

# Desenvolvimento Ágil

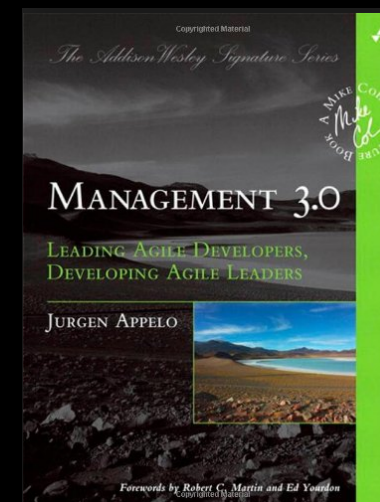
Diana Braga

# Retrospectiva



# Bibliografia

- Técnicas para Gerenc. De Projtos de Software- José Carlos C Martins (Caps. 3 a 8)
- Eng Software - Roger Pressman (Cap. 4)
- Eng Software - Sommerville (Cap. 17)
- Eng Moderna – Marco Tulio (Cap 2)
- Desenvolvimento de Software com Scrum
- Management 3.0



# Antes de começar

- Nenhum processo é uma bala-de-prata
- Processo ajuda a não cometer certos "grandes erros"
- Processos não são adotados 100% igual ao manual
  - Bom senso é importante
  - Experimentação é importante

# Manifesto Ágil

- 2001
- Envolvidos (Aliança Ágil)
  - 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



# O Manifesto

**Indivíduos e Iterações** mais do que *Processos e Ferramentas*

**Software Funcionando** mais do que *Documentação Extensa*

**Colaboração do Cliente** mais do que *Negociação de Contratos*

**Responder a Mudanças** mais do que *Seguir um Plano*

# Doze princípios da agilidade

1. Ter como maior prioridade satisfazer o cliente por meio da entrega de software
2. Modificações contínuas são bem-vindas e levam à competitividade para o cliente
3. Entrega de software funcionando em períodos de duas semanas a dois meses
4. O pessoal de negócio e os desenvolvedores devem trabalhar juntos diariamente
5. Construção de projetos em torno de indivíduos motivados e talentosos
6. Usar métodos informais de comunicação como conversar pessoalmente

# Doze princípios da agilidade

- 7. Software funcionando é a principal medida de progresso
- 8. Promover o desenvolvimento sustentável, mantendo um ritmo de produção
- 9. Atenção contínua à excelência técnica e ao bom projeto
- 10. Simplicidade é essencial
- 11. As melhores arquiteturas, requisitos e projetos surgem de equipes auto-organizadas
- 12. A equipe deve, em intervalos regulares, refletir sobre como se tornar mais efetiva



# O que é o desenvolvimento ágil?

- É uma filosofia e um conjunto de diretrizes que encorajam
  - A entrega do software logo de início
  - Equipes de projeto pequenas, motivadas e auto-organizáveis
  - Métodos informais de comunicação ao invés de documentos escritos
  - Adotar o cliente como parte da equipe



**E isso é realmente  
inovador?**

### **Métodos Tradicionais**

(Engenharia Tradicional)

Couger (1973)

Fagan (1976)

Quinnan (1980)

### **Times auto-organizáveis**

(Psicologia, Sociologia,  
Comportamento Organizacional)

Bachuk & Goode (1951)

Herbst (1962)

Thoms, Moore & Scott (1996)

### **Processo Flexível**

(Manufatura, Inovação e Internet)

Thomke & Reinertsen (1998)

MacCormack, Verganti & Iansiti  
(2001)

Verganti & Buganza (2005)

### **Prototipação**

(Engenharia de Software)

McCracken & Jackson (1982)

Naumann & Jenkins. (1982)

### **Releases e Iterações**

(Engenharia de Software)

Basili & Turner (1975)

Boehm (1986)

Bauer, Collar & Tang (1992)

### **Envolvimento antecipado do cliente**

(Marketing, Psicologia e Sociologia)

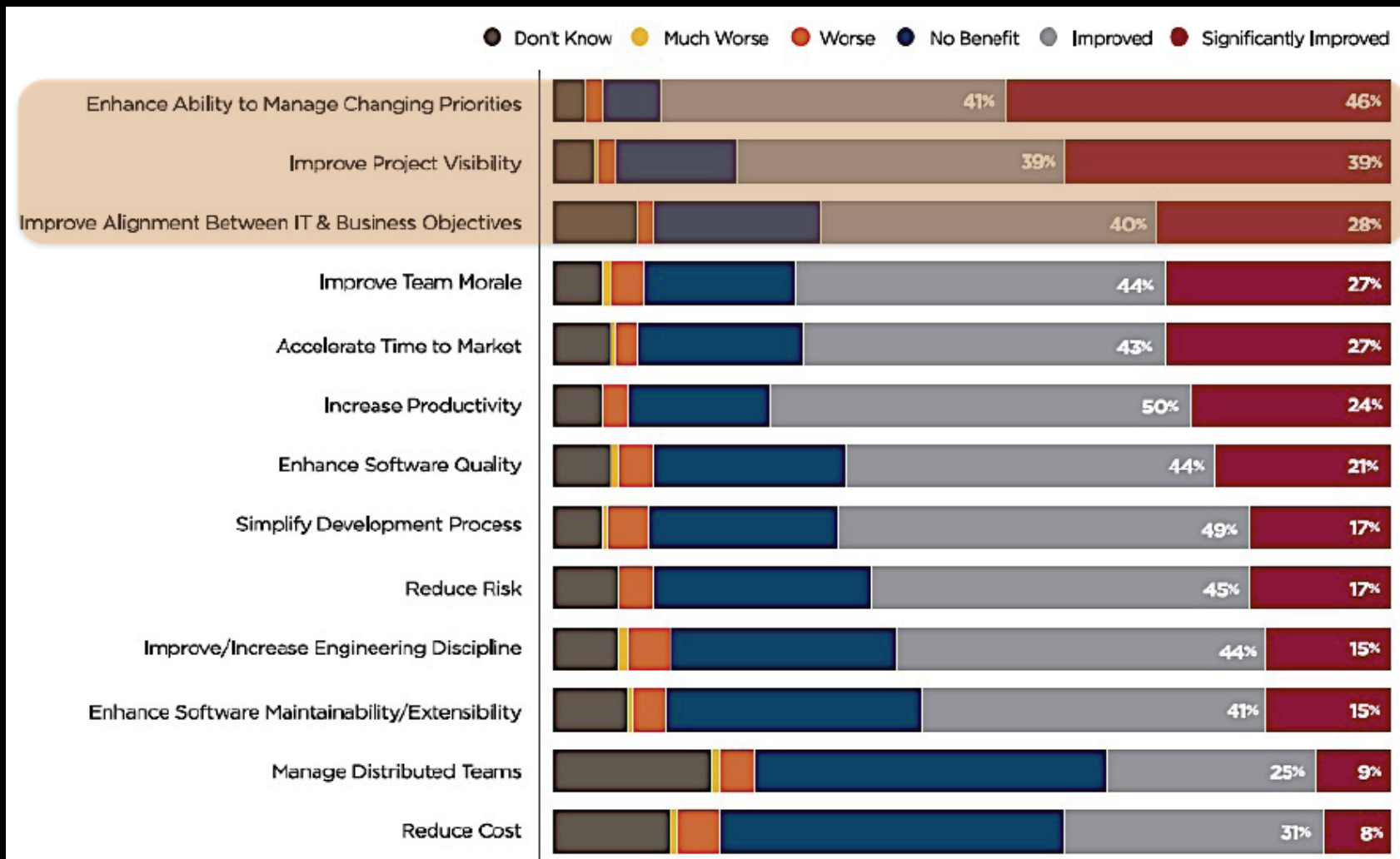
Von Hippel (1978)

King & Rodriguez (1981)

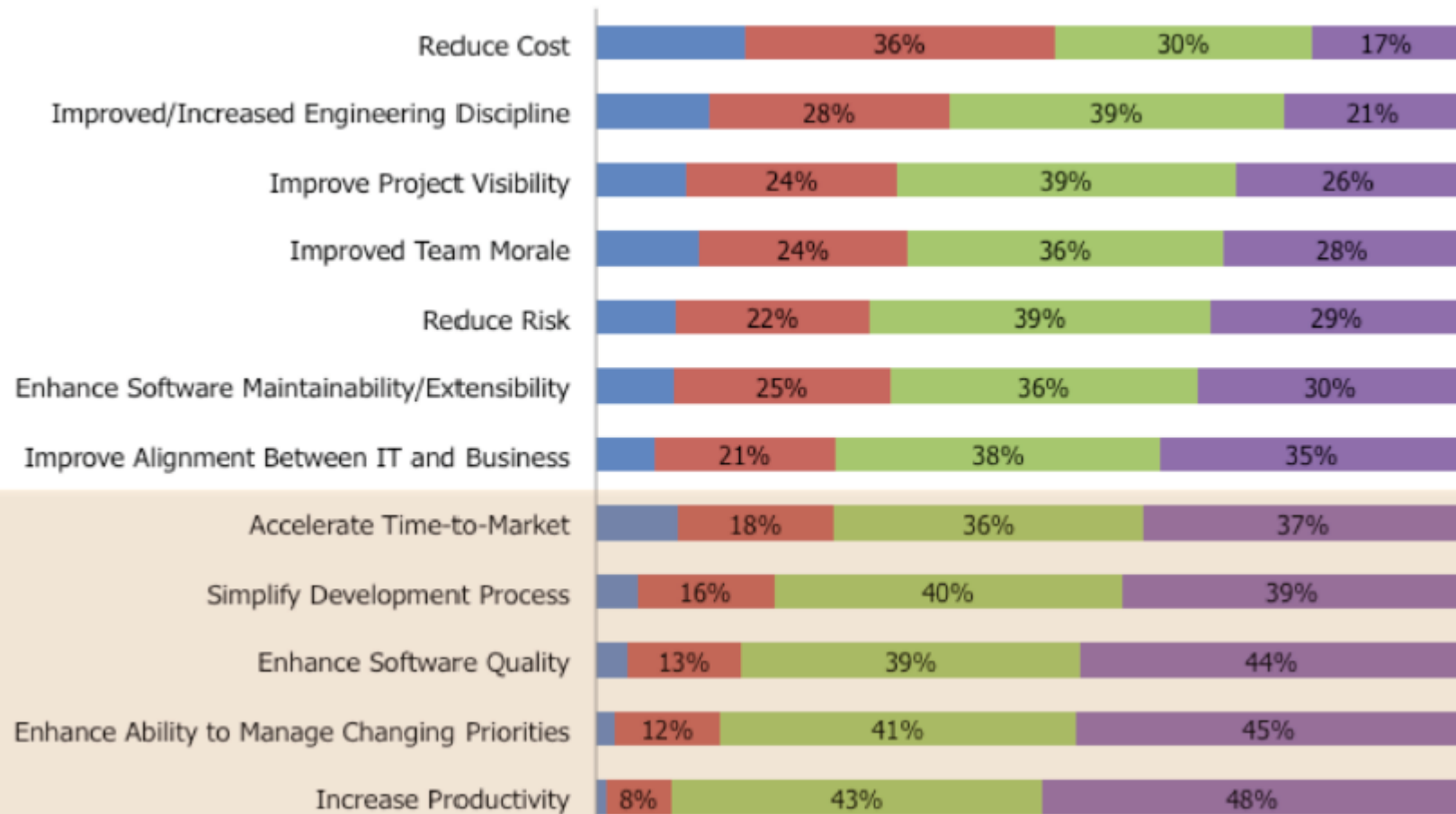
Robey & Farrow (1982)

Adaptado de: Rico, D. 2005. Agile Methods and the Link to Firm Performance e  
Abbas et al. 2008. Historical Roots of Agile Methods: Where did "Agile Thinking" Come from?

# Principais Benefícios Observados



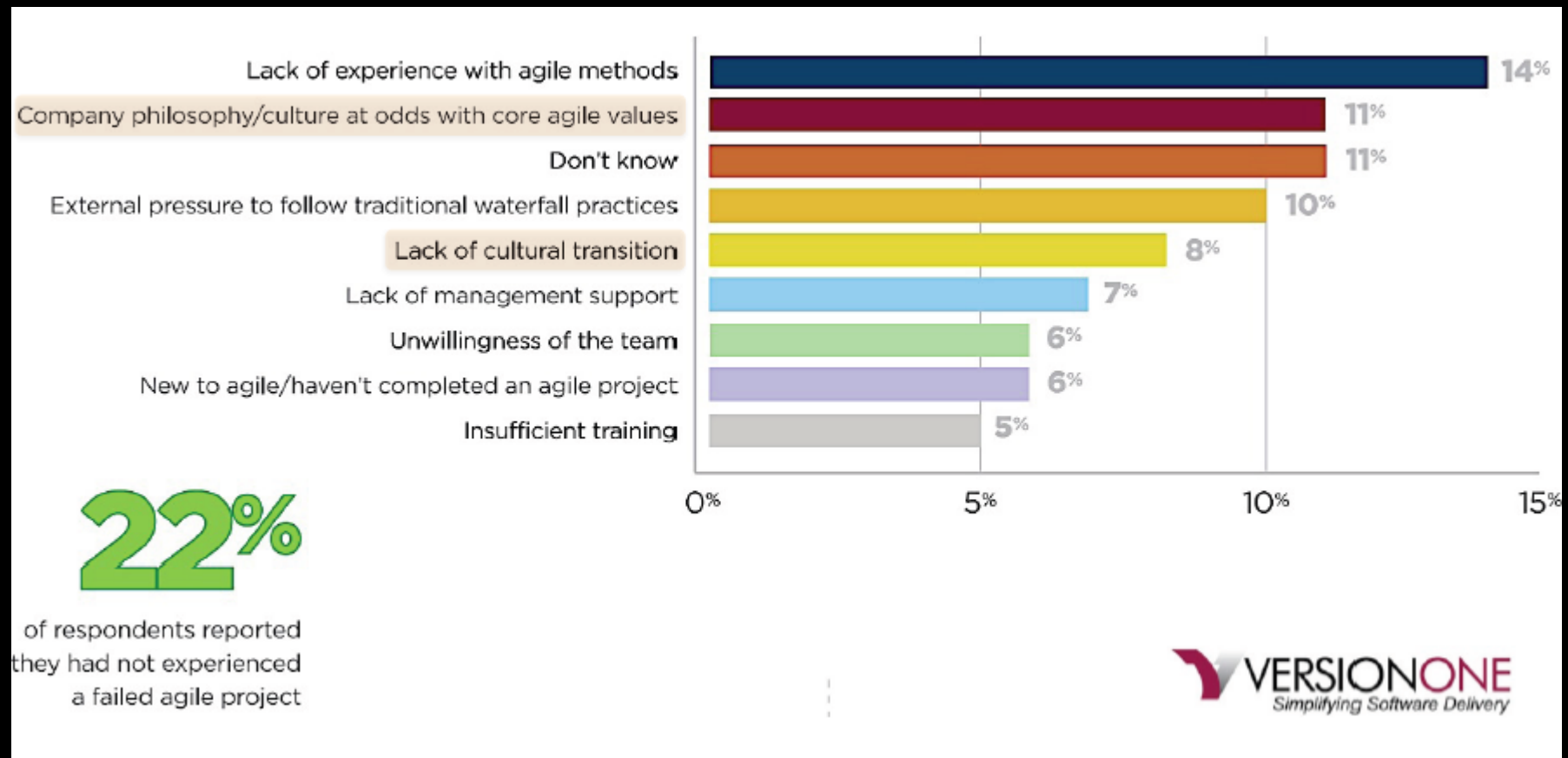
# Razão para Adoção



An aerial photograph of a bridge under construction or repair over a body of water. The bridge deck is partially completed, with several workers in orange safety vests and hard hats standing on the existing structure. A white SUV is parked on the right side of the bridge. Orange traffic cones are placed along the edge of the bridge deck. The water is a deep blue, and the bridge piers are visible in the water.

**MAS OS MÉTODOS ÁGEIS FALHAM?**

# Por que Métodos Ágeis Falham?





# Adaptação

- Requer grandes mudanças culturais no seu time e organização





# Fatores Humanos

- Características-chave de uma equipe ágil
  - Competência
  - Foco comum
  - Colaboração
  - Capacidade de tomada de decisão
  - Habilidade de resolver problemas vagos
  - Respeito e confiança mútua
  - Auto-organização
    - Organização do trabalho a ser feito
    - Organização do processo para melhor acomodar o ambiente local
    - Organização do cronograma para conseguir a melhor entrega do incremento

# SCRUM

- [http://www.slideshare.net/serge\\_rehem/scrum-em-15-minutos](http://www.slideshare.net/serge_rehem/scrum-em-15-minutos)

# Scrum

- **Scrum** é um processo para construir software incrementalmente em ambientes complexos, onde os requisitos não são claros ou mudam com muita frequência (considera antecipadamente a existência do caos)
- A metodologia é baseada em princípios semelhantes aos de XP: equipes pequenas, requisitos pouco estáveis ou desconhecidos, e iterações curtas para promover visibilidade para o desenvolvimento

# Papéis



# Papéis

- Product Owner
  - Define as funcionalidades do Produto
  - Decide a data de entrega e o conteúdo
  - Representa os stakeholders
  - Prioriza as funcionalidades conforme o valor de negócio
  - Ajusta as funcionalidades e suas prioridades a cada Sprint
  - Aceita ou rejeita os resultados

# Papéis

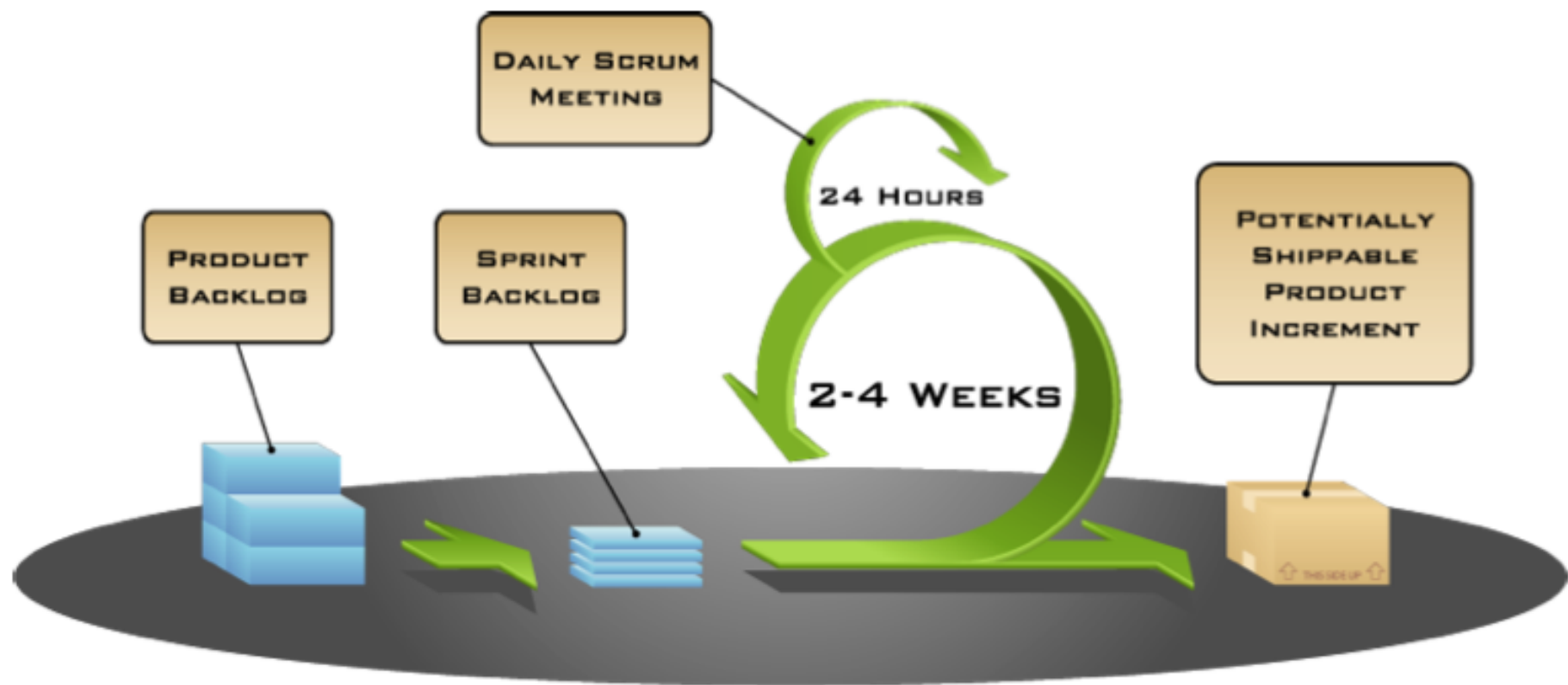
- Scrum Master
  - Responsável pela aplicação dos valores e práticas do Scrum
  - Remove impedimentos
  - Assegura que a equipe está totalmente funcional e produtivo
  - Permite a cooperação entre os diversos papéis e funções
  - Protege o time das interferências externas

# Papéis

- A Equipe
  - Entre 3-9 Pessoas
  - Multifuncional
  - Dedicados ao projeto
  - Auto-organizado



# Scrum



COPYRIGHT © 2005, MOUNTAIN GOAT SOFTWARE

# Scrum

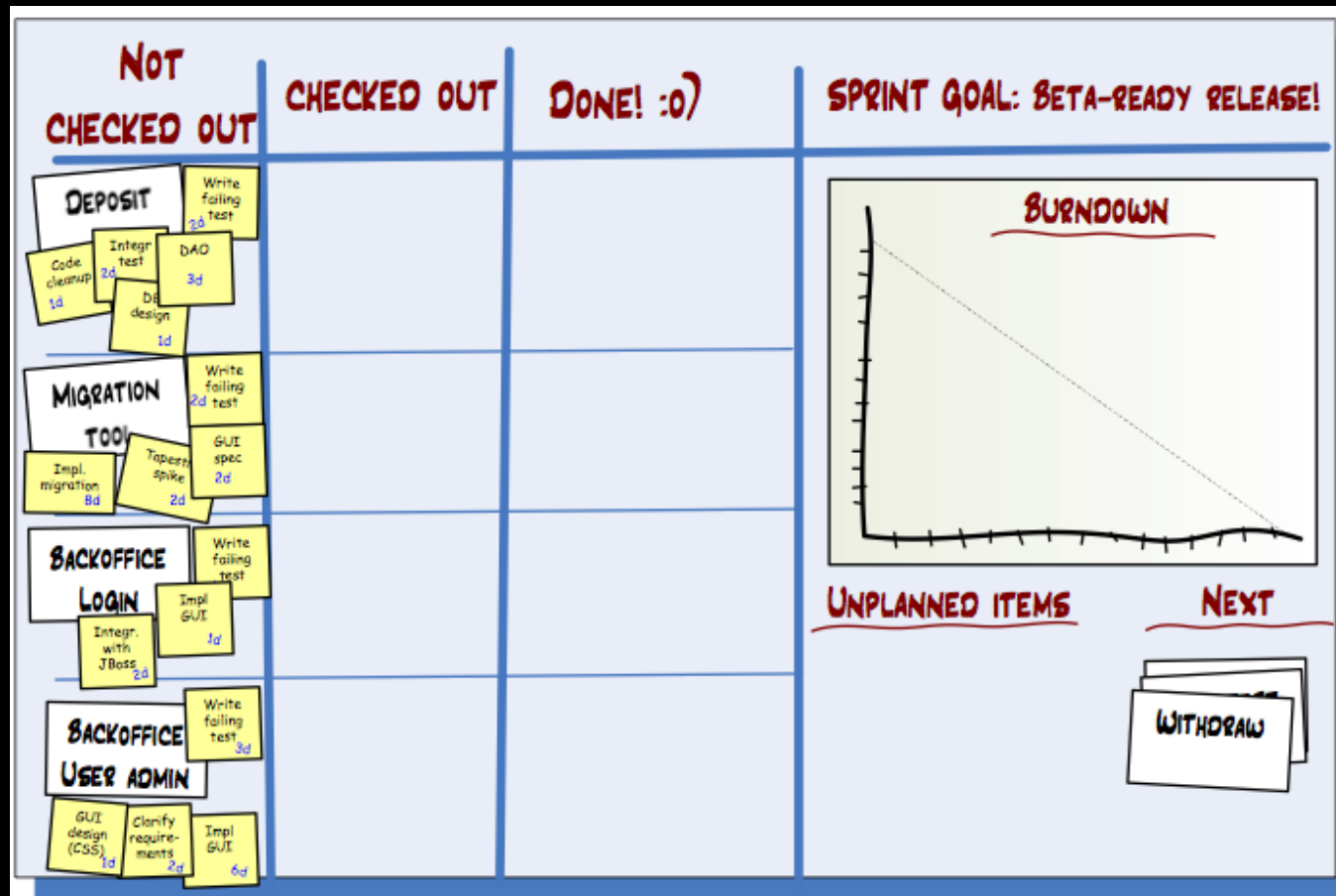
- Artefatos
  - Product Backlog
  - Sprint Backlog
  - Burndown Chart
- Fundamentos
  - Transparência
  - Comunicação
  - Simplicidade
- Podemos ter mais artefatos em um projeto?

# Product Backlog

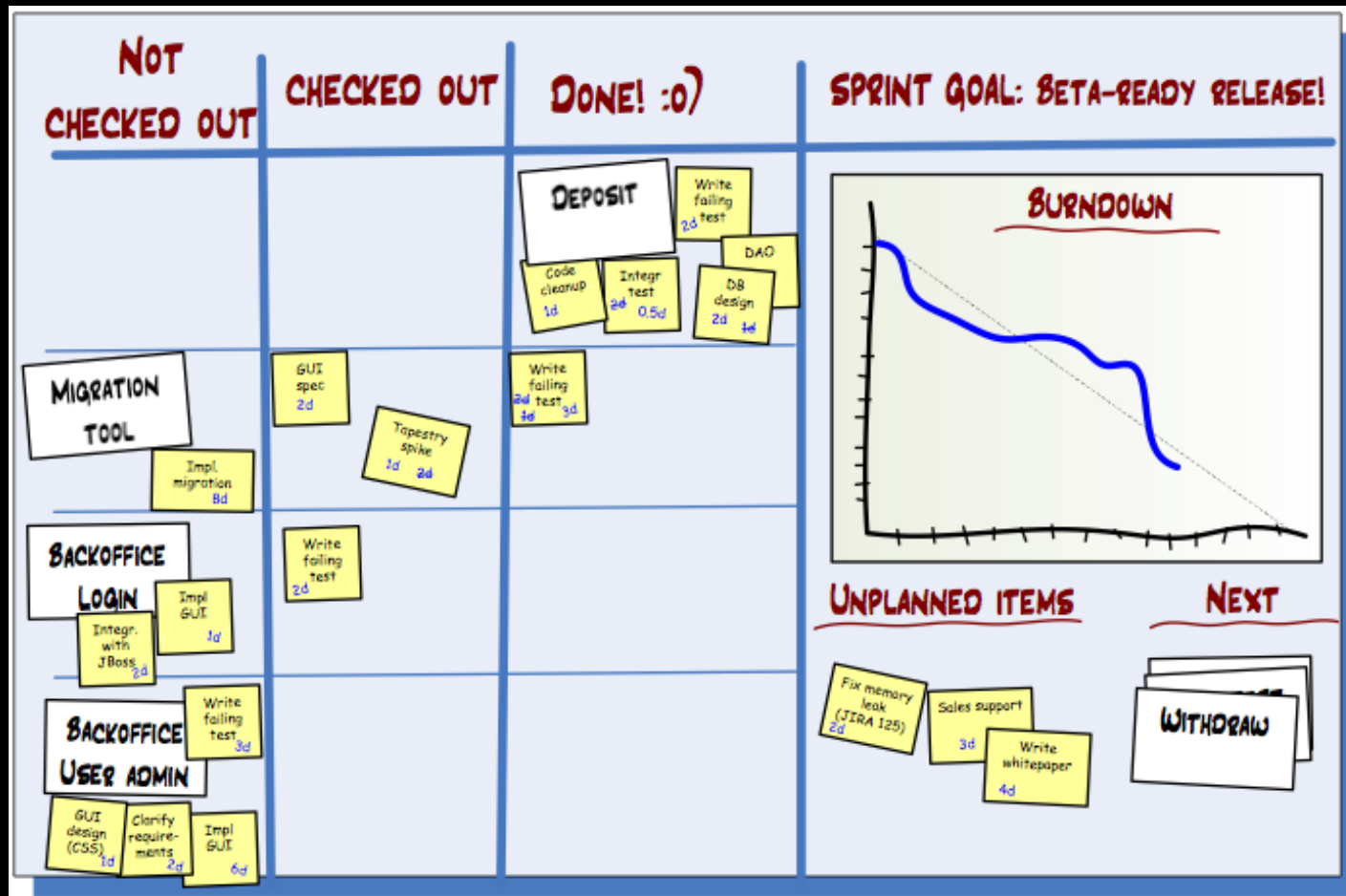
- Lista priorizada de requisitos do projeto
- Criado, gerenciado e desenvolvido pelo Product Owner
- Detalhado durante o projeto

User Story	Priority	Estimate
As a new player, I can create an account	H	8
As an existing player, I can log in to play a game	H	5
As a player, I can reset my password	M	3
As an administrator, I can view account records	M	3
As a player, I can choose from a list of all games in order to choose which I want to play	M	2
As a player, I can deposit my winnings in the system bank in order to use them later	H	5
As a player, I can transfer my winnings to my real bank account	H	13

# Task Board



# Task Board



# Burndown Chart

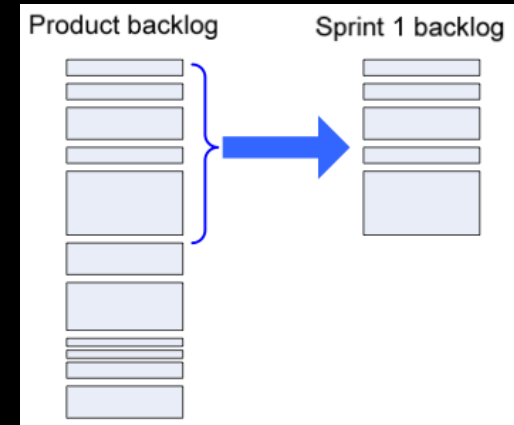
- Permite medir o progresso
- Mostra a quantidade de trabalho restante no Sprint a cada dia
- Atualizado a cada dia pelos membros do time

# Cerimônias Scrum

- Moderadas pelo Scrum Master
- Devem ter seu tempo reduzido à medida que o time e a organização amadurecem os valores e os processos
- Cerimônias
  - Release Planning
  - Sprint Planning
  - Daily Scrum
  - Sprint Review
  - Sprint Retrospective

# Sprint Planning

- Quem participa?
- Quanto tempo?
- Reunião de planejamento da sprint
  - PO explica a meta e resume o product backlog
  - Time faz as estimativas
  - Time seleciona as histórias para a sprint
- Deve dar ao time informações suficientes para que eles possam trabalhar na sprint
- O Product Owner já priorizou as histórias, ele ainda precisa participar desta reunião?



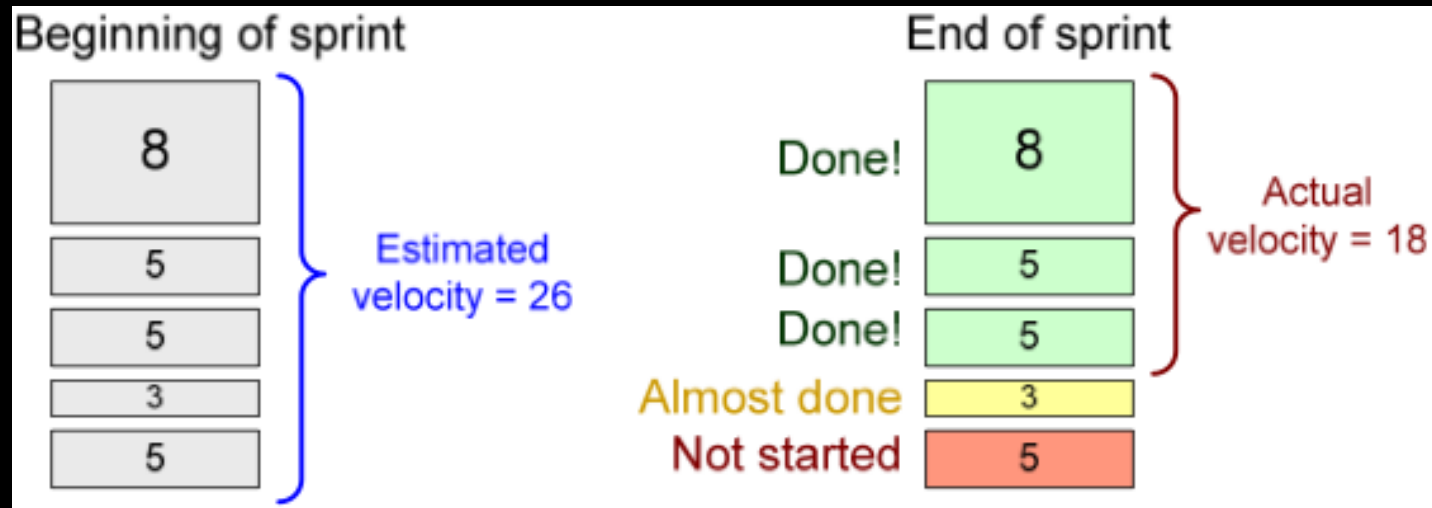


# Sprint Planning

- Como o time decide quais estórias incluir na sprint?
- Como o Product Owner pode afetar em quais estórias devem ser feitas na sprint?

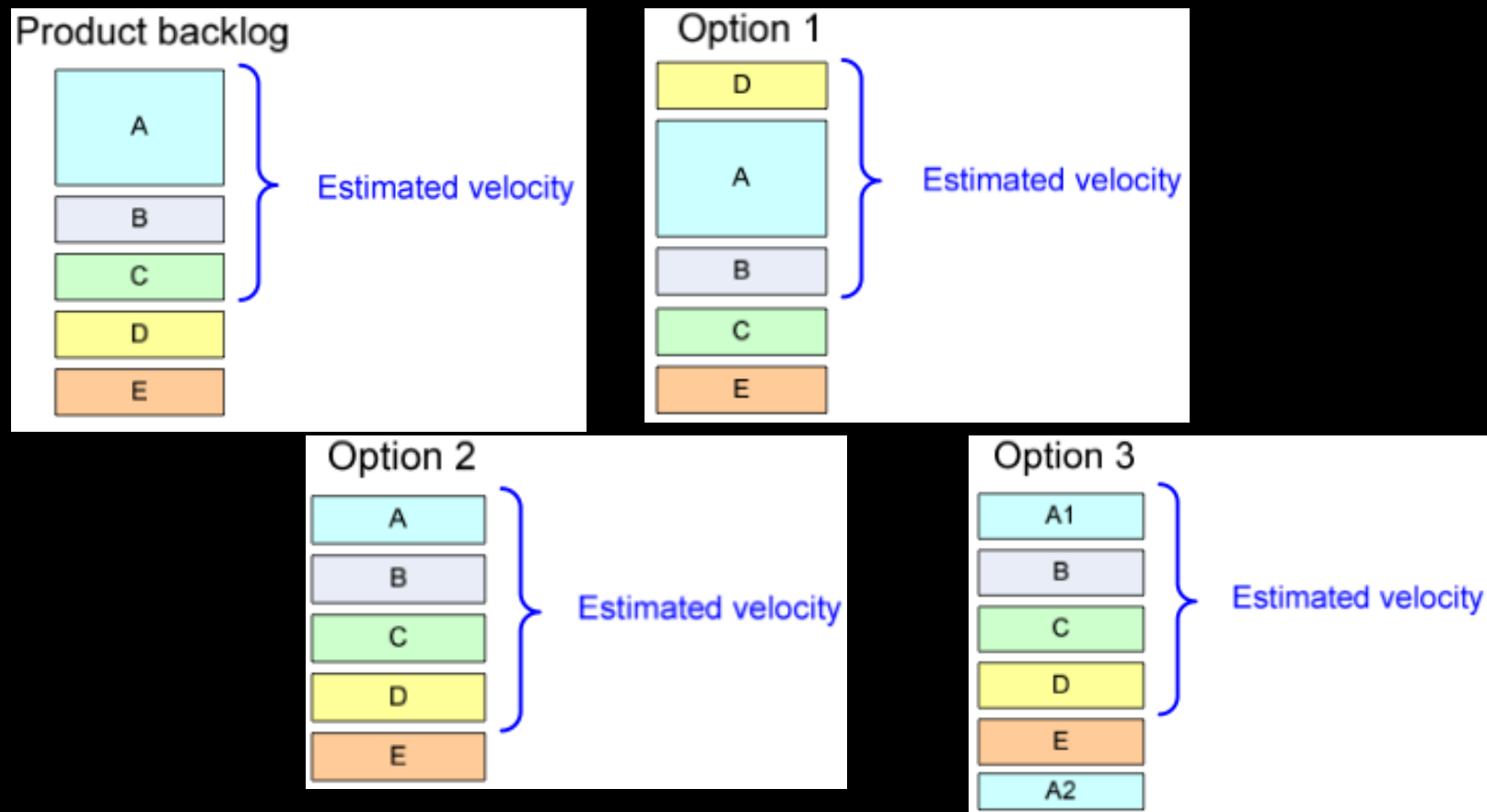
# Sprint Planning

- Como o time decide quais estórias incluir na sprint?
  - Velocidade do time
    - Quantidade de trabalho que pode ser feito
  - Calcula-se quais estórias podem ser adicionadas sem extrapolar a velocidade



# Sprint Planning

- Como o Product Owner pode afetar em quais estórias devem ser feitas na sprint?



# Daily Scrum

- Quem participa?
- Quanto tempo?
- Melhora a comunicação
- Identifica e remove impedimentos para o desenvolvimento
- Melhora o nível de conhecimento de todos acerca do projeto: compartilhamento do conhecimento.
- Técnicas para amadurecimento do time
  - Speech Token
  - Ampulheta

# Daily Scrum



O que você fez ontem?



O que você vai fazer hoje?



Tem alguma coisa te  
atrapalhando?

# Sprint Review

- Quem participa?
- Quanto tempo?
- A equipe apresenta os resultados obtidos durante o Sprint
- Normalmente assume a forma de uma demonstração de novas funcionalidades
- Sugestão: pode-se convidar integrantes da organização que possam dar um feedback positivo como testadores, usuário final

# Sprint Retrospective

- Quem participa?
- Quanto tempo?
- Ocorre ao final de um *Sprint* e serve para identificar o que funcionou bem, o que pode ser melhorado e que ações serão tomadas para melhorar

# Story Points

- O tamanho de uma User Story
- Influenciado por
  - O quanto difícil é a estória
  - Qual o tamanho do trabalho
  - Série de Fibonacci ou similar
    - 1,2,3,5,8,13,21,...
  - Comparação relativa

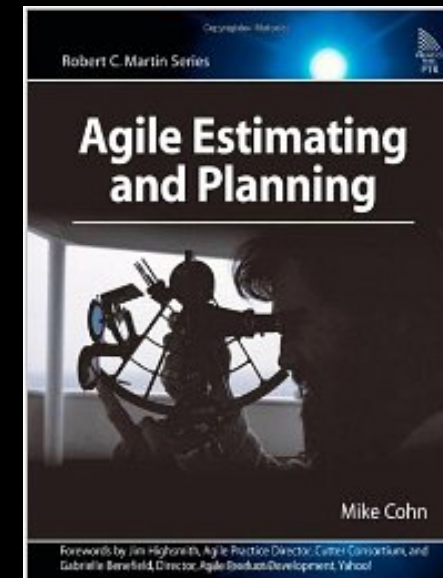


# Story Points

- Usualmente, para cada estória do *backlog* deverá ser atribuído um valor da série utilizada (Por exemplo: 1,2,3,5,8,13,21,...)
- Essa atribuição deve ser feita pelo time e não por uma única pessoa
- Estratégia
  - Para começar a pontuar as estórias de um *product backlog*, pega-se a estória que o time julga ser a de menor esforço e atribui pontuação 2
  - As demais estórias deverão seguir uma pontuação relativa a essa primeira

# Planning Poker

- Método ágil para estimativas, descrito inicialmente por James Grenning em 2002 e depois popularizado por Mike Cohn no seu livro *Agile Estimating and Planning*
- Para "jogar" o *Planning Poker* precisamos apenas de uma lista de *backlog* e um baralho
  - O baralho, geralmente composto por cartas que possuem uma sequência Fibonacci ou uma sequência similar (Exemplo: 0, 1, 2, 3, 5, 8, 13, 21, 34, 50, 80, 100, ?), uma carta opcional com um sinal de interrogação,..
    - Por que não utilizar a sequência de números naturais?
    - Para que serve o zero e o ponto de interrogação?



# Planning Poker

- O jogo começa após as explicações feitas pelo *Product Owner*, onde todos os membros discutem para chegar a um entendimento do problema
  - É importante frizar que os membros do time devem estimar todo o trabalho relacionado ao item do *backlog*, e não apenas a tarefa específica
- Por exemplo, se o item do *backlog* é "implementar testes", não se deve preocupar com os



# Planning Poker

- Cada membro da equipe, pega uma carta que acha que corresponde com os *Story points* e depois que todos escolheram suas cartas, tem inicio a rodada

Membro do time	Rodada 1	Rodada 2
Maria	80	50
Carlos	21	34
João	34	50
Rafael	50	34
Rita	50	50

# Story Points

- Vantagens
  - Foco em estimativa relativa
  - Desempenhos individuais diferentes
    - Um problema inerente da abordagem homem/dia é que essa medida depende do desempenho de quem está executando
    - Os indivíduos diferem em know-how, experiência, competência, etc
  - Foco em tamanho/esforço e não em duração

# Story Points

- Vantagens
  - Usar homem/dia sempre leva a um fator de ajuste
    - Quando se usa essa medida acaba-se por contar com um dia cheio de trabalho o que de fato não acontece
      - Sempre existem as paradas eventuais: emails, cafezinho, conversa paralela, notícias
  - Obrigatoriamente é necessário fazer um ajuste no homem/dia, um fator pelo qual você deve multiplicar seu homem/dia real em homem/dia produtivo
    - Por exemplo, Kniberg da Crisp [1] concluiu que para o seu time o fator de ajuste *é de 70%*

# Story Points

- Vantagens
  - Com homem/dia a aceleração da equipe pode ficar mascarada
    - À medida que o seu time vai ganhando experiência no projeto, tarefas de esforço similares tendem a diminuir o tempo necessário para a execução
    - Na abordagem homem/dia, o resultado é uma diminuição na estimativa em horas de novas tarefas e o ganho de produtividade não ficará explícito (ruim para a auto-estima e valorização do time)
    - Story Points: Mais pontos por sprint

# Story Points

- Vantagens
  - Entrada de novo membro na equipe
    - Pode fazer com que a equipe perca um pouco da sua velocidade na primeira sprint
    - Em geral leva-se 3 sprints para voltar a estabilizar a velocidade de referência
    - Com homem/dia pode haver um aumento irreal de produtividade prevista, pois aumenta-se o número de horas trabalhadas



# Story Points

- Vantagens
  - Evita-se pressões externas sobre as estimativas
  - Lei de Parkson
    - A lei de Parkson é um problema clássico no gerenciamento de pessoas
    - A lei sentencia que “o trabalho se expande para preencher o tempo disponível para sua realização”
    - Ou seja, se você der 5 dias para uma pessoa realizar uma tarefa ela será realizada no tempo estipulado, mesmo se pudesse ter sido feita em 1 dia
    - A lei de Parkson pode ser evitada se usarmos pontos ao invés de dias nas estimativas, pois se desacopla a execução de uma story a uma data exata de término (apenas a sprint tem data para terminar)
      - É claro que ela também poder ser evitada em times altamente comprometidos

# Story Points

- Desvantagens
  - Medida não universal
    - Medir em pontos é uma coisa muito particular e subjetiva, o seu significado acaba fazendo sentido apenas para um time em um determinado projeto
    - Isso pode atrapalhar ao se juntar grupos diferentes em um mesmo projeto com um *product backlog* em comum

# Story Points

- Desvantagens
  - Desconforto inicial de alguns
    - Pode ser difícil convencer algumas pessoas
      - Especialmente porque todos sabem que ao final os pontos vão se transformar em estimativas de tempo
  - Após algumas poucas *sprints*, cria-se uma noção intuitiva de pontos
    - Ao ser confrontado com uma nova estória, os indivíduos do time já começam a pensar se ela custa oito ou cinco pontos e não se vai durar x ou y dias

# Velocidade

- Para fazer um *Release Planning*, precisamos saber ou ter uma ideia da **Velocidade**
- 3 formas de termos a Velocidade
  - Média das Velocidades anteriores
  - Faça 1 ou 2 Sprints e veja o
  - Faça uma previsão

