos registrar os dados dos clientes	8

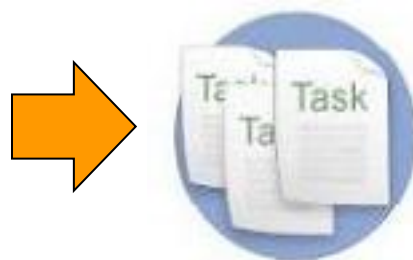
Estória de Usuário:

	<i>Titulo: Precisamos registrar os dados dos clientes</i>	<i>Prioridade: 1-Alta</i>
	<i>Todos os dados do cliente deverá ser registrado. A busca de cliente deverá ser fácil e intuitiva.</i>	
	<i>Quando os clientes estão registrado, será possível alterar os dados se necessário.</i>	
	<i>O cliente deverá ter um "status" para que se possa definir quais são os clientes ativos e os inativos</i>	
	<i>Pontos: 8</i>	

Tarefa:



Sprint Backlog



### Dicas para "montar" um bom Sprint Backlog:

- 1 – Toda a equipe deve participar da elaboração do Sprint Backlog;
- 2 – Faça a definição do "Pronto" (DoD), ou seja, quando o produto está pronto de fato;
- 3 – Tente identificar todas as tarefas, lembre-se que algumas tarefas são puramente técnicas, por exemplo: Realização de Teste Unitário.
- 4 – Respeite o tempo para realização desta cerimônia, pois a Reunião de Planejamento é um **timebox**.

## Definição de “Pronto” (DoD):

Ao final de cada Sprint a equipe deverá fazer uma entrega valor para o cliente (PO e demais stakeholders).

Segundo Manifesto Ágil, **valor para o cliente é igual a software funcionando.**

Logo, para fazer tal entrega, na reunião de **Planejamento da Sprint**, será imprescindível estabelecer a “**Definição de Pronto**”.

Isto evitará problemas e frustrações futuras nas reuniões de Revisão da Sprint e na Retrospectiva da Sprint.

**Definir claramente quando o produto estará “pronto”:**

**Pronto, para desenvolvedor:**

- Encerrou a codificação...

**Pronto, para Analista de Teste (Q&A):**

- Quando ele encerrou o teste e não encontrou nenhum bug...

**Pronto, para PO:**

- Quando foi entregue...

**Pronto, para os usuários finais e/ou clientes:**

- Quando o software começou a funcionar em ambiente de produção...

**“Pronto” é quando o produto está feito de fato para o cliente;**



**Evite: A síndrome dos 90% prontos.**



# Artefato: Burndown

O gráfico **Burndown** é a principal ferramenta de **gerenciamento do processo de desenvolvimento de software**.

Pois, ele representa o **trabalho restante sobre tempo**, ou seja, permite visualizar o progresso e/ou a evolução do trabalho executado pela a equipe e a quantidade trabalho x tempo (pontos) que ainda faltam para completar a Sprint.

Atualização do Burndown deve ser diária, isto facilita a tomada de decisão, pois, poderemos decidir em melhorar a produtividade da equipe e/ou para mitigar risco da Sprint.

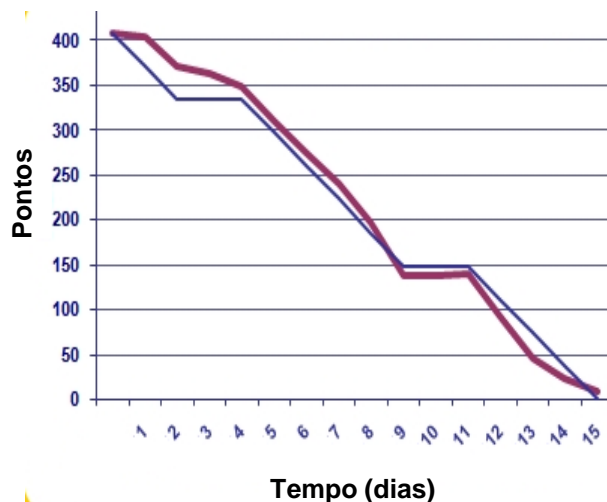
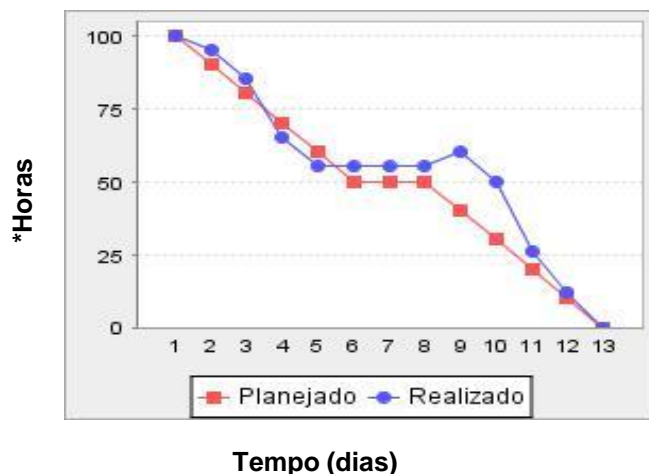
## Exemplo:

Através da leitura do Burndown poderemos decidir, que devemos adicionar novas tarefas na Sprint (velocidade da equipe está acima do planejado, melhorando sua produtividade) ou retirar tarefas (a velocidade da equipe está abaixo do planejado, caso não seja feita uma redução de tarefas a meta da Sprint estará comprometida).

O ideal, neste caso, é retirar as tarefas que não afetem a meta da Sprint. Se a meta for afetada pode-se também decidir pelo cancelamento da Sprint.



## Exemplos de Burndown:

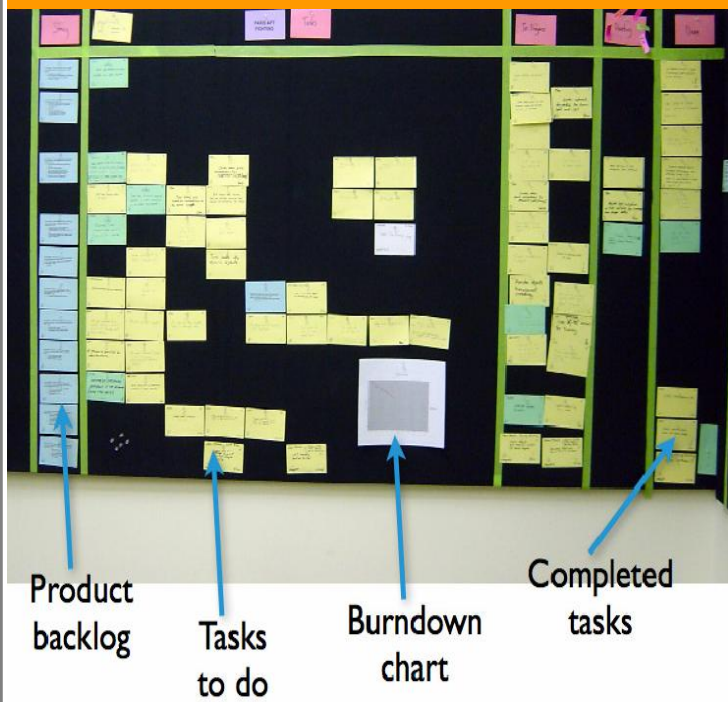


**Burndown demonstra a produtividade da equipe**

\* **Dias Ideais (Ideal Days)**, é uma outra forma de fazer estimativa, ele é baseado na duração de tarefas.  
 - **Dias ou horas** é unidade bem definida, contudo o "tempo ideal" quase nunca é igual ao "tempo real"...

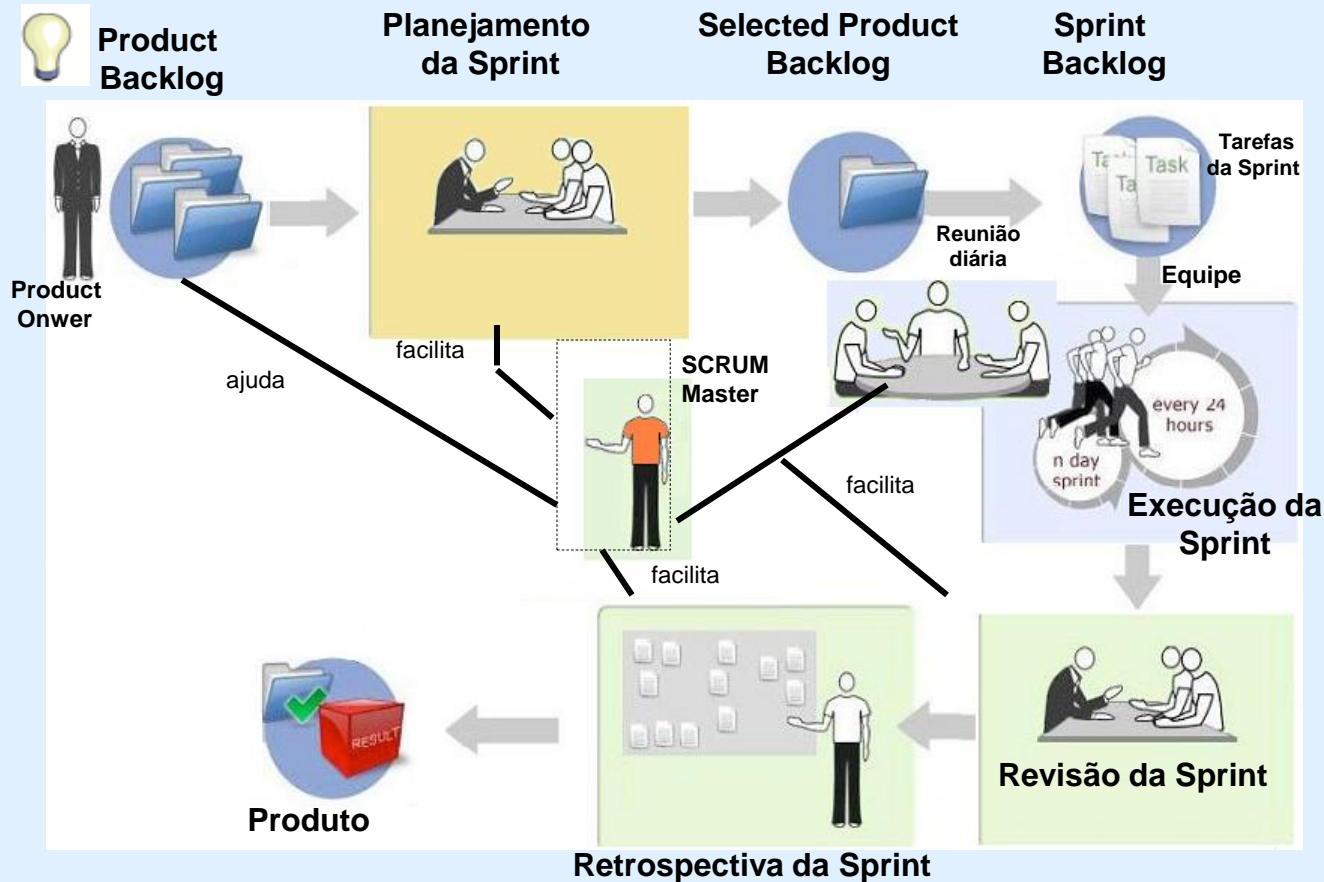
# Gestão à Vista: Task Board

**Gestão à Vista: Dá visibilidade e transparência ao projeto de desenvolvimento de software**



# Road Map: SCRUM

Visão do Produto







# Estudo de Caso

*baseado em fatos reais*



# Product Backlog: Sistema de Reserva On-Line

Nível de Prioridade	Categoria	Descrição do Item
A	Reserva	Os clientes poderão fazer reserva de apartamento
A	Reserva	Os clientes poderão cancelar a reserva
A	Reserva	Os clientes poderão fazer alterações de data da reserva
A	Reserva	Os cliente poderão fazer consulta de reservas
M	Reserva	Criação do Book de Reserva
A	Pagamento	O meio de pagamento da reserva será cartão de crédito
B	Apartamento	Os apartamentos deverão ser cadastros
B	Apartamento	Os apartamentos são classificados por categoria
A	Cliente	Precisamos registrar os dados dos clientes

Níveis de Prioridade: A- alto, M-médio e B-baixo



**Product Owner** define os itens do **Product Backlog** e o nível de **prioridade** de cada item.



**Scrum Master** deve ajudar o **Product Owner** na elaboração do **Product Backlog**.

# Product Backlog: Sistema de Reserva On-Line

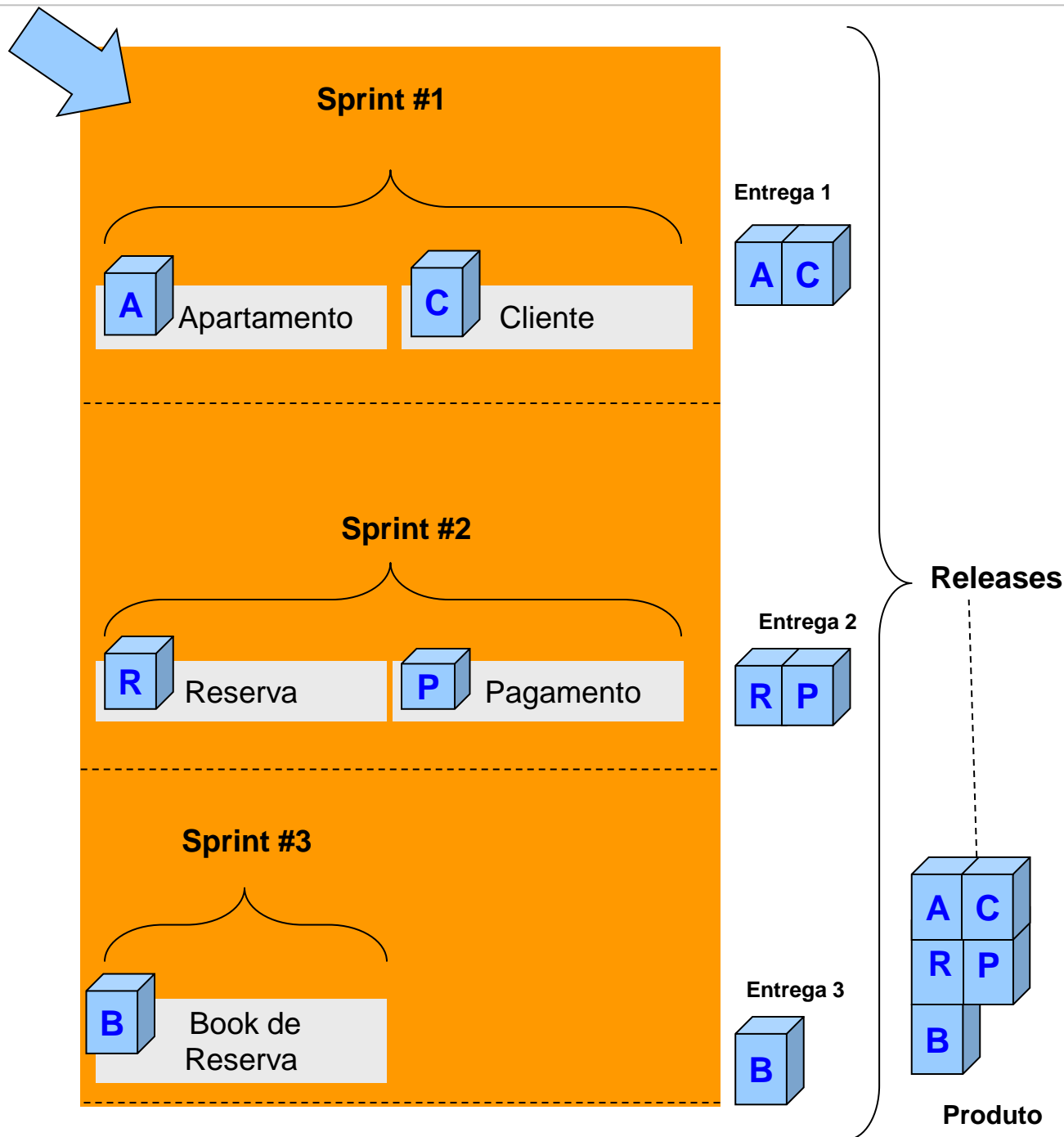
Nível de Prioridade	Categoria	Descrição do Item
A	Reserva	Os clientes poderão fazer reserva de apartamento
A	Reserva	Os clientes poderão cancelar a reserva
A	Reserva	Os clientes poderão fazer alterações de data da reserva
A	Reserva	Os cliente poderão fazer consulta de reservas
M	Reserva	Criação do Book de Reserva
A	Pagamento	O meio de pagamento da reserva será cartão de crédito
B	Apartamento	Os apartamentos deverão ser cadastros
B	Apartamento	Os apartamentos são classificados por categoria
A	Cliente	Precisamos registrar os dados dos clientes



O PO e Scrum Master analisam o **Product Backlog** e tomam a decisão de mudar os níveis de prioridade dos itens do **Product Backlog**, com o intuito de facilitar o plano de construção do software.

A mudança do nível de prioridade dos itens do **Product Backlog** também vai ajudar na definição do **Plano de Release** e suas iterações (**Sprints**) e no **Plano de Construção (Build Plan)** do software.

# Plano de Release (Release Plan)



O PO deve elaborar e validar o Plano de Release (Release Plan).

# Reunião de Planejamento da Sprint

## Product Backlog: Sistema de Reserva On-Line

Nível de Prioridade	Categoria	Descrição do Item Backlog	Estimativa em pontos
M	Reserva	Os clientes poderão fazer reserva de apartamento	-
M	Reserva	Os clientes poderão cancelar a reserva	-
M	Reserva	Os clientes poderão fazer alterações de data da reserva	-
M	Reserva	Os cliente poderão fazer consulta de reservas	-
B	Reserva	Criação do Book de Reserva	-
M	Pagamento	O meio de pagamento da reserva será cartão de crédito	-
A	Apartamento	Os apartamentos deverão ser cadastros	-
A	Apartamento	Os apartamentos são classificados por categoria	-
A	Cliente	Precisamos registrar os dados dos clientes	-



### Reunião de Planejamento da Sprint (1a. Parte): Participantes: PO, Equipe e SCRUM Master (facilitador)

*Se for a primeira reunião o PO deverá apresentar a visão do produto, expectativas e prioridades.*

Nesta reunião, PO deverá definir uma **meta** para **Sprint** e falar quais são os **itens** são mais **prioritários** do Product Backlog.

A equipe realizará o planejamento do que deverá ser entregue no final da Sprint (de 2 a 4 semanas).

A equipe deverá selecionar quais os **itens serão feitos** na **Sprint**, resultando no **Selected Product Backlog**. ✓



# Reunião de Planejamento da Sprint

## Product Backlog: Sistema de Reserva On-Line

Nível de Prioridade	Categoria	Descrição do Item Backlog	Estimativa em pontos
2	Reserva	Os clientes poderão fazer reserva de apartamento	-
2	Reserva	Os clientes poderão cancelar a reserva	-
2	Reserva	Os clientes poderão fazer alterações de data da reserva	-
2	Reserva	Os cliente poderão fazer consulta de reservas	-
3	Reserva	Criação de o Book de Reserva	-
2	Pagamento	O meio de pagamento da reserva serão por cartão de crédito	-
1	Apartamento	Os apartamentos deverão ser cadastros	8
1	Apartamento	Os apartamentos são classificados por categoria	7
1	Cliente	Precisamos registrar os dados dos clientes	8

### Continuação (da 1ª. parte da reunião)

A equipe deverá se preocupar em levantar mais detalhes dos itens selecionados do **Selected Product Backlog**, escrever as **estórias** podem ser uma técnica útil para melhorar entendimento dos itens selecionados (a).

Para estimar a velocidade da equipe, que é necessária para implementar os itens selecionados e duração da Sprint, será utilizadas as estórias para fazer as **estimativas em pontos** (ou horas/dias), através do **Planning Poker**. (b)

### Reunião de Planejamento da Sprint: (2a. Parte)

**Participante:** Equipe (e SCRUM Master - opcional)

É onde as estórias serão divididas em **tarefas**, gerando o **Sprint Backlog**. (c)

**Última coisa a ser feita nesta reunião é Decidir quem executará as Tarefas:** Cada pessoa da equipe deve escolher **as tarefas da Sprint Backlog** que deseja fazer.

Legenda:  
(a) pág: 31  
(b) pág: 31  
(c) pág: 32

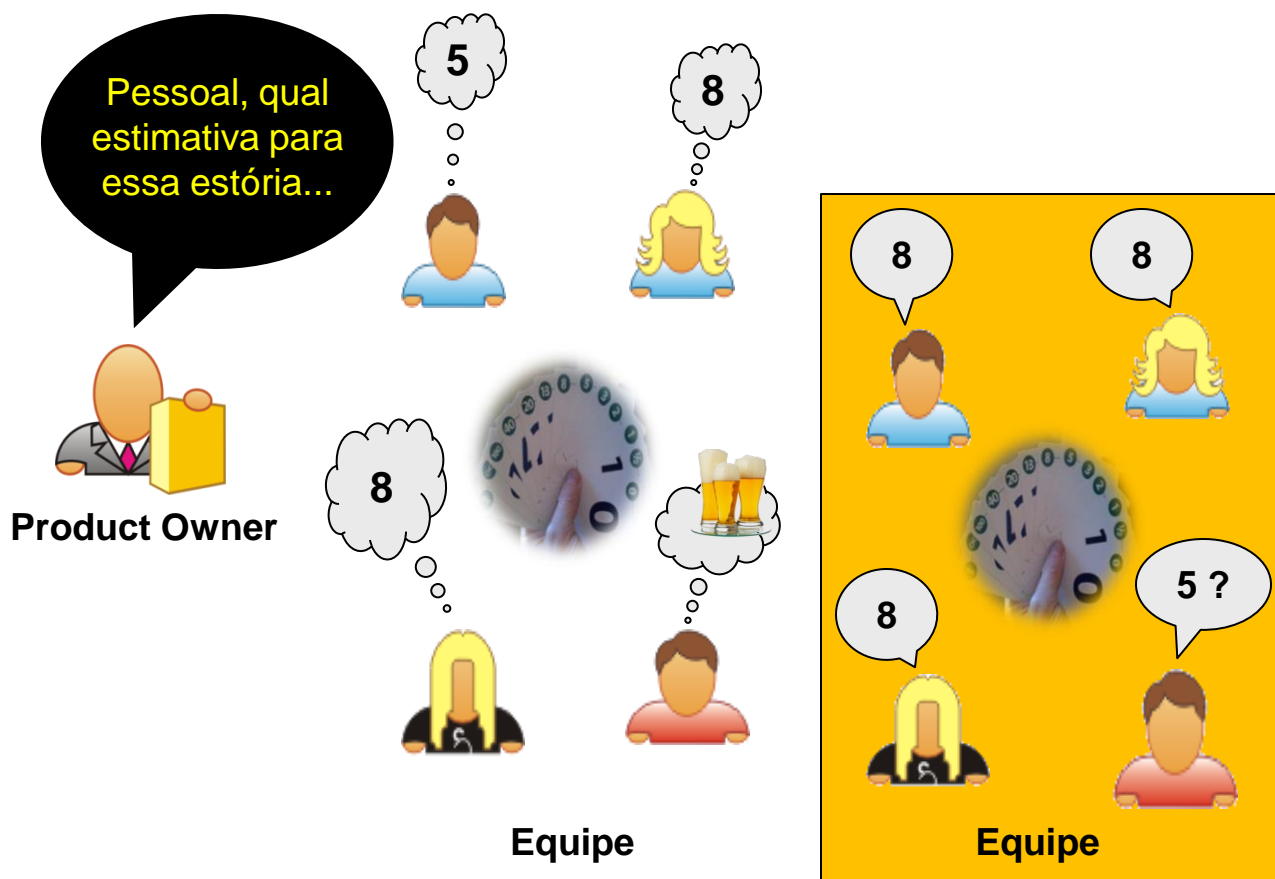


✓ Itens selecionados

# Fazendo Estimativa com Planning Poker:

## Estória do Usuário:

	<i>Titulo: Precisamos registrar os dados dos clientes</i>	<i>Prioridade: 1-Alta</i>
	<i>Todos os dados do cliente deverá ser registrado. A busca de cliente deverá ser fácil e intuitiva.</i>	
	<i>Quando os clientes estão registrado, será possível alterar os dados se necessário.</i>	
	<i>O cliente deverá ter um "status" para que se possa definir quais são os clientes ativos e os inativos</i>	



Na reunião de **Planejamento da Sprint**, a equipe joga o **Planning Poker** e define a estimativa de velocidade da equipe necessária para implementar as estórias (na verdade as tarefas)..

# Tarefas, quebrando a Estória...

As histórias são divididas (quebradas) em tarefas.

As tarefas devem compor a “Sprint Backlog”...

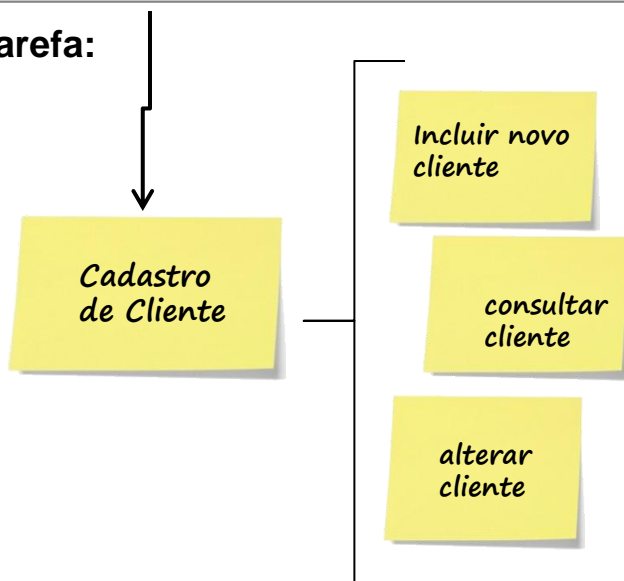
## Selected Product Backlog (item selecionado do Product Backlog)

Cliente	Precisamos registrar os dados dos clientes	8
---------	--	---

### Estória do Usuário:

	<i>Titulo: Precisamos registrar os dados dos clientes</i>	<i>Prioridade: 1-Alta</i>
	<i>Todos os dados do cliente deverá ser registrado. A busca de cliente deverá ser fácil e intuitiva.</i>	
	<i>Quando os clientes estão registrado, será possível alterar os dados se necessário.</i>	
	<i>O cliente deverá ter um “status” para que se possa definir quais são os clientes ativos e os inativos</i>	
	<i>Pontos: 8</i>	

### Tarefa:



### Sprint Backlog



# Check List do Planejamento da Sprint:

## Primeira parte da reunião:

- 1.1 – A visão do produto foi completamente entendida;
- 1.2 – Os níveis de Prioridade dos itens do Product Backlog foi definido;
- 1.3 – Os itens do Product Backlog foram selecionados;
- 1.4 – A meta da Sprint (o que deve ser entregue no final da Sprint) foi estabelecida ;

## Segunda parte da reunião:

- 2.1 – Os itens são detalhados através da escrita de histórias de usuário;
- 2.2 – Estimativa em Pontos é definida. (as histórias de usuário são utilizadas para fazer as estimadas;
- 2.3 - As histórias de usuário são quebradas em tarefas;
- 2.4 - Sprint Backlog é definido;
- 2.5 – As pessoas da equipe definem entre elas quem ira fazer as tarefas do Sprint Backlog.

## Outros itens (fora da reunião do planejamento, mas necessários para começar uma Sprint):

- 3.1- Preparar o “**Task Board**” quadro de tarefas (também chamado de quadro de Kanban)
- 3.2 - Preparar o gráfico “**Burndown**”
- 3.3 - Fazer o Kick-off (Sprint #0)

A close-up photograph of a checklist. The word 'CHECKLIST' is printed in a bold, sans-serif font at the top. Below it, there is a vertical list of checkboxes. A black pen with a gold-colored tip is positioned over the list, with its tip touching one of the checkboxes. The checkboxes are arranged in a slightly curved line, and some are already marked with checkmarks.



# Task Board: Sprint #1 - Dia 0:

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

**Nota:**

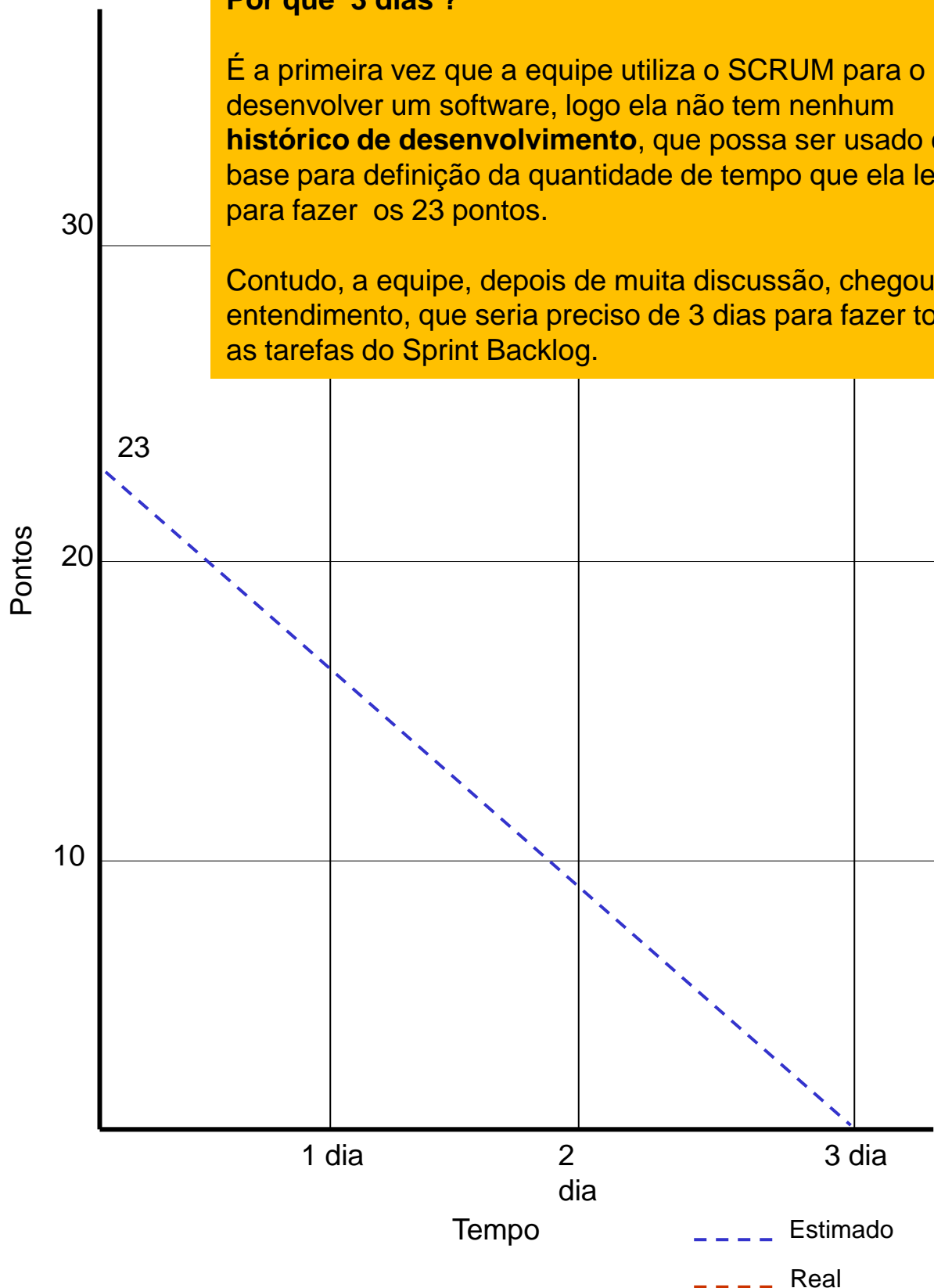
Optamos por apresentar somente as atividades e não as tarefas, somente por questão de facilitar a apresentação.

# Burndown. Sprint #1 - Dia 0:

## Por que 3 dias ?

É a primeira vez que a equipe utiliza o SCRUM para o desenvolver um software, logo ela não tem nenhum **histórico de desenvolvimento**, que possa ser usado como base para definição da quantidade de tempo que ela levará para fazer os 23 pontos.

Contudo, a equipe, depois de muita discussão, chegou ao entendimento, que seria preciso de 3 dias para fazer todas as tarefas do Sprint Backlog.



# [Kick-off] Sprint #1 - Dia 0:

## Sprint Backlog

Cadastro de  
Categoria de  
Apartamentos

Cadastro de  
Apartamentos

Cadastro de  
Clientes



SCRUM Master



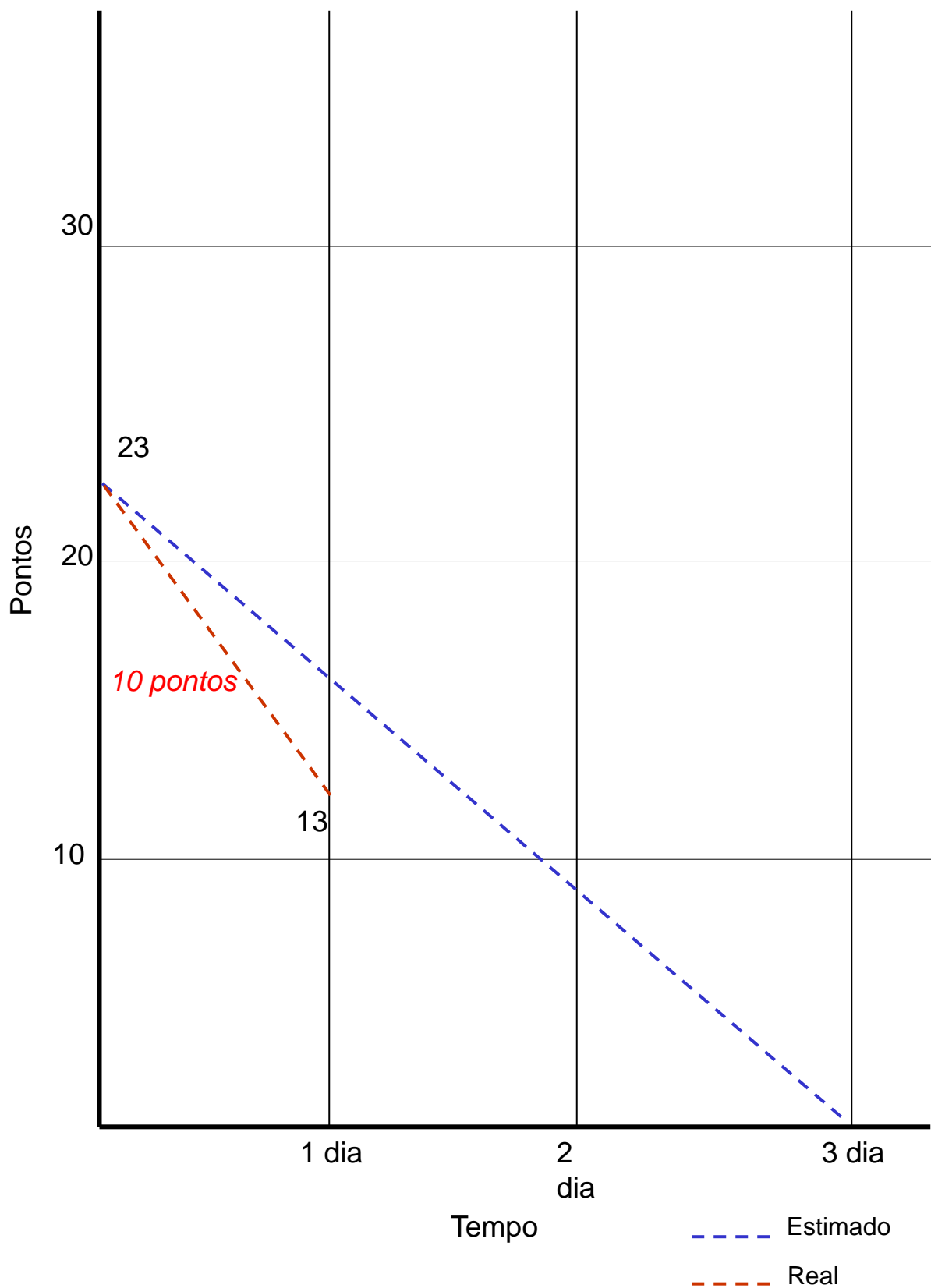
Equipe

## Task Board da Sprint #1: Dia 1 (após o 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>		



## Burndown da Sprint: #1 – Final do Dia 1:



# A Primeira Reunião Diária:

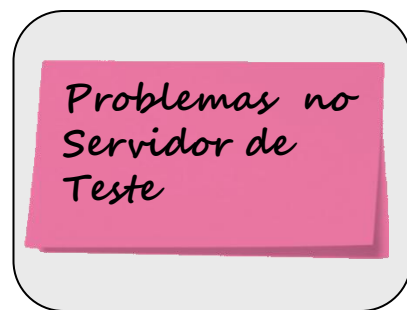
## Sprint Backlog

Cadastro de  
Categoria de  
Apartamentos

OK

Cadastro de  
Apartamentos

Cadastro de  
Clientes



SCRUM Master





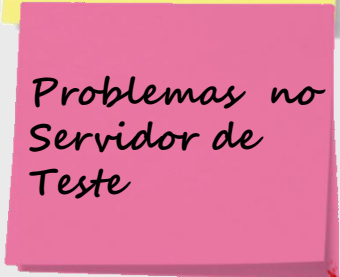

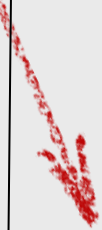
Equipe

**Check List –  
Responder as 3 questões:**


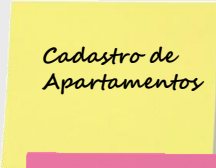
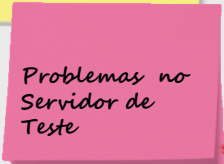

O que foi feito ontem?  
O que você planeja fazer hoje?  
Você tem algum impedimento?

**15  
minutos**

## Task Board da Sprint: #1 – Após primeira reunião

Sprint Backlog	Em Execução	Concluído	BurnDown
 <p>Cadastro de Clientes</p>	 <p>Cadastro de Apartamentos</p>  <p>Problemas no Servidor de Teste</p>	 <p>Cadastro de Categoria de Apartamentos</p>	
		 <p><b>SCRUM Master deverá resolver (remover) este impedimento</b></p>	

# Task Board da Sprint: #1 – Impedimento

Sprint Backlog	Em Execução	Concluído	BurnDown
	 		

SCRUM Master  
deverá resolver  
(remover) este  
impedimento



**SCRUM Master**

Cabe ao **SCRUM Master** remover todos os **impedimentos**, identificados e demonstrados no **Task Board** (quadro de tarefas), para que estes não afetem o desempenho da equipe. Caso contrário, o impedimento poderá comprometer a meta e a entrega de valor que deverá ocorrer no final da Sprint.

Após remoção do impedimento o SCRUM Master deverá “registrar em base de conhecimento” a “causa raiz do impedimento”. Esta informação será útil para melhorar o processo, logo ela será apresentada e discutida na Retrospectiva da Sprint.

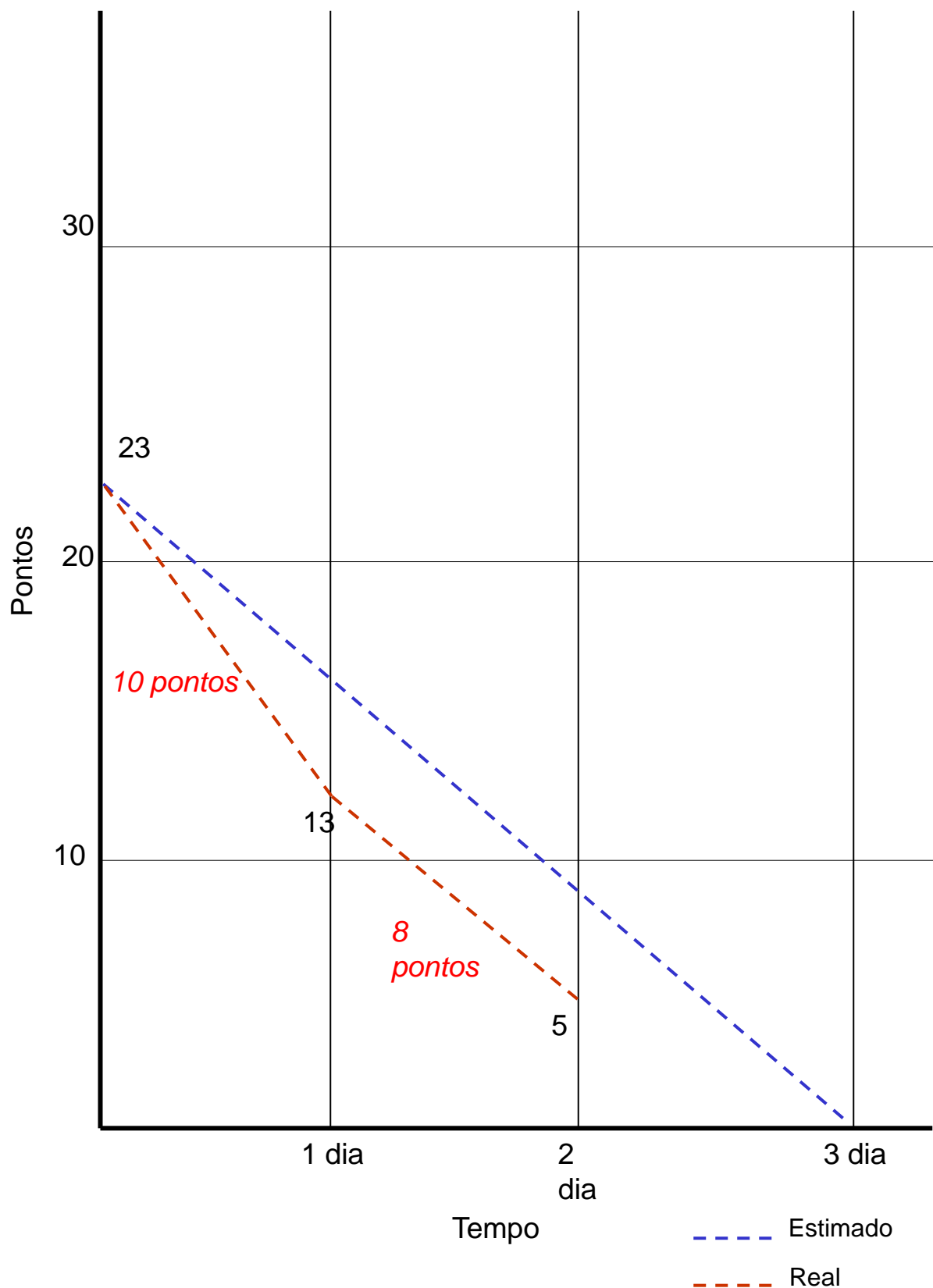
Problemas no  
Servidor de  
Teste

## O que é um impedimento ?

Impedimento tudo aquilo que impede a equipe de realizar seu trabalho e atingir a meta da Sprint.

Um impedimento pode ser um problema de rede, falha no servidor, falta de ambiente para testes, a lentidão do banco de dados ou falta de informação para realização de uma tarefa.

# Burndown da Sprint: #1 – 2º. Dia:





# A Segunda Reunião Diária

## Sprint Backlog

Cadastro de  
Categoria de  
Apartamentos

OK

Cadastro de  
Apartamentos

OK

Cadastro de  
Clientes

Cadastro de  
Apartamentos

OK

Cadastro de  
Clientes



SCRUM Master



Equipe



**Check List –  
Responder as 3 questões:**

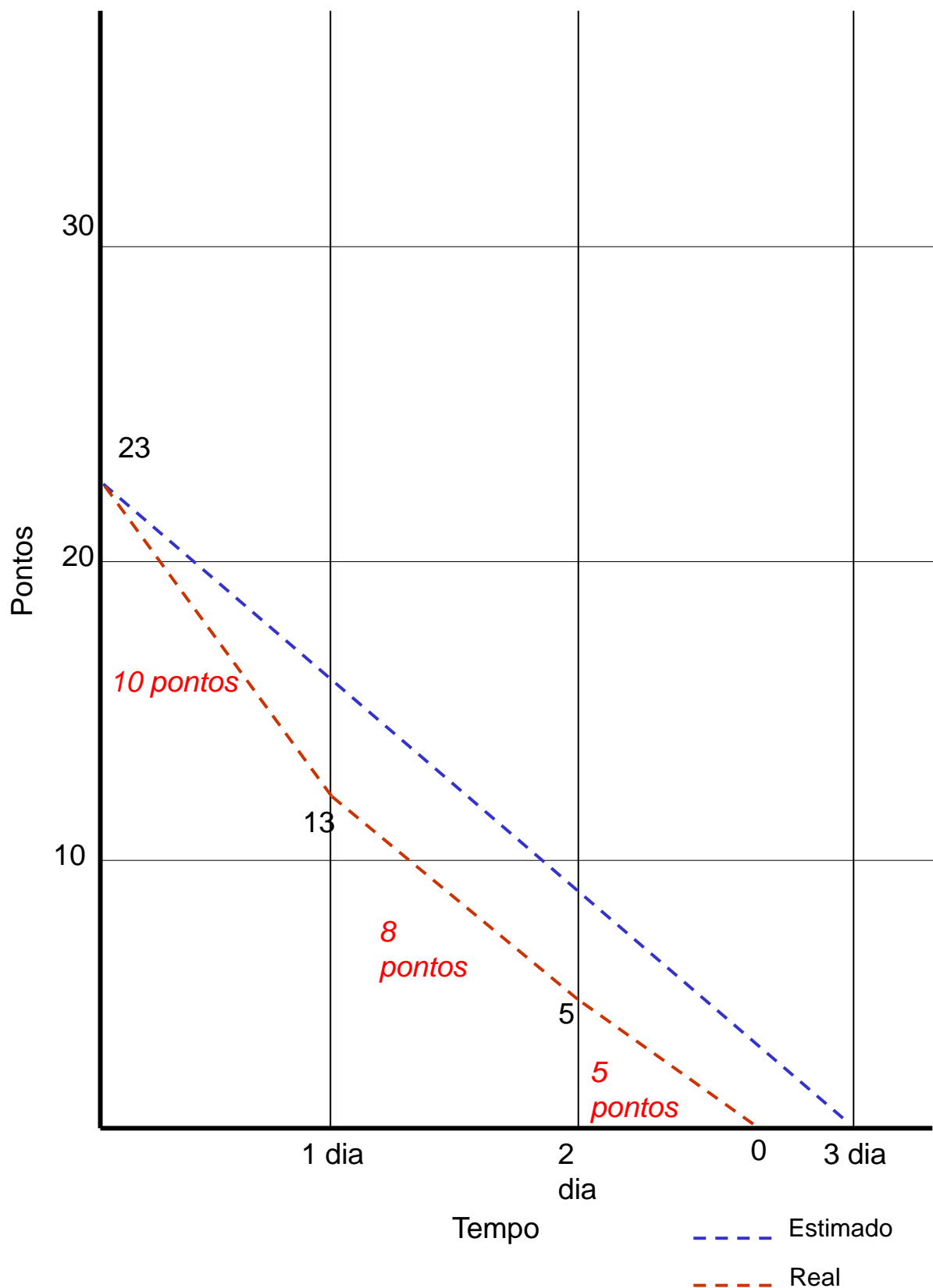
O que foi feito ontem?  
O que você planeja fazer hoje?  
Você tem algum impedimento?

**15  
minutos**

## Task Board da Sprint #1 - 2º. Dia:

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

# Burndown da Sprint #1 - 3º. Dia



# A Terceira Reunião Diária:

## Sprint Backlog

Cadastro de  
Categoria de  
Apartamentos

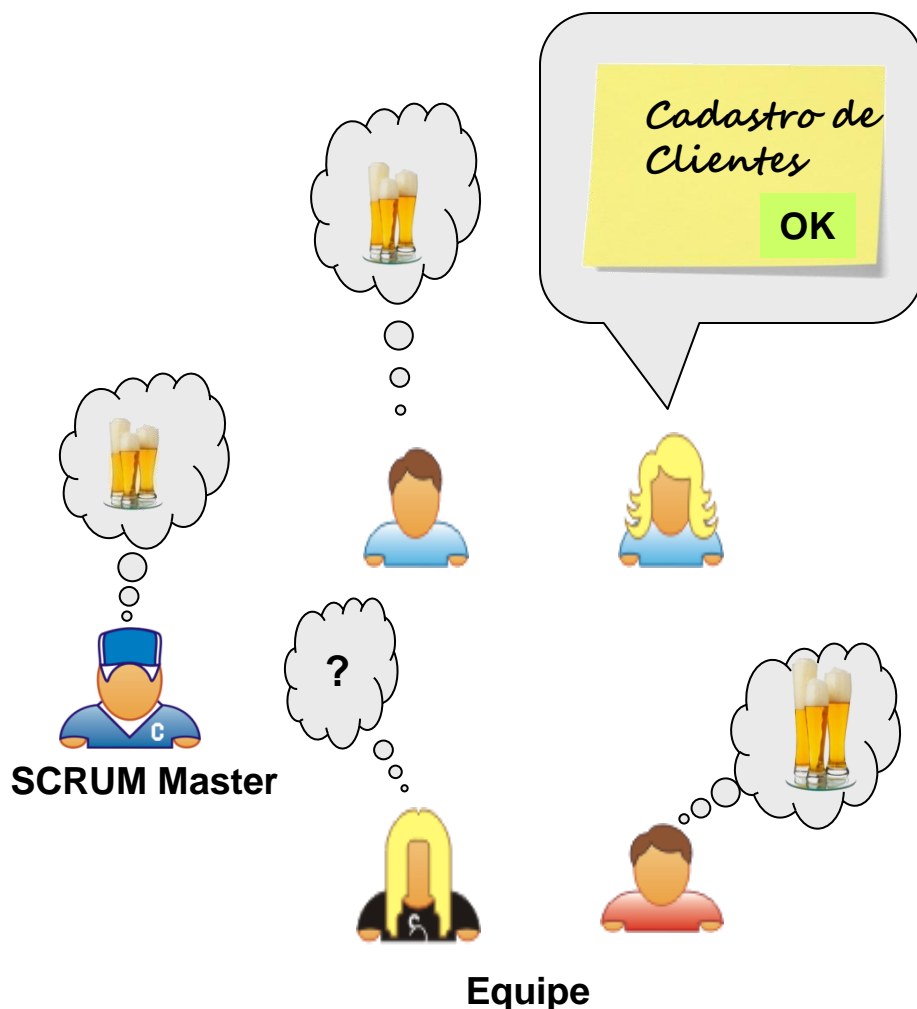
OK

Cadastro de  
Apartamentos

OK

Cadastro de  
Clientes

OK



**Check List –  
Responder as questões:**

O que foi feito ontem?  
Você teve algum impedimento?

**15  
minutos**

## Task Board da Sprint #1 - 3º. Dia:

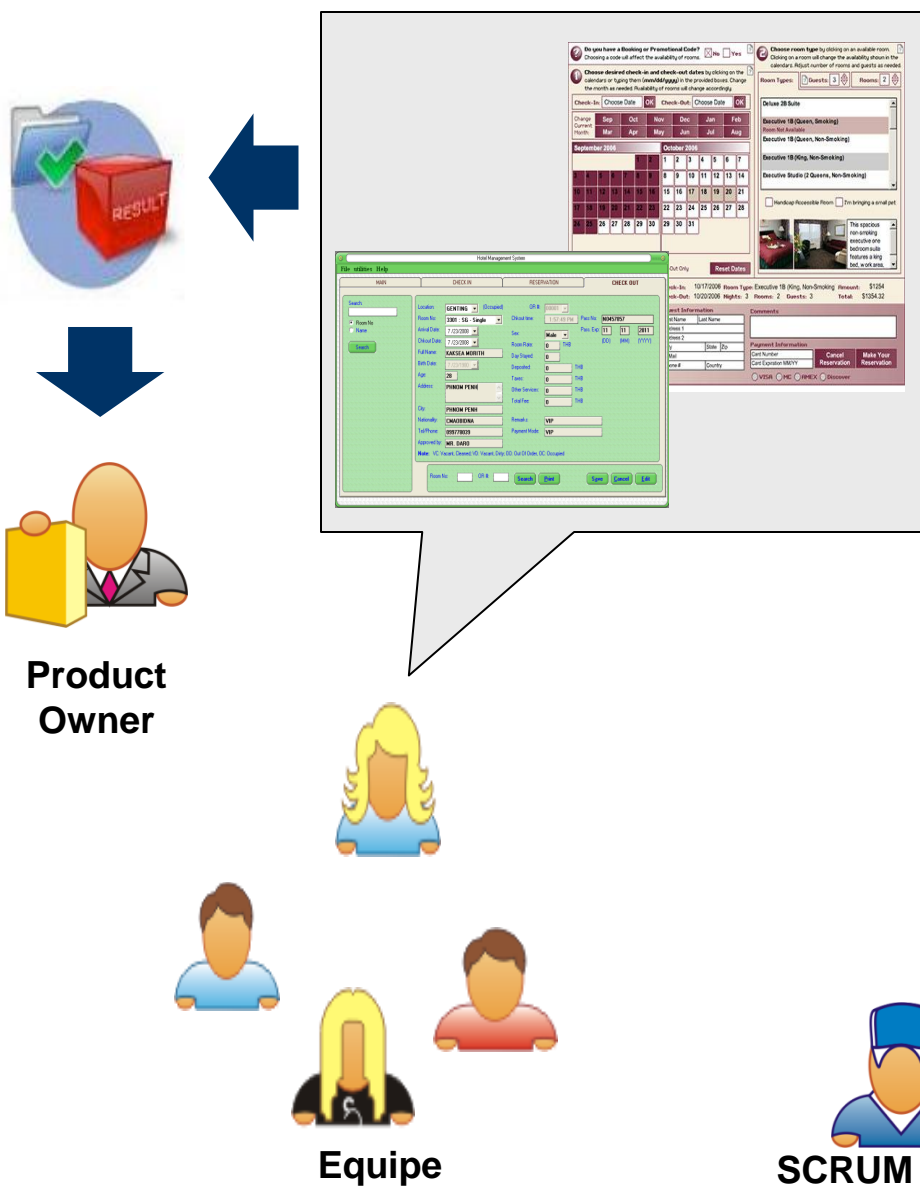
Sprint Backlog	Em Execução	Concluído	BurnDown
		<div>Cadastro de Categoria de Apartamentos</div> <div>Cadastro de Apartamentos</div> <div>Cadastro de Clientes</div>	

Após a conclusão das tarefas (aqui atividades) o próximo passo é fazer a reunião de **Revisão da Sprint**.



# Revisão da Sprint:

## Reunião da Revisão da Sprint



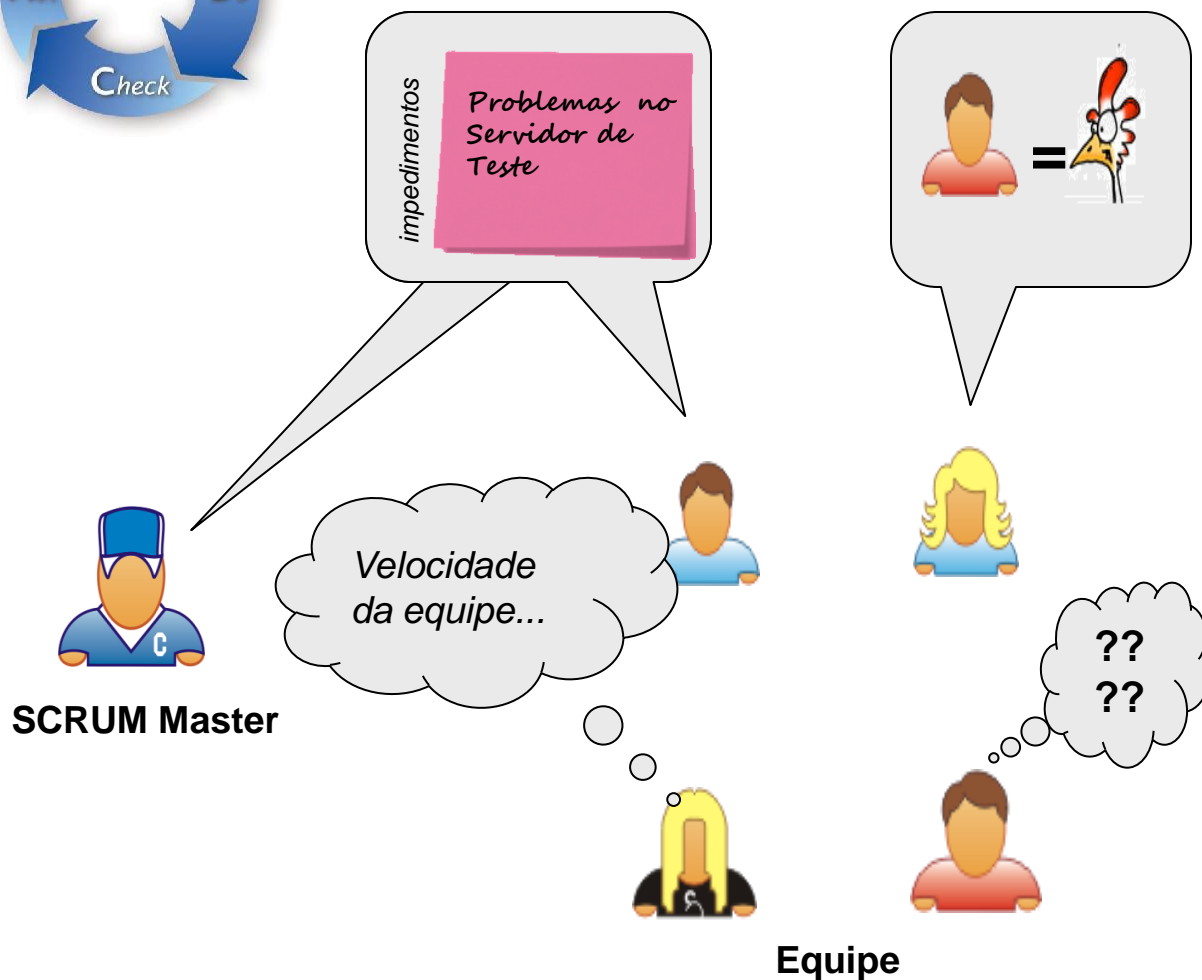
Equipe apresenta que foi produzido e faz entrega para PO, que avaliará o valor da entrega. PO poderá aceitar ou rejeitar a entrega do produto.

# Retrospectiva da Sprint

## Reunião Retrospectiva da Sprint



As retrospectivas são a essência do conceito de Inspeção e Adaptação.



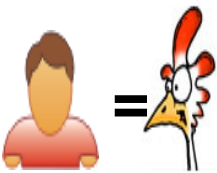


3  
horas

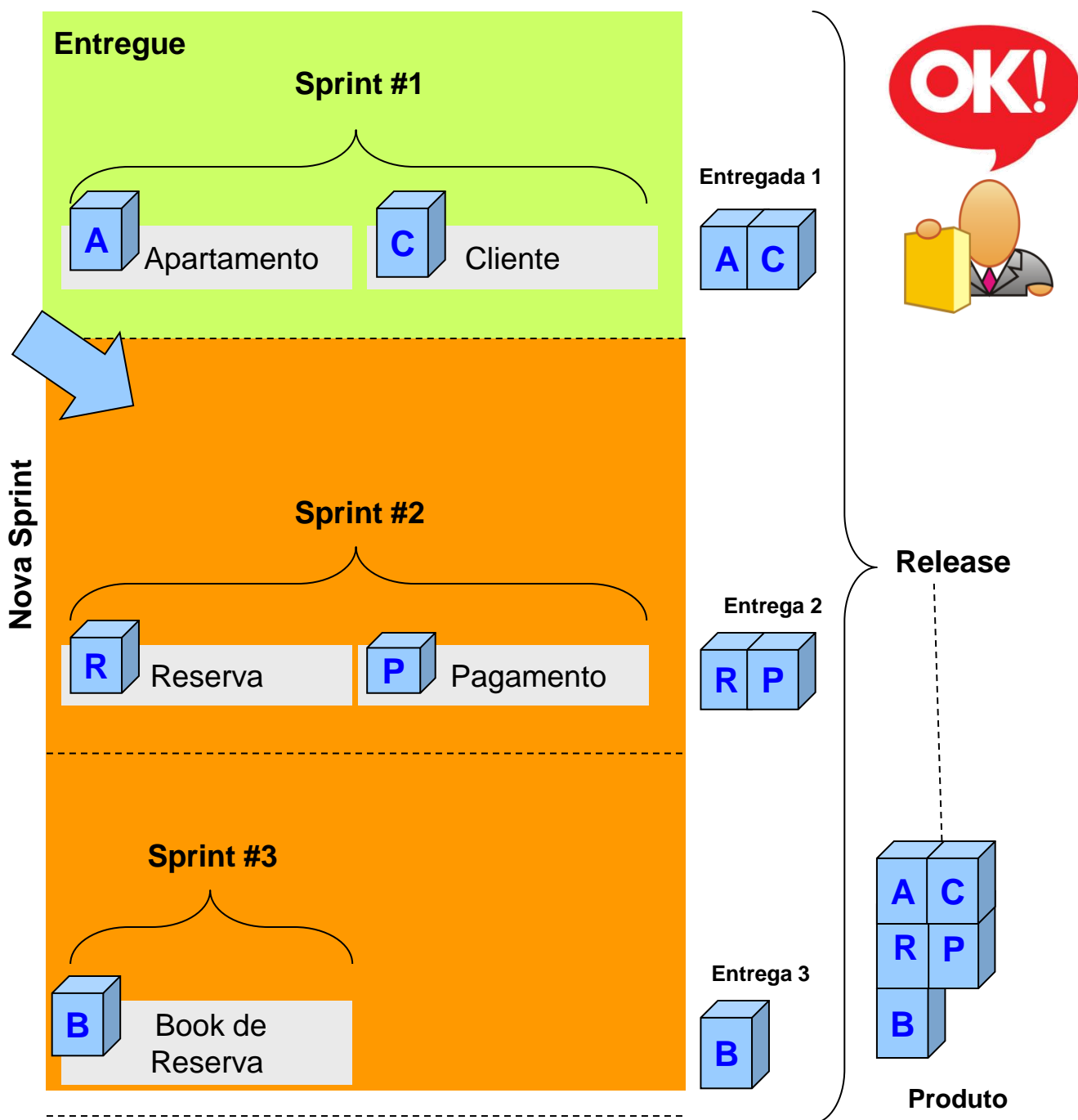
Equipe discute o que deu errado e que deu certo... O que precisará ser melhorado para a próxima Sprint

# Retrospectiva da Sprint

Lições Aprendidas, o que deve melhorado para a próxima Sprint

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

# Plano de Release - Sprint (Sprint #2):



Começar a Sprint # 2, fazer o **Planejamento da Sprint**:

- Definir os itens selecionados do Product Backlog
- ...



# Vendendo a idéia do SCRUM



# Vendendo a idéia do SCRUM:



Agora é discutiremos a parte, que para muitos é considerada a mais difícil e as vezes quase intransponível.

Que é vender a idéia do SCRUM (do uso do SCRUM) para seu chefe, dono da empresa, gerente de TI e etc..

Poderíamos reunir muitos argumentos para fazer esta venda, todavia, acredito que a melhor forma é vender os resultados que podem ser alcançadas com a adoção das práticas SCRUM.

Poderíamos reunir muitos argumentos para fazer esta venda, todavia, acredito que a melhor forma é vender os resultados que podem ser alcançadas com a adoção das práticas SCRUM.

## O que NÃO VENDER:

- Não tente vender o SCRUM (ou qualquer outra coisa) que você não acredite...para vender ou influenciar pessoas, você tem que ter comprada a idéia primeiro.
- Jamais venda aquilo que não se possa entregar.

Não venda o SCRUM, venda os resultados (veja a lista abaixo) que você e sua equipe poderão atingir utilizando o SCRUM.

Não tente vender a idéia do SCRUM por puro modismo (somente por que na empresa dos seus “amigos” estão usando SCRUM e você não quer ficar de fora)..

Não tente vender a idéia sem fatos (veja as quais empresas que já utilizam o SCRUM, obtenha dados destas empresas, por exemplo: se houve aumentou a produtividade, crescimento da motivação das pessoas, se custos reduziram e diminui o prazo de entrega dos produtos e etc..)

## O que os chefes gostam de comprar:

Aumento de produtividade (sem a contratação de pessoas), para muitos chefes isto é sonho encantado;

Redução do prazo de entrega;

Redução de custo;

Busca pela excelência da qualidade, ou seja, melhorar a qualidade dos produtos e/ou serviços continuamente sem a necessidade de comprar novas ferramentas;

Pessoas trabalhando com foco em resultado;

Começar qualquer coisa, sem ter que fazer investimento inicial;

A sensação de controle absoluto (todo chefe gostaria de saber o que seu o pessoal esta fazendo..) .

O Task Board e Product Backlog produzem esta sensação, pois tudo que precisa para controlar está à vista.

Clientes mais satisfeitos

E por fim: A liberdade de fazer mudanças, mesmo que o desenvolvimento esteja em estágio avançado.

# Como começar:

## Após a Venda, Prepare-se para Começar:

Selecione um ponto de partida (um projeto piloto). Mas tome muito cuidado com projeto piloto. Pois, ele deve ser bem-sucedido e deve ter alguma importância para o negócio, **mas não devem ser deve ser crítico ao negócio (de missão crítica).**

Será necessário ter **SCRUM Master e PO (Product Owner) experientes;**

Forneça **treinamento a todas as pessoas da equipe**, simule bastante antes de começar, tenha certeza que todos entenderam as **práticas SCRUM, os princípios e os valores.**

### Prepare as ferramentas:

- Task Board (Quadro de Tarefas) é extremamente importante para dar visibilidade e transparência ao projeto desenvolvimento de software, lembre que Task Board é também uma ferramenta de comunicação para a Gestão à Vista;

Você precisará de pinceis, post-its, cartas para o Planning Poker, Cartões para as histórias e etc...

**Estimativa:** Inicialmente você poderá fazer as estimativas baseada em dias (dias ideais), pois, é mais fácil para equipe. Quando a equipe tiver mais experiência passe a usar pontos (Story points).

Nos primeiros projetos considere uma ajuda externa, a contratação de **Agile Coach** poderá ser bastante útil.

### Considere Pessoas, Processos e Ferramentas.

Lembre-se que a **Colaboração** é um fator-chave para o sucesso;

**Esteja preparado para viver o desafio.**



# Mini-Vocabulário

**Sprint** = iteração

**Product Backlog** = Lista de requisitos funcionais de um produto (com o nível de prioridade definido)

**Product Owner** = Analista de Negócio ou Especialista de Negócio

**SCRUM Master** = Líder servidor, se papel é muito próximo de técnico de futebol ele trabalha para que a equipe produza resultado, mas não entra em campo para jogar.

**Task Board** = Quadro de tarefas

**Impedimento** = É tudo aquilo que pode impedir a equipe de realizar seu trabalho, seja falta de informação ou falta de recursos de infraestrutura.

**Execução das práticas do SCRUM** = muito parecido com o velho e infalível PDCA.

**Timebox** = tempo (horas/dias) bem definido e imutável, sonho de todo gestor de projeto.

**Burndown** = É um gráfico que ele representa o **trabalho restante sobre tempo**



**Equipe SCRUM** = Equipe engajada, auto-gestão e multifuncional (pig dream team).



# Referências

Agile Project Management with Scrum

Autor: Ken Schwaber

Agile Software Development with Scrum

Autor: Ken Schwaber e Mike Beedle

The Enterprise and Scrum

Autor: Ken Schwaber

Agile Retrospectives: Making Good Teams Great -

Autores: Esther Derby, Diana Larsen e Ken Schwaber

**Jeff Sutherland:**

<http://jeffsutherland.com>

**Ken Schwaber:**

<http://www.controlchaos.com>

[www.mountangoatsoftware.com/scrum](http://www.mountangoatsoftware.com/scrum)

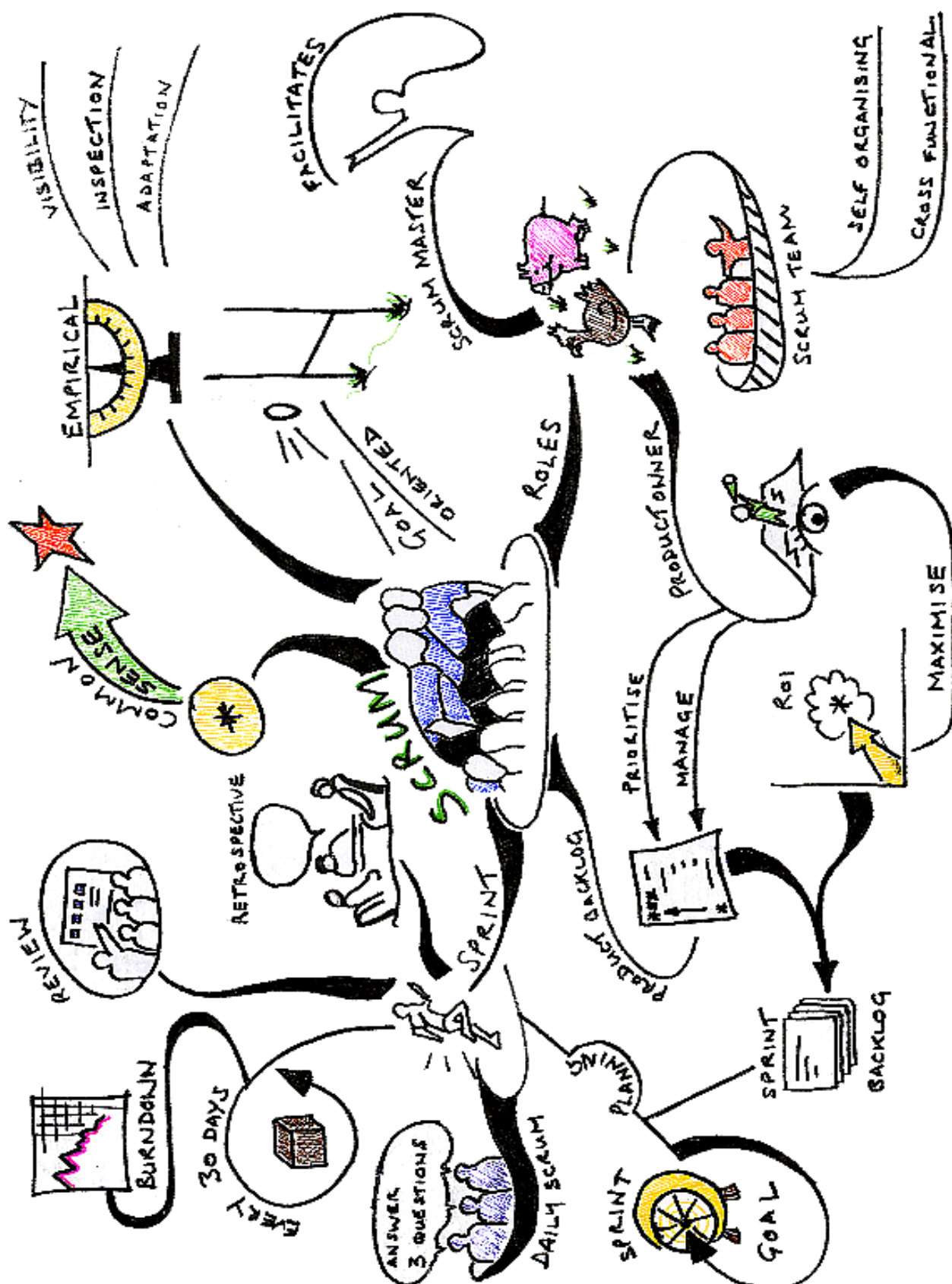
[www.scrumalliance.org](http://www.scrumalliance.org)

[www.mountangoatsoftware.com/scrum-a-presentation](http://www.mountangoatsoftware.com/scrum-a-presentation)



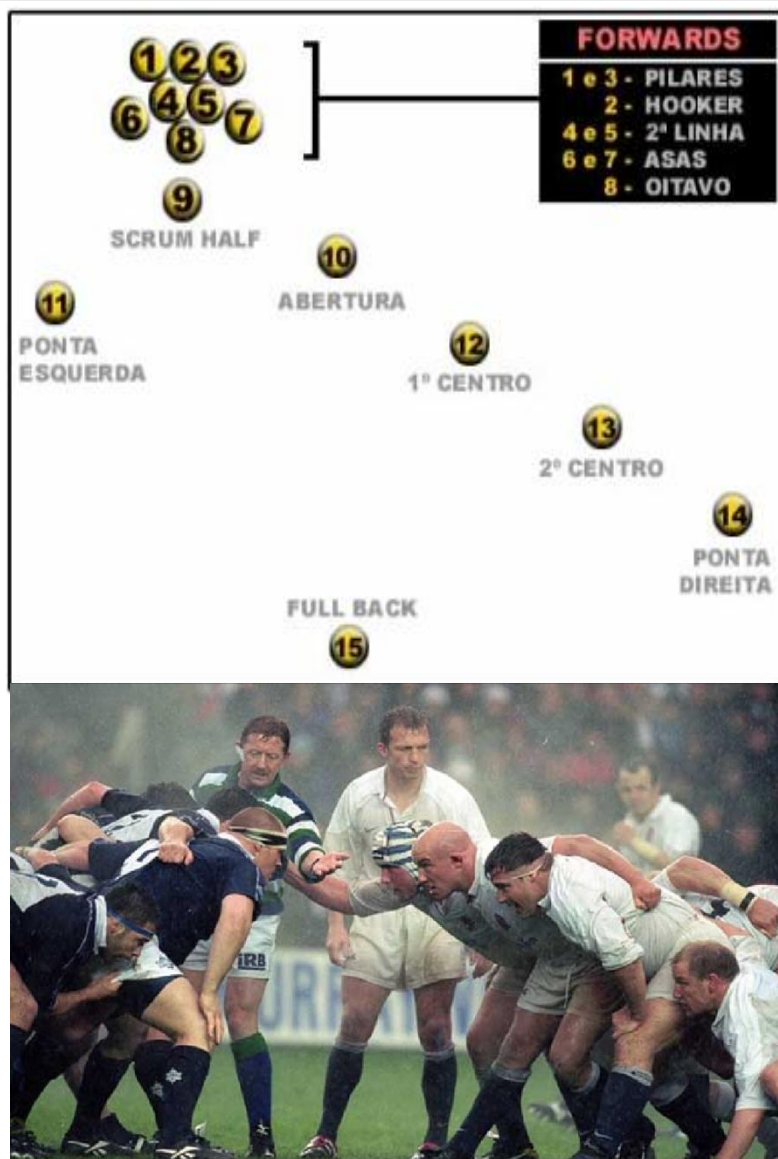


# SCRUM (em única página)





# SCRUM. Rugby



## SCRUM no jogo de Rugby:

A formação fixa chamada de **Scrum**, os fowards, que é um conjunto de oito jogadores abraçados, realizam uma força onde o objetivo é empurrar o outro time.

Essa jogada só é eficaz quando os oito “fowards” fazem força ao mesmo tempo e na mesma direção, fazendo assim que o vetor resultante, de suas forças individuais, seja maior que o do time adversário e possa assim pegar bola.

# Notas:

## Marcas Registradas:

Todos os termos mencionados e reconhecidos como Marca Registrada e/ou comercial são de responsabilidade de seus proprietários. O autor informa não estar associada a nenhum produto e/ou fornecedor apresentado neste material. No decorrer deste, imagens, nomes de produtos e fabricantes podem ter sido utilizados, e desde já o autor informa que o uso é apenas ilustrativo e/ou educativo, não visando ao lucro, favorecimento ou desmerecimento do produto/fabricante.

---

## Melhoria e Revisão:

Este material esta em processo constante de revisão e melhoria, se você encontrou algum problema ou erro envie um e-mail nós.

---

## Criticas e Sugestões:

Nós estamos abertos para receber criticas e sugestões que possam melhorar o material, por favor envie um e-mail para nós.

---

## Imagens:

Google, Flickr e Banco de Imagem.

tecnologia.com.br

Rildo F dos Santos (rildosan@uol.com.br)

# Licença:



Atribuição 3.0 Genérica

## Você pode:



copiar, distribuir, exhibir 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.

- ♦ Para cada novo uso ou distribuição, você deve deixar claro para outros os termos da licença desta obra.
- ♦ Qualquer uma destas condições podem ser renunciadas, desde que Você obtenha permissão do autor.
- ♦ Nothing in this license impairs or restricts the author's moral rights.

Termo de exoneração de responsabilidade

**Qualquer direito de uso legítimo (ou "fair use") concedido por lei, ou qualquer outro direito protegido pela legislação local, não são em hipótese alguma afetados pelo disposto acima.**

Este é um sumário para leigos da [Licença Jurídica \(na íntegra\)](#).



# SCRUM Experience

@tecnologia.com.br

Rildo F Santos

[rildosan@etecnologia.com.br](mailto:rildosan@etecnologia.com.br)

[rildo.santos@companyweb.com.br](mailto:rildo.santos@companyweb.com.br)

Twitter: <http://twitter.com/rildosan>  
Blog: <http://rildosan.blogspot.com/>

