

# METODOLOGIAS DE DESENVOLVIMENTO DE SOFTWARE (MDS)

## Cap. 4

### Scrum

Ano Lectivo 2019/2020

POLITÉCNICO DE LEIRIA . TeSP PSI

1

POLITÉCNICO DE LEIRIA . TeSP PSI . MDS

## Scrum

### Origens

- Criada no início dos anos 90 por Ken Schwaber e Jeff Sutherland
- A metodologia Scrum é uma **metodologia ágil genérica que se foca na gestão do processo de desenvolvimento iterativo e incremental** em vez de em práticas ágeis específicas
- Scrum é um termo de rugby para a formação *close-knit shoulder-to-shoulder* que uma equipa de forma para mover conjuntamente a bola para frente
- A palavra foi usada pela primeira vez por Takeuchi e Nonaka num famoso artigo publicado na Harvard Business Review, no qual descreviam os projetos de desenvolvimento de produtos mais bem-sucedidos do Japão.

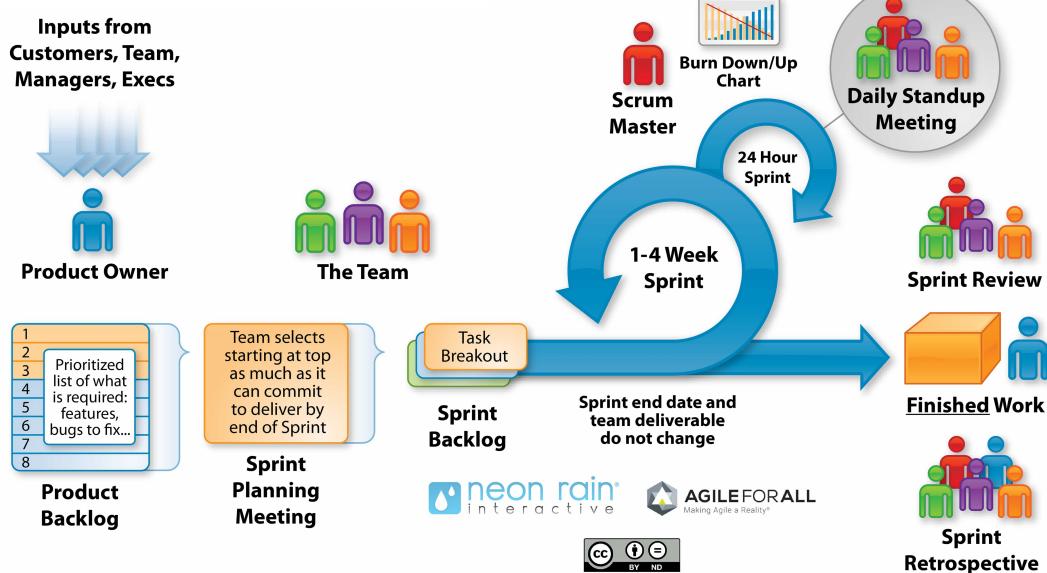


© 2020 Diana Santos

2

## Scrum

### Scrum Framework



3

## Scrum

### Objetivos gerais

- Metodologia ágil
- Utiliza uma abordagem iterativa e incremental
- Trabalho é organizado em três “camadas” de complexidade:
  - **Produto > Release > Sprint**
- Procura aumentar a previsibilidade
- Procura diminuir o risco

4

## Scrum

---

### Componentes

#### 1. Artifacts

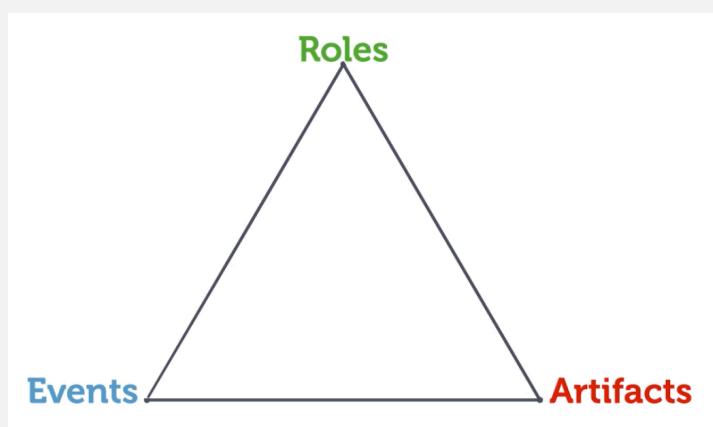
- *Product Backlog*
- *Sprint Backlog*
- *Product Increment*

#### 2. Roles

- *Product Owner*
- *Scrum Master*
- *Development Team*

#### 3. Events

- *Sprint Planning*
- *Daily Scrum*
- *Sprint Review*
- *Sprint Retrospective*



© 2020 Diana Santos

5

## Scrum

---

### Componentes: Artifacts



1. **Product backlog** – coleção de todos os itens ordenados que se quer incluir no Produto
  - Cada Item contém descrição, ordem ou prioridade, estimativa de custo e valor
  - Cada um destes Itens pode ser: uma User Story, uma “feature”, etc.



2. **Sprint backlog** – coleção de itens do *Product Backlog* que a Equipa de Desenvolvimento decide incluir num *Sprint* durante a *Sprint Planning*



3. **Incremento do Produto** – é o trabalho feito (*Done*) da porção do produto completado durante o *Sprint* mais o valor dos incrementos anteriores
  - Maximizam a transparência (priorização, planeamento, progresso) e partilham o entendimento por todos do trabalho

© 2020 Diana Santos

6

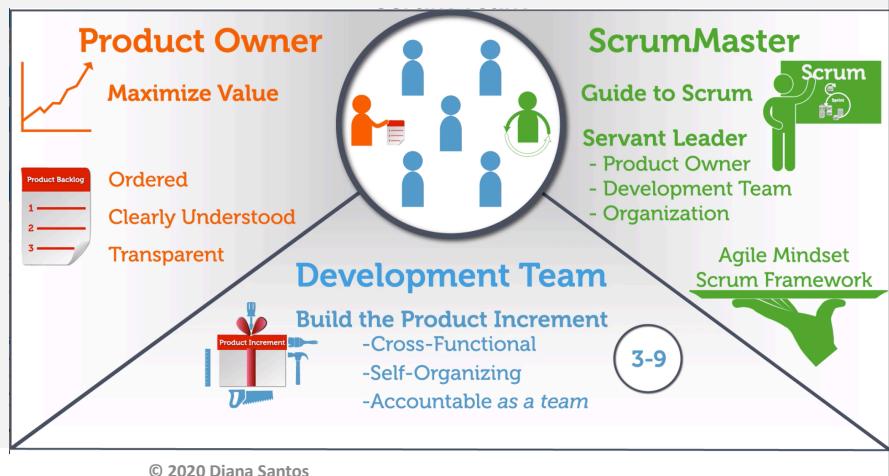
POLITÉCNICO DE LEIRIA . TeSP PSI . MDS

## Scrum

Componentes: Roles

Dentro da Scrum Team

1. **Product Owner**
2. **Scrum Master**
3. **Development Team**



7

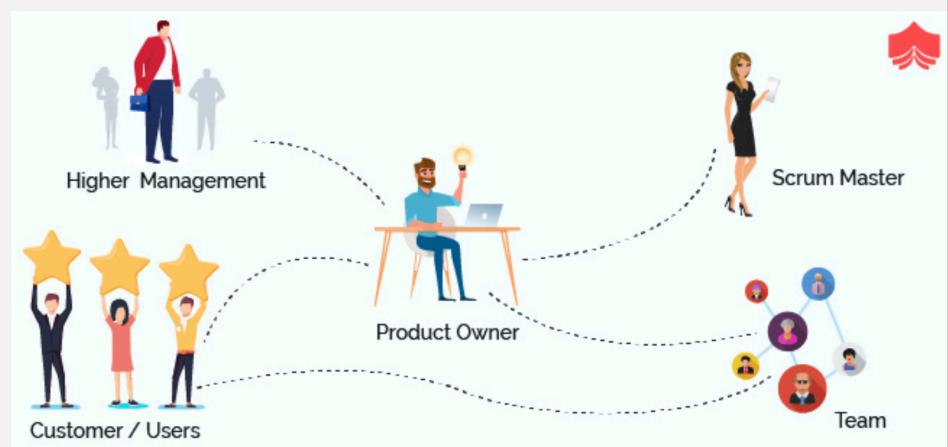
POLITÉCNICO DE LEIRIA . TeSP PSI . MDS

## Scrum

Componentes: Roles

Fora da Scrum Team

4. **Stakeholders**



8

POLITÉCNICO DE LEIRIA . TeSP PSI . MDS

## Scrum

---

Componentes: *Roles*

### 1. Product Owner

- Objetivo principal: garantir o tipo e a qualidade do produto final e do trabalho da equipa de desenvolvimento
- Representa os *Stakeholders* dentro do Scrum
- Gere o *Product Backlog*
  - Define claramente os itens do *Product Backlog*
  - Prioriza os itens do *Backlog* de acordo com a sua importância e valor
  - Maximiza o valor do trabalho realizado pela equipa de desenvolvimento
  - Garante a clareza e transparência do *Backlog* para todos
  - Torna visível a cada momento o trabalho a realizar pela equipa de desenvolvimento
- Comunica à equipa os itens do *Product Backlog*

© 2020 Diana Santos

9

POLITÉCNICO DE LEIRIA . TeSP PSI . MDS

## Scrum

---

Componentes: *Roles*

### 2. Scrum Master

- Objetivo principal: o cumprimento do Scrum
- Promove a adoção do Scrum dentro da Equipa Scrum e da Organização
- Facilita os eventos Scrum (ex.: Daily Scrum)
- Colabora com o *Product Owner* na comunicação à Equipa de Desenvolvimento do *Product Backlog* e no seu cumprimento
- Dar as condições necessárias à equipa de desenvolvimento para cumprir as suas tarefas da forma mais eficaz (incluindo eliminação de qualquer constrangimento, proteção de interferências externas e motivação para cumprir os prazos)
- Organizar e gerir as diversas reuniões com a equipa de desenvolvimento

© 2020 Diana Santos

10

POLITÉCNICO DE LEIRIA . TeSP PSI . MDS

## Scrum

---

Componentes: *Roles*

### 3. Development Team

- **Objetivo principal:** acrescenta funcionalidade ao produto (incremento do Produto)
- É auto-organizada – capaz de organizar o seu próprio trabalho
- Multi-funcional – no seu conjunto, tem todas as competências necessárias para criar incremento do Produto
- No Scrum não há funções diferenciadas dentro da Equipa de Desenvolvimento
- "Membros individuais da Equipa de Desenvolvimento podem ter competências e áreas de especialização, mas a **responsabilidade pertence à Equipa de Desenvolvimento como um todo**"
- O Scrum está pensado para equipas de 3 a 9 membros

© 2020 Diana Santos

11

POLITÉCNICO DE LEIRIA . TeSP PSI . MDS

## Scrum

---

Componentes: *Roles*

### 4. Stakeholders

- Qualquer pessoa interessada/com influência no produto que não faz parte da *Scrum Team*
- Inclui:
  - Clientes
  - Gestores séniores dentro da organização
  - Patrocinadores do projeto
  - Pessoas/entidades que se relacionam com o projeto para entregar o produto
  - Partes externas (ex: reguladores)

© 2020 Diana Santos

12

## Scrum

---

Componentes: *Events*

- Procuram criar rotinas dentro da equipa e minimizar “reuniões extraordinárias”
- Todos devem ter uma duração máxima estipulada
- Tipos:
  1. *Sprint Planning*
  2. *Daily Scrum*
  3. *Sprint Review*
  4. *Sprint Retrospective*

© 2020 Diana Santos

13

## Scrum

---

Componentes: *Events*

### ***Sprint***

- Duração: 1 a 4 semanas (tipicamente: 2 semanas) – deve ser regular ao longo do projeto para permitir maximizar a qualidade das estimativas
- Começa com uma *Sprint Planning* e inclui *Daily Scrums*, *Sprint Review* e *Sprint Retrospective* (incluindo o trabalho de desenvolvimento em si)
- O objetivo do *Sprint* não pode ser alterado dentro do *Sprint*
- A versão do produto no final do *Sprint* deve representar uma melhoria em relação à versão apresentada no *Sprint* anterior
- O *Product Owner* pode decidir cancelar o *Sprint* (ex.: se a meta do *Sprint* se tiver tornado obsoleta)



© 2020 Diana Santos

14

POLITÉCNICO DE LEIRIA . TeSP PSI . MDS

## Scrum

Componentes: *Events*

### **Sprint Planning**

- Máximo de 8 horas para um *Sprint* de um mês
- O *Scrum Master* garante que decorra dentro do limite de tempo estipulado e que todos os participantes entendem o seu propósito



Inputs	Outputs
<i>Product Backlog</i>	O que vai ser entregue no final do <i>Sprint</i> ( <i>Sprint Backlog</i> , Meta do <i>Sprint</i> )
Última versão do Produto	Como vai ser realizado – Plano de Entrega
Capacidade da Equipa de Desenvolvimento (estimativa baseada no histórico)	

© 2020 Diana Santos

15

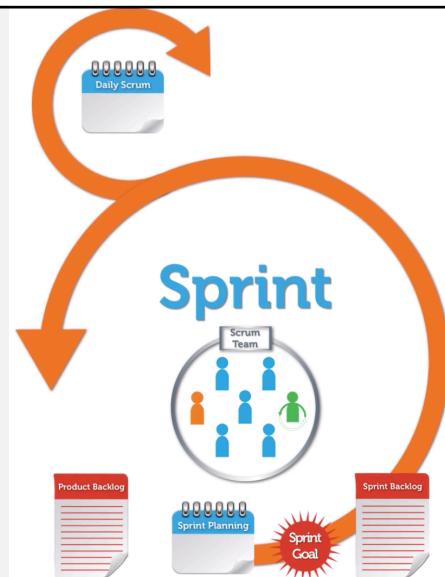
POLITÉCNICO DE LEIRIA . TeSP PSI . MDS

## Scrum

Componentes: *Events*

### **Daily Scrum**

- |  |  |
|--|--|
| <ul style="list-style-type: none"> <li>• <u>Duração:</u> 15m</li> <li>• Deve acontecer num horário fixo, no mesmo local e com os intervenientes em pé</li> <li>• O <i>Scrum Master</i> deve garantir que se realiza</li> <li>• Envolve que cada elemento da equipa explique aos restantes:           <ul style="list-style-type: none"> <li>• o que fez desde a última reunião</li> <li>• o que irá fazer no dia atual</li> <li>• que dificuldades prevê em relação ao trabalho a realizar</li> </ul> </li> <li>• <u>Vantagem:</u> todos os elementos da equipa estão a par de tudo o que acontece no projeto</li> </ul> | <u>Frequência:</u> uma por dia de trabalho |
|--|--|



© 2020 Diana Santos

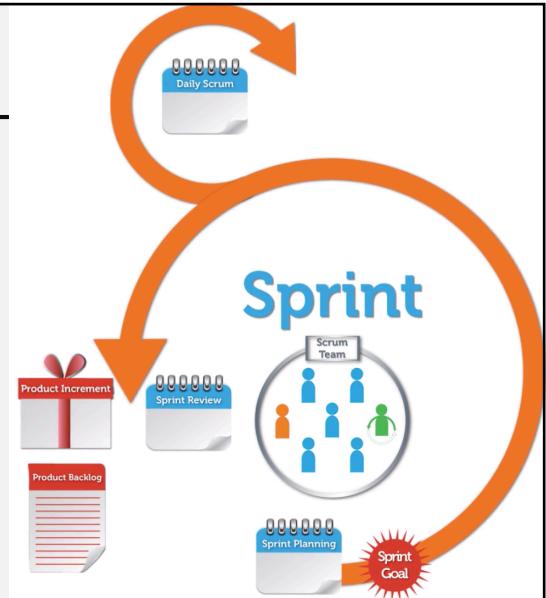
16

## Scrum

Componentes: *Events*

### **Sprint Review**

- Duração: máx. 4 horas num *Sprint* de 1 mês
- Realizada no final de cada *Sprint*
- O *Scrum Master* deve garantir que se realiza
- Serve para inspecionar o incremento
- Participam a Equipa de Desenvolvimento e, não obrigatoriamente, os *Stakeholders*



© 2020 Diana Santos

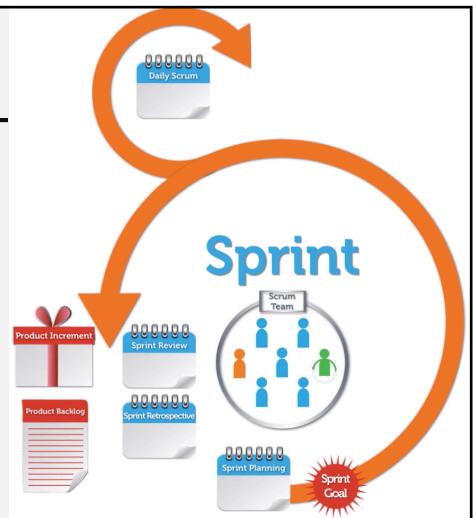
17

## Scrum

Componentes: *Events*

### **Sprint Retrospective**

- Duração: máx. 3 horas num *Sprint* de 1 mês
- Realizada pouco depois da *Sprint Review*
- O *Scrum Master* deve garantir que se realiza
- Serve para:
  - inspecionar como decorreu a execução do *Sprint* (i.e., pontos positivos e negativos)
  - identificar possíveis melhorias no processo de desenvolvimento para evitar novos erros
- Participa apenas a Equipa de Desenvolvimento



© 2020 Diana Santos

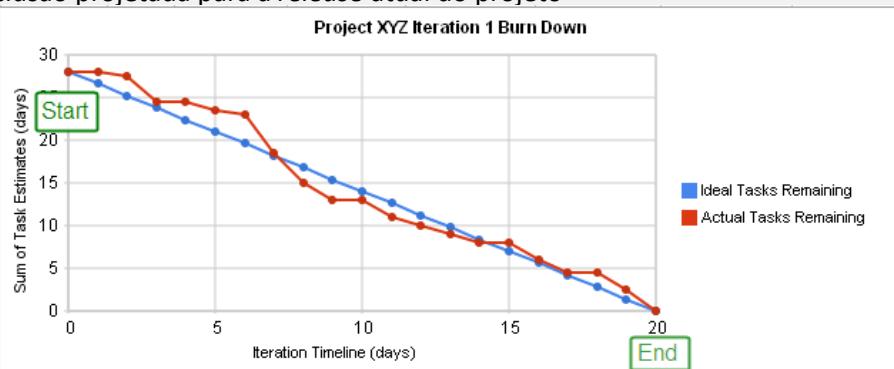
18

## Scrum

---

### Sprint Tracking Mechanisms

- Ferramentas que permitem que o projeto esteja no bom caminho para fornecer a solução esperada dentro do planeamento desejado
- **Burn-Down chart** – gráfico que mostra o trabalho concluído (por dia no exemplo) em relação à taxa de conclusão projetada para a *release* atual do projeto



19

## Scrum

---

### Burn-Down Chart

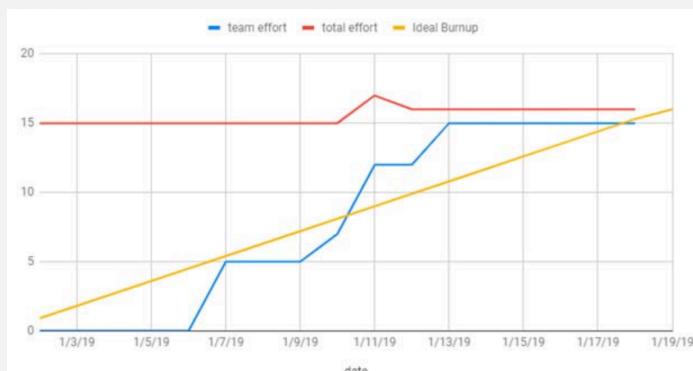
- **Sprint burn-down chart** – ferramenta de representação visual de medição que mostra o trabalho que resta no *Sprint* (usualmente em termos de horas de tarefa)
- Geralmente é atualizado na *Daily Scrum*
- À medida que o *Sprint* progride, a quantidade de trabalho restante deve diminuir constantemente e tender a ser concluída no último dia do *Sprint*
- *Burn-downs* que mostram o trabalho a crescer ou com poucas tarefas concluídas são sinais para o *Scrum Master* e para a equipa de que o *Sprint* não está bem encaminhado

20

## Scrum

### Burn-Up Chart

- Representação visual do processo do trabalho da equipa
- Mostra o *scope* do projeto e do trabalho concluído



© 2020 Diana Santos

21

## Scrum

### Burn-Up Chart

- **Eixo horizontal:** Tempo (dias)
- **Eixo vertical:** Quantidade de trabalho (*story points*)
- **Total effort** — esforço total requerido para atingir o objetivo da equipa (*sprint/release*)
- **Team effort** — progresso da equipa em relação ao objetivo dentro de um período (*story points* concluídos)
- **Ideal line** — ritmo de trabalho ideal necessário para cumprir o prazo do projeto



© 2020 Diana Santos

22

## Scrum

---

### Burn-Up Chart

- O *Burn-Up chart* é atualizado no final de cada unidade de tempo e mostra o progresso real da equipa
- A linha de *team effort* sobe de baixo para cima em direção à linha de *total effort*. Quando as duas linhas se encontram, o objetivo da equipa é atingido
- A equipa pode ver o trabalho restante pela observação da distância entre a linha de *team effort* e a linha de *total effort*
- **Vantagem do burn-up chart:** simplifica a monitorização, a modificação e melhora o processo de trabalho da equipa

© 2020 Diana Santos

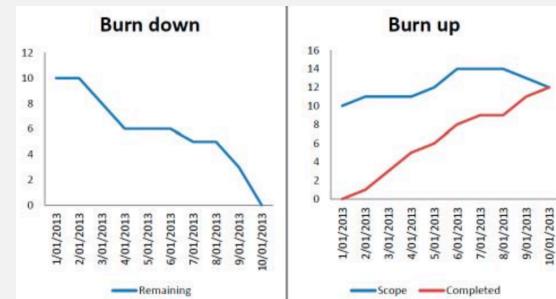
23

## Scrum

---

### Burn-Down vs Burn-Up Chart

- **Burn-down chart**
  - Mostra a quantidade de trabalho restante num projeto
  - A linha é decrescente
  - Mais simples de entender porque só tem 1 linha
- **Burn-up chart**
  - Mostra a quantidade de trabalho que foi concluída e o scope total do projeto
  - A linha é crescente
  - É mais completo porque mostra o trabalho atingido e o trabalho concluído nos incrementos anteriores
- Ambos usam os mesmos eixos



© 2020 Diana Santos

24

## Scrum

### Velocity Chart

- **Velocity** – taxa de progresso de uma *Scrum Team* que expressa a quantidade de, p. ex., *story points* concluídos por *Sprint*
- Regra para calcular a velocidade: apenas as *stories* concluídas no final da iteração são contadas. É estritamente proibido contar trabalho parcialmente concluído (p. ex., apenas código mas sem ter sido testado)
- Após alguns *Sprints*, a velocidade de uma equipa é previsível e permite uma estimativa bastante precisa sobre o tempo necessário até que todas as entradas no *Product Backlog* sejam concluídas
  - Exemplo:

**Velocidade da equipa = 30 story points**

**Quant. restante de trabalho = 155 story points**

Nº Sprints?

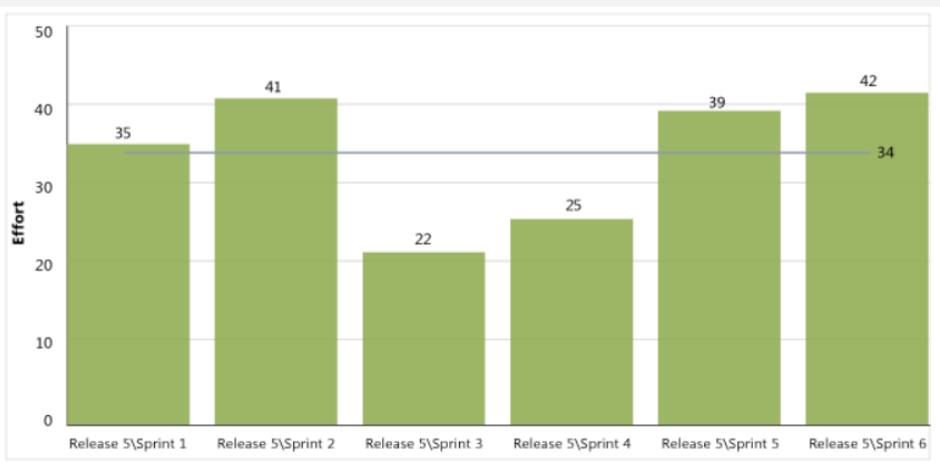


© 2020 Diana Santos

25

## Scrum

### Velocity Chart



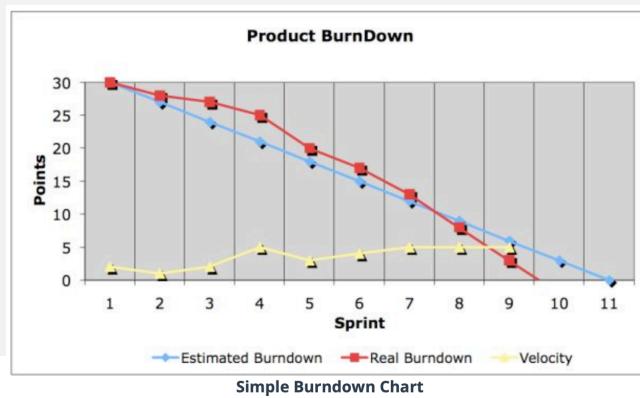
© 2020 Diana Santos

26

## Scrum

### Velocity Chart

- Na prática, as entradas no *Product Backlog* são alteradas ao longo da duração do projeto. Novas *stories* são adicionadas e outras são alteradas/eliminadas
- No gráfico *Burndown* simples, a velocidade da equipa e a mudança no *scope* não são diferenciadas

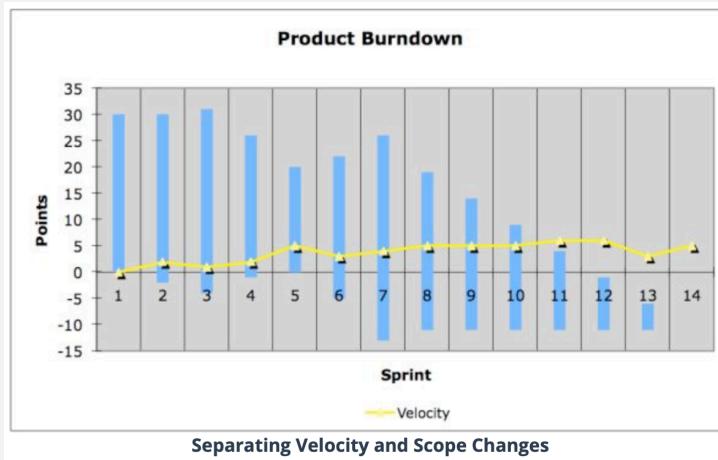


27

## Scrum

### Velocity Chart

- O tamanho de cada barra representa a quantidade total de trabalho restante no início de cada *sprint*
- A velocidade da equipa é subtraída do topo, enquanto as mudanças no *scope* alteram a parte inferior da barra



© 2020 Diana Santos

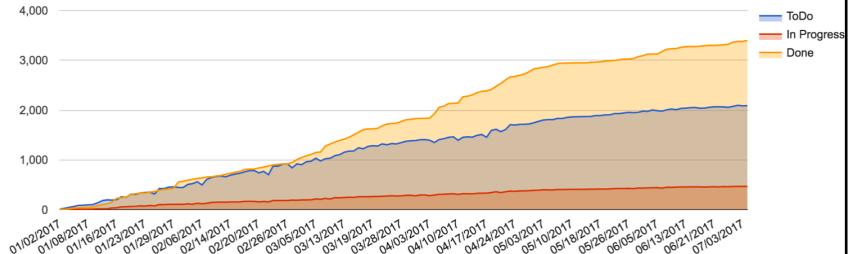
28

## Scrum

---

### *Cumulative Flow Diagram*

- Gráfico de área que mostra o progresso dos itens de trabalho de um projeto por um período de tempo específico
- Ferramenta fundamental para visualizar o progresso do projeto e ajuda a identificar possíveis problemas
- Mostra a contagem e ritmo de progresso de itens do *Backlog* no período selecionado
- É uma versão avançada do gráfico *Burn-Up*



29

## Bibliografia

---

- Ian Sommerville, "Software Engineering", 10th Edition, Addison-Wesley, 2015  
<http://dinus.ac.id/repository/docs/ajar/Sommerville-Software-Engineering-10ed.pdf>
- Apontamentos teóricos de Metodologias de Projecto Multimédia (TeSP Desenvolvimento Web e Multimédia), IPLeiria 2018/19

30