



Scrum

Gestão Ágil de Projetos

Fernando Pedrosa – fpedrosa@gmail.com

Bibliografia

- ▶ Schwaber, Ken. Scrum Guide.
Disponível em: www.scrumguides.org
- ▶ Sabbagh, Rafael. *Scrum – Gestão Ágil para Projetos de Sucesso*. Editora: Casa do Código Edição: 1

Por que métodos tradicionais não funcionam?

Porque pressupõem que os requisitos ficarão estáveis ao longo do projeto

- ▶ Mas os requisitos nunca são bem compreendidos no início do projeto
- ▶ Eles mudam frequentemente durante o processo de desenvolvimento
- ▶ Usuários só sabem exatamente o que querem após ver o produto funcionando

O Movimento Ágil

- ▶ Motivação: insatisfação com os excessos dos métodos tradicionais, considerados “pesados”
- ▶ Métodos ágeis buscam flexibilizar o desenvolvimento de software
 - Focam no código, e não no projeto
 - São baseados em abordagens iterativas
 - Têm o objetivo de entregar software funcionando o mais rápido possível

Manifesto Ágil (2001)

“Estamos evidenciando maneiras melhores de desenvolver software fazendo-o nós mesmos e ajudando outros a fazê-lo. Através desse trabalho, passamos a entender que:

- ▶ **Indivíduos e interações** são mais importantes que processos e ferramentas.
- ▶ **Software funcionando** é mais importante do que documentação completa e detalhada.
- ▶ **Colaboração com o cliente** é mais importante do que negociação de contratos.
- ▶ **Adaptação a mudanças** é mais importante do que seguir o plano inicial.

Ou seja, mesmo tendo valor os itens à direita, valorizamos mais os itens à esquerda. “

Princípios Ágeis

1. *“Nossa maior prioridade é satisfazer o cliente por meio da entrega cedo e frequente de software com valor”*
2. *“Mudanças de requisitos são bem-vindas, mesmo em fases tardias do projeto”*
3. *“Entregar software em funcionamento com frequência, desde a cada duas semanas até a cada dois meses, com uma preferência por prazos mais curtos”*

Princípios Ágeis

4. *“As pessoas do negócio e os desenvolvedores devem trabalhar em conjunto diariamente ao longo do projeto”*
5. *“Construa projetos em torno de indivíduos motivados. [...] confie neles para realizarem o trabalho”*
6. *“O método mais efetivo de se transmitir informação para e entre uma equipe de desenvolvimento é a conversa face a face”*

Princípios Ágeis

- 7. *“Software em funcionamento é a principal medida de progresso.”*
- 8. *“Os processos Ágeis promovem o desenvolvimento sustentável. Os patrocinadores, desenvolvedores e usuários devem ser capazes de manter indefinidamente um ritmo constante”*
- 9. *“A atenção contínua à excelência técnica e a um bom projeto aumentam a agilidade.”*

Princípios Ágeis

- 10. *“Simplicidade – a arte de se maximizar a quantidade de trabalho não feito – é essencial.”*
- 11. *“ As melhores arquiteturas, requisitos e projetos emergem de equipes que se auto-organizam”*
- 12. *“Em intervalos de tempo regulares, a equipe reflete sobre como se tornar mais efetiva [...] e se ajusta de acordo”*

Exercícios [1]

(ANTAQ – CESPE 2014)

A principal prioridade dos processos e práticas ágeis de desenvolvimento de *software* é satisfazer o cliente com entregas rápidas e contínuas, ainda que haja mudanças de requisitos nas etapas finais do projeto.

(STF – CESPE 2013)

As características básicas pertinentes aos métodos ágeis incluem a dependência de requisitos bem definidos, que não devem ser constantemente alterados, e a dependência do uso de UML para a modelagem.

(CNJ – CESPE 2013)

O desenvolvimento ágil de sistemas consiste em uma linguagem de modelagem que permite aos desenvolvedores visualizarem os produtos de seu trabalho em gráficos padronizados.

Exercícios [2]

(MPE/AP – FCC 2012)

Os processos de desenvolvimento rápido de *software* são concebidos para produzir, rapidamente, *softwares* úteis. O *software* não é desenvolvido como uma única unidade, mas como uma série de incrementos, em que cada incremento inclui uma nova funcionalidade no sistema. Embora existam muitas abordagens para o desenvolvimento rápido de *software*, elas compartilham algumas características fundamentais, como

a) a definição de requisitos e alterações no sistema, que devem ser definidas antes do início do processo, podendo ser alteradas somente após a total entrega e aceite do produto, trazendo agilidade ao processo, pois a equipe de desenvolvimento pode trabalhar sem a interferência do usuário final.

Exercícios [2]

b) os processos de especificação, projeto e implementação, que são criados em uma única etapa do sistema, diferindo de processos tradicionais de desenvolvimento de *software*, que os intercalam e reduzem a velocidade em que o *software* é produzido.

c) a especificação detalhada de todo o projeto, que contém documentos de requisitos elaborados para cada detalhe funcional e não funcional e também de cada novo item incluído nas etapas de iteração.

d) a diminuição do número de versões, que aumenta a quantidade de itens funcionais incluídos em cada entrega. A especificação e avaliação de cada versão são exclusivas da equipe de desenvolvimento, fato que contribui para o aumento da velocidade e rapidez do processo.

e) as interfaces de usuário do sistema, que geralmente são desenvolvidas com o auxílio de um sistema interativo que permite a criação rápida do projeto da interface por meio de desenho e inserção de ícones.

Scrum



Onde tudo começou?

- ▶ Década de 90 – Jeff Sutherland & Ken Schwaber – “SCRUM Software Development Process”
- ▶ Em 2001, ambos estavam entre os 17 proponentes do Manifesto Ágil
- ▶ A Scrum Alliance é criada, junto com o programa de certificação Scrum Master
- ▶ Hoje, Scrum é método de trabalho adotado por dois em cada três profissionais em projetos de desenvolvimento de software

O que é?

- ▶ Framework ágil, simples e leve, para a gestão do desenvolvimento de produtos complexos
- ▶ Iterativo e incremental
 - Entrega valor com frequência e reduz os riscos do projeto
- ▶ Baseado no empirismo
 - Experiências práticas

O que NÃO é?

- ▶ Não é uma metodologia ou conjunto de processos
 - É um Framework – uma estrutura básica que serve de guia
- ▶ Não é prescritivo
 - Não há regras sobre como executar as tarefas
 - São prescritos apenas 3 papeis, 3 artefatos e 5 cerimônias
- ▶ Não é fácil de aplicar, ainda que simples

Benefícios

- ▶ Entregas frequentes que dão retorno ao investimento de clientes
- ▶ Redução de riscos do projeto
- ▶ Maior qualidade no produto gerado
- ▶ Mudanças utilizadas como vantagem competitiva
- ▶ Visibilidade do progresso do projeto
- ▶ Redução do desperdício
- ▶ Aumento de produtividade

Aplicabilidade

- ▶ Scrum é a solução para tudo?
- ▶ Quando utilizar Scrum?
- ▶ E quando NÃO utilizar?

Teoria da Complexidade

- ▶ Scrum foi criado para o desenvolvimento de projetos “complexos”
- ▶ Mas o que é algo complexo?

Modelo Cynefin



Contexto Simples

- ▶ Domínio da previsibilidade, onde as relações são bem conhecidas
- ▶ Dada uma situação, deve-se:
 - Identificar os fatos (perceber)
 - Definir que tipo de situação se trata (categorizar)
 - Aplicar a melhor prática conhecida (responder)

Contexto Complicado

- ▶ Domínio onde existe uma relação entre causa e efeito, mas ela não é óbvia para todos
- ▶ Pode haver várias respostas viáveis para o mesmo problema
- ▶ Dada uma situação, deve-se:
 - Identificar os fatos (perceber)
 - Definir qual é a solução mais adequada (analisar)
 - Aplicar a solução (responder)

Contexto Complexo

- ▶ Domínio onde a relação entre causa e efeito só pode ser percebida retrospectivamente
- ▶ Não é possível prever o que irá acontecer, e por isso se faz necessária uma abordagem empírica
- ▶ A grande maioria dos projetos de Des. de Software se enquadra aqui
 - Há grande imprevisibilidade e potencial de mudanças!

Contexto Complexo (cont.)

- ▶ Para uma dada situação, deve-se:
 - Investigar os fatos por meio de experimentos (investigar)
 - Entender seus impactos (perceber)
 - Aplicar o que funciona e descartar o que não funciona (responder)

Contexto Caótico

- ▶ Domínio onde a relação entre causa e efeito é impossível de ser determinada
- ▶ Não há padrões, apenas turbulência
- ▶ Para uma dada situação, deve-se:
 - Agir rapidamente (agir)
 - Entender onde existe alguma estabilidade (perceber)
 - Trabalhar para trazer a situação do caos para algo mais facilmente gerenciável (responder)

Contexto de Desordem

- ▶ Nesse contexto, não conseguimos nem definir com que tipo de problema ou sistema estamos lidando
- ▶ Normalmente acontece quando o problema é grande demais ou não há nenhuma compreensão sobre ele
- ▶ Nesse caso, o recomendado é quebrar o problema em partes e definir o contexto de cada uma delas

Abordagem tradicional

- ▶ Métodos tradicionais de gestão de projetos os tratam como se fossem simples ou, no máximo, complicados
- ▶ Supõe-se que é possível estabelecer uma relação clara entre causa e efeito e traçar um plano com ações bem definidas
- ▶ Mas a grande maioria dos projetos de software é de domínio COMPLEXO!

Abordagem Scrum

- ▶ Scrum entende que o desenvolvimento de software deve ser realizado de forma empírica
- ▶ A base é a experiência prática e não teorias

Exercícios [3]

(TRE/BA – CESPE 2010)

Um princípio chave do Scrum é o reconhecimento de que desafios fundamentalmente empíricos não podem ser resolvidos com sucesso utilizando-se uma abordagem tradicional de controle. O Scrum adota uma abordagem empírica, aceitando que o problema não pode ser totalmente entendido ou definido, focando na maximização da habilidade da equipe de responder de forma ágil aos desafios emergentes.

(TRE/BA – CESPE 2011)

Entre as metodologias ágeis para o desenvolvimento de software, o Scrum permite a criação de equipes auto-organizadas e, conseqüentemente, possibilita o incentivo à comunicação verbal entre todos os membros da equipe. Da mesma forma que as abordagens típicas de Project Management Body of Knowledge ou PRINCE2, o Scrum caracteriza-se por apresentar uma abordagem elementar do gerenciamento de projetos.

Exercícios [4]

(ANVISA – CETRO 2013)

Leia o parágrafo abaixo, relacionado à metodologia Scrum, e, em seguida, assinale a alternativa que preenche correta e respectivamente as lacunas.

É um(a) _____ dentro do(a) qual as pessoas podem tratar de problemas complexos e adaptativos e resolvê-los, enquanto produtiva e criativamente entregam produtos com o mais alto valor possível.

Fundamentado _____ de controle de processo, o Scrum emprega uma abordagem _____ para aperfeiçoar a previsibilidade e o _____.

a) estrutura processual/ na experimentação/ estática/ controle de pessoas

b) técnica/ na experiência científica/ iterativa e incremental/ controle de riscos

Exercícios [4]

(ANVISA – CETRO 2013)

É um(a) _____ dentro do(a) qual as pessoas podem tratar de problemas complexos e adaptativos e resolvê-los, enquanto produtiva e criativamente entregam produtos com o mais alto valor possível.

Fundamentado _____ de controle de processo, o Scrum emprega uma abordagem _____ para aperfeiçoar a previsibilidade e o _____.

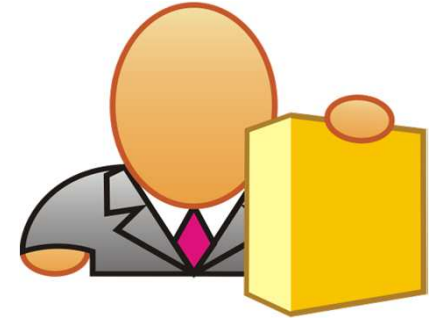
c) processo/ nas teorias empíricas/ estática/ controle de pessoas

d) processo/ na experimentação/ iterativa e incremental/ controle de pessoas

e) estrutura processual/ nas teorias empíricas/ iterativa e incremental/ controle de riscos

O Time de Scrum (Papeis e Responsabilidades)

Product Owner



- ▶ Define as funcionalidades do produto
- ▶ Decide as datas de lançamento e conteúdo
- ▶ Prioriza as funcionalidades de acordo com o valor para a empresa
- ▶ Aceita ou rejeita os resultados dos trabalhos

Product Owner (Características)

Único

- ▶ O papel de PO deve ser exercido por uma única pessoa
- ▶ O PO não é um comitê ou departamento da organização
- ▶ Ele pode trabalhar com vários times, mas cada time pode ter apenas um PO
- ▶ Substituições eventuais podem acontecer, mas devem ser exceções

Product Owner (Características)

Disponível

- ▶ O PO deve estar disponível sempre que necessário
- ▶ Ele deve:
 - Participar de todas as cerimônias cuja sua presença seja necessária
 - Tirar dúvidas sobre o desenvolvimento do produto a qualquer momento
 - Interagir com os demais *stakeholders* para entender as necessidades de negócio

Product Owner (Características)

Conhecimento do Negócio

- ▶ O PO deve passar ao time de desenvolvimento todo o conhecimento de negócio necessário à construção do produto
- ▶ Cuidado: o PO não é o “*Superman*”, ele não precisa saber de tudo, mas é esperado que ele busque as respostas necessárias

Product Owner (Tarefas)

Gerenciar o Produto

- ▶ É de responsabilidade do PO definir os objetivos do projeto
- ▶ Ele define as necessidades e as prioriza em uma lista chamada “Product Backlog”
- ▶ O PO define a Visão do Produto e o Roadmap do Produto

Product Owner (Tarefas)

Gerenciar as partes interessadas

- ▶ O PO deve identificar corretamente que são os clientes e pessoas relevantes do projeto
- ▶ Ele deve entender as diversas necessidades, muitas vezes conflitantes, e gerenciar as expectativas dos *stakeholders*

Product Owner (Tarefas)

Gerenciar Releases

- ▶ Uma Release acontece quando o PO decide disponibilizar o produto aos usuários finais em seu ambiente-alvo
- ▶ A decisão de fazer uma Release é influenciada por vários fatores
- ▶ É de responsabilidade do PO decidir quando e como será feita uma Release

Product Owner (Tarefas)

Planejar Sprints

- ▶ Junto com o Time de Desenvolvimento, o PO decide que itens farão parte de uma determinada Sprint
- ▶ Ele é responsável por definir uma Meta da Sprint

Product Owner (Tarefas)

Colaborar com o Time de Desenvolvimento

- ▶ Durante toda a Sprint, o PO soluciona dúvidas do Time de Desenvolvimento
- ▶ Além disso, ele faz o Refinamento de Backlog

Product Owner (Tarefas)

Aceitar ou rejeitar resultados

- ▶ Na entrega da Sprint, o PO aceita ou rejeita os resultados gerados pelo Time de Desenvolvimento
- ▶ Apenas os itens “prontos” serão aceitos
- ▶ O PO comunica claramente se a Meta da Sprint foi alcançada ou não

Exercícios [5]

(INPI – CESPE 2013)

No Scrum, o *Product Owner* (PO) é responsável por definir a visão do produto e remover os impedimentos, enquanto o Scrum Master (SM) é responsável por elaborar e manter o *Product Backlog*, bem como por ajudar o PO a executar suas atividades diárias.

(Banco da Amazônia – CESPE 2012)

O escopo, a importância e a estimativa de um *Sprint* do *Scrum* são definidos pelo *product owner*.

(ANAC – CESPE 2012)

O único papel definido pelo *Scrum* com autoridade para cancelar uma *Sprint* é o do *product owner*.

Exercícios [6]

(MPE/AP – FCC 2012)

Na metodologia de desenvolvimento SCRUM, o proprietário do produto (Product Owner) é responsável por maximizar o valor do produto e o trabalho da equipe de desenvolvimento. O proprietário do produto é a única pessoa responsável pela manutenção do Backlog do produto. Este gerenciamento inclui

- a) assegurar que a equipe de desenvolvimento compreenda os itens do *Backlog* do produto no nível necessário.
- b) encontrar técnicas para a manutenção efetiva do *Backlog* do produto e transmitir essas técnicas para a equipe de desenvolvimento.
- c) comunicar, nas reuniões diárias, as metas e itens do *Backlog* do produto para a equipe de desenvolvimento.
- d) treinar o time Scrum para que crie, de forma clara e precisa, os itens do *Backlog* do produto.
- e) entender o planejamento do produto a longo termo e de forma empírica.

Time de Desenvolvimento



- ▶ Contém tipicamente entre 3 e 9 pessoas
- ▶ Multifuncional e auto-organizado
 - Programadores, testadores, desenvolvedores de interfaces, etc.
- ▶ Orientado à excelência técnica
- ▶ Focado nas metas estabelecidas pelo PO

Time de Desenvolvimento (Características)

Multidisciplinar

- ▶ O time possui em seus membros todas as habilidades necessárias para desenvolver o produto
- ▶ Não significa que todos têm que saber de tudo – é possível haver especialistas

Time de Desenvolvimento (Características)

Auto-organizado

- ▶ O time planeja e gerencia o seu próprio trabalho
- ▶ Não há gerentes de projeto dizendo como o time deve trabalhar
- ▶ Não é anarquia – ainda há regras!
- ▶ Para que a auto-organização funcione, o time deve se comunicar muito bem (mais uma razão para mantê-lo pequeno)

Time de Desenvolvimento (Tarefas)

Planejar o trabalho técnico

- ▶ O time de desenvolvimento é responsável por estimar o esforço necessário para implementar cada item do backlog

Time de Desenvolvimento (Tarefas)

Desenvolver o produto

- ▶ O “desenvolvimento” do produto inclui todas as tarefas necessárias para entregar o incremento do produto
 - Programação, testes, documentação, etc.
- ▶ O objetivo é alcançar a Meta proposta pelo PO
 - O que não significa, necessariamente, desenvolver todo os itens da Sprint

Time de Desenvolvimento (Tarefas)

Identificar e informar impedimentos

- ▶ Impedimentos são barreiras ou obstáculos que dificultam significativamente o trabalho do time
- ▶ A sua resolução está fora do alcance do time ou lhe tomaria muito tempo
- ▶ Atenção: problemas que podem ser resolvidos pelo time em um tempo razoável NÃO são impedimentos

Exercícios [7]

(ANP – CESPE 2013)

De acordo com a metodologia Scrum, a constituição ideal da equipe de desenvolvimento para que o trabalho se mantenha ágil deve ser de menos de três pessoas.

Exercícios [8]

(ANVISA – CETRO 2013)

Quanto à metodologia Scrum e no que diz respeito às características das Equipes de Desenvolvimento, é **incorreto** afirmar que:

- a) não há exceções para a regra de que o único título dos integrantes é o de Desenvolvedor, independentemente do trabalho que está sendo realizado pela pessoa.
- b) são auto-organizadas e nem mesmo o Scrum Master diz à Equipe de Desenvolvimento como transformar o Backlog do Produto em incrementos de funcionalidades potencialmente utilizáveis.
- c) são multifuncionais, possuindo todas as habilidades necessárias, enquanto equipe, para criar o incremento do Produto.

Exercícios [8]

- d) Individualmente os integrantes podem ter habilidades especializadas e área de especialização, mas a responsabilidade pertence à Equipe de Desenvolvimento como um todo.
- e) elas contêm subequipes dedicadas a domínios específicos de conhecimento, tais como teste ou análise de negócios.

Scrum Master



- ▶ Responsável pela aplicação dos valores e práticas Scrum
- ▶ Remove obstáculos, facilita resultados
- ▶ Garante a plena funcionalidade e produtividade da equipe
- ▶ Escudo para interferência externas

Scrum Master (Características)

Competente em *soft skills*

- ▶ *Soft Skills* = Habilidades Interpessoais
- ▶ O Scrum Master deve ter habilidade para resolver conflitos e promover as mudanças necessárias para alcançar os objetivos do projeto
- ▶ Habilidade de comunicação é essencial, assim como habilidade política e negociação

Scrum Master (Características)

Neutro

- ▶ Como facilitador, o Scrum Master deve ser o mais neutro possível
- ▶ Ele deve trabalhar para Time como um todo, e não apenas para os desenvolvedores
- ▶ É possível que o Scrum Master seja um dos desenvolvedores (porém, não recomendado)
- ▶ O Scrum Master nunca deve ser, ao mesmo tempo, o Product Owner

Scrum Master (Tarefas)

Facilitar o trabalho

- ▶ O Scrum Master ajuda todos a trabalharem de forma mais efetiva
- ▶ Ele pode aconselhar, dar autonomia e estimular o Time
 - Interferir diretamente não é recomendado
- ▶ Como um especialista em Scrum, ele deve identificar rapidamente o que não está funcionando
- ▶ Ironicamente, o grande objetivo do Scrum Master é se tornar desnecessário!

Scrum Master (Tarefas)

Remover impedimentos

- ▶ Impedimentos podem paralisar o trabalho da equipe
- ▶ O Scrum Master deve tomar ações rápidas e efetivas para resolver a causa-raiz do impedimento
- ▶ É importante diferenciar impedimentos de problemas que podem ser resolvidos pelo Time de Desenvolvimento

Scrum Master (Tarefas)

Promover mudanças organizacionais

- ▶ O Scrum Master atua como um agente de mudanças na organização
- ▶ As mudanças podem incluir:
 - Treinar pessoas em Scrum
 - Mudar regulamentos e a cultura organizacional
 - Promover eventos de integração e sensibilização sobre métodos ágeis

Scrum Master (Tarefas)

Garantir o uso do Scrum

- ▶ O Scrum Master garante que os valores e práticas do Scrum estão sendo seguidos por todos
- ▶ É de responsabilidade do Scrum Master que todas as cerimônias aconteçam dentro de suas *timeboxes*
- ▶ No entanto, ele não deve ser a Polícia do Scrum, mas alguém que consegue convencer e sensibilizar

Exercícios [9]

(TRE/BA – CESPE 2010)

A metodologia Scrum é facilitada por um scrum master, que atua como um mediador entre a equipe e qualquer influência desestabilizadora, além de assegurar que a equipe esteja utilizando corretamente as práticas de Scrum, motivando e mantendo o foco na meta da sprint.

(TCE/RO – CESPE 2013)

Na metodologia Scrum, a equipe trabalha nos processos e não há cargos na equipe. Como um dos papéis necessários, o Scrum master deve garantir que o processo seja entendido e atuar como facilitador para ajudar a equipe.

Artefatos e Produtos de Trabalho



Artefatos e Produtos de Trabalho

Scrum

- Product Backlog
- Sprint Backlog
- Definition of Done
- Incremento do Produto
- Meta da Sprint

Extra Scrum

- Definition of Ready
- Gráficos de Acompanhamento



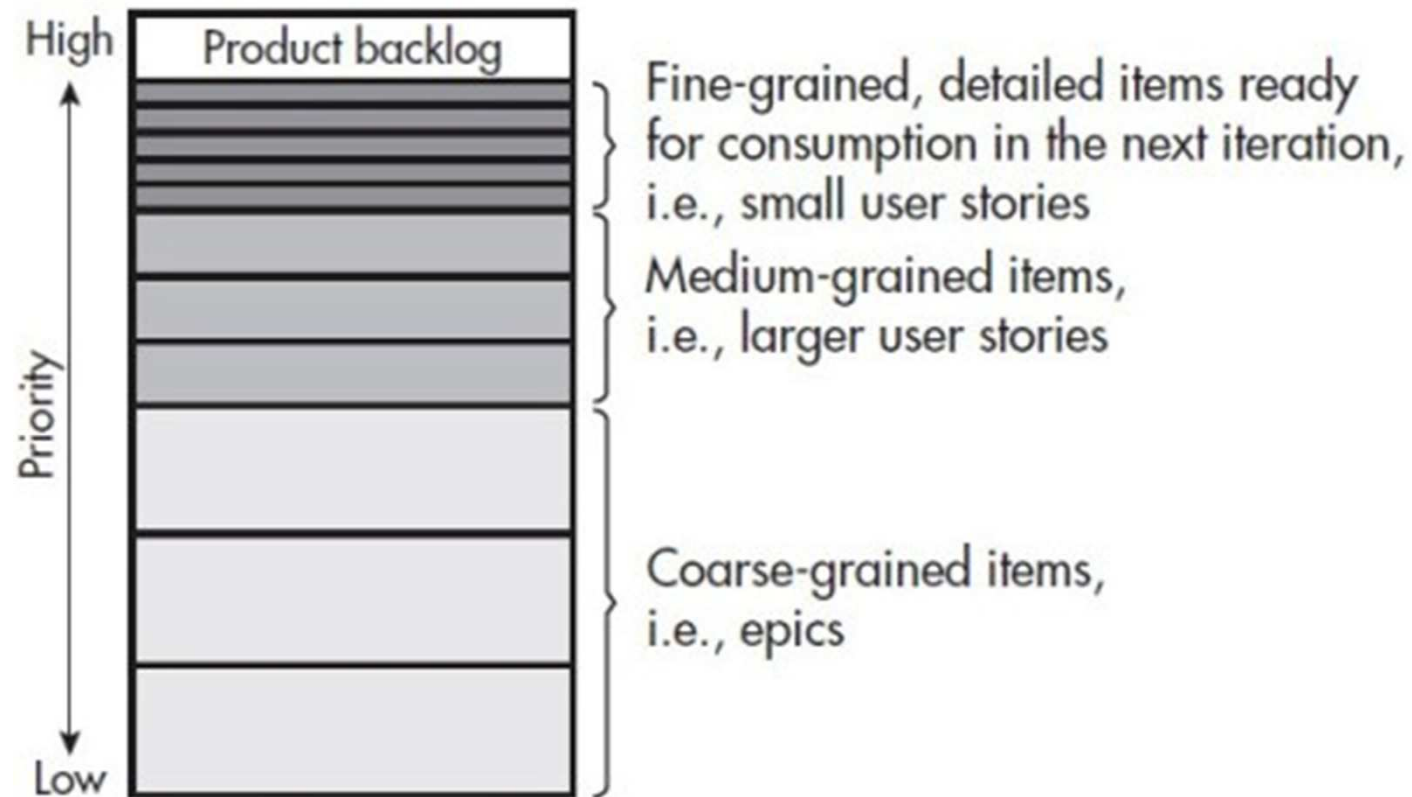
Product Backlog

- ▶ É uma lista de tudo aquilo que precisa ser feito durante o projeto
- ▶ Pode conter:
 - Necessidades ou objetivos de negócio
 - Questões arquiteturais e/ou técnicas
 - Melhorias e correções a serem realizadas no produto*

Product Backlog (cont.)

- ▶ É um artefato “vivo” e sofre constantes modificações
- ▶ O Product Owner é o único responsável por gerenciá-lo, podendo:
 - Adicionar, remover, atualizar e reordenar itens
 - Para isso, ele deve interagir com as demais partes interessadas no projeto
- ▶ Não há nenhum formato prescrito para o Product Backlog

Product Backlog (formato típico)



Product Backlog (características)

Ordenado

- ▶ Os itens do Product Backlog são ordenados de acordo com seu grau de importância
- ▶ Os itens mais acima possuem mais detalhes e importância maior
- ▶ A ordenação é um trabalho contínuo, podendo ser feita por:
 - Necessidade
 - Risco
 - Tecnologia
 - Oportunidade Financeira (ROI), etc.

Product Backlog (características)

Dinâmico

- ▶ O Product Backlog é dinâmico, sendo constantemente refinado e atualizado
- ▶ Ele é gradualmente detalhado
 - Os itens mais acima têm mais detalhes especificados e serão implementados em curto prazo
 - Os itens mais abaixo são apenas esboços
- ▶ Nunca está completo, está sempre evoluindo em um processo contínuo de descoberta

Product Backlog (características)

Planejável

- ▶ O Product Backlog deve permitir o planejamento e dar visibilidade sobre o que será feito em seguida
- ▶ Deve ser possível estabelecer escopo, velocidade de desenvolvimento e progresso do trabalho
- ▶ Para ajudar no planejamento, é feita uma estimativa de esforço sobre cada item

Exercícios [10]

(ANCINE – CESPE 2013)

Se for averiguado, em uma organização, que o Scrum master gerencia o backlog do produto, é correto afirmar que houve falha na execução de papéis, visto que cabe unicamente ao product owner gerenciar o backlog do produto.

(BACEN – CESPE 2013)

No SCRUM, o product owner é responsável por alterar o backlog da sprint durante a sprint.

(ABIN – CESPE 2013)

No SCRUM, um backlog consiste em uma lista de itens priorizados a serem desenvolvidos para um software. Essa lista é mantida no product owner, o qual pode alterá-la a qualquer momento, desde que os itens alterados não estejam na sprint backlog. Isso significa que product backlog e sprint backlog são estruturas similares.

Exercícios [1 1]

(MPE/AP – FCC 2012)

Na metodologia de desenvolvimento SCRUM, o proprietário do produto (Product Owner) é responsável por maximizar o valor do produto e o trabalho da equipe de desenvolvimento. O proprietário do produto é a única pessoa responsável pela manutenção do Backlog do produto. Este gerenciamento inclui

- a) assegurar que a equipe de desenvolvimento compreenda os itens do Backlog do produto no nível necessário.
- b) encontrar técnicas para a manutenção efetiva do Backlog do produto e transmitir essas técnicas para a equipe de desenvolvimento.
- c) comunicar, nas reuniões diárias, as metas e itens do Backlog do produto para a equipe de desenvolvimento.
- d) treinar o time Scrum para que crie, de forma clara e precisa, os itens do Backlog do produto.
- e) entender o planejamento do produto a longo termo e de forma empírica.

Estimativas

- ▶ Abordagens tradicionais costumam utilizar estimativas de “alta precisão”
- ▶ Na prática, essas estimativas são artificiais e raramente se concretizam
 - Situação conhecida como Lei de Parkinson (a famosa “Síndrome do Estudante”)
- ▶ No Scrum, quanto menos detalhes se conhece sobre um item, menor será a precisão de estimativa sobre ele

Unidades de Estimativas

▶ Tempo Real

- Dias ou horas reais sobre um item. É a unidade mais tradicional

▶ Tempo Ideal

- Dias ou horas de trabalho sobre um item considerando que nenhuma interrupção existirá

▶ Story Points

- Unidade relativa de tempo. É a mais utilizada por equipes ágeis

Story Points

- ▶ Utiliza uma escala relativa
- ▶ Entende-se que é mais fácil realizar estimativas fazendo comparações do que em termos absolutos
- ▶ Em vez de dizermos que determinada atividade terá “X tempo”, dizemos que ela tem o “dobro de duração” de outra atividade ou “metade do tempo”, etc.

Story Points (cont.)

- ▶ Em vez de utilizar horas ou dias, o Time utiliza uma escala adotada como referência
- ▶ Fibonacci (adaptada por Mike Cohn)
 - 1 2 3 5 8 13 20 40 100
- ▶ T-Shirt Sizing
 - PP P M G GG XG
- ▶ Uma vez selecionada a escala e o ponto de referência, as demais estimativas podem ser feitas

Story Points (cont.)

- ▶ As estimativas sobre os itens de backlog são feitas exclusivamente pelo Time de Desenvolvimento
- ▶ Regra de Ouro: “quem estima é quem faz”
- ▶ O tempo gasto para estimar deve ser aceitável (não pode ser muito demorado por si só)
 - Do contrário teríamos que estimar “o tempo para estimar o tempo”!

Exercícios [1 2]

(TRT4 – FCC 2011)

Para utilizar o processo de estimativa por Story Points em Scrum, inicialmente

- a) o Product Owner deve atribuir valores de negócio para cada um dos itens do Product Backlog.
- b) o Product Backlog deve considerar todos os fatores de Sprint contidos no Backlog Owner.
- c) os Stakeholders devem atribuir os riscos do Product Owner para cada Sprint Planning.
- d) os Stakeholders devem atribuir valores de negócio do Product Owner para cada Sprint.
- e) o Product Planning deve avaliar cada Sprint contida no Backlog transacional e decidir pela prioridade de atividades.

Exercícios [13]

(ANTAQ – CESPE 2014)

No SCRUM, cada ponto de história (PH) implica uma hora de trabalho de uma pessoa.

Planning Poker

- ▶ Técnica mais utilizada por equipes ágeis para estimativa de backlog
- ▶ Consiste em chegar a um consenso após algumas rodadas de estimativas
- ▶ O processo é o seguinte:
 - Dado um item de backlog, cada membro do time escolhe uma carta (estimativa) sem mostrá-la
Todos mostram suas cartas ao mesmo tempo
 - Os membros que deram as estimativas mais altas ou baixas discutem suas razões
 - Uma ou mais rodadas de estimativas são realizadas até se alcançar um consenso na equipe

Planning Poker (exemplo)



Velocidade do Time

- ▶ Conhecer a velocidade do Time de Desenvolvimento é importante para planejar os itens de backlog
- ▶ A velocidade é a taxa média de produção de “pontos ágeis” por Sprint
- ▶ Espera-se que a velocidade se mantenha estável, considerando a mesma equipe e tamanho de Sprint
- ▶ O valor é considerado uma referência

Como representar os itens do Product Backlog?

- ▶ Scrum não prescreve uma forma para descrever os itens de backlog, sendo possível usar diversas formas
- ▶ A técnica mais comum utilizada por equipes ágeis é a “User Story” (História de Usuário)
- ▶ Ela busca descrever uma necessidade de forma simples e leve

User Story (História de Usuário)

- ▶ User Story é uma descrição concisa de uma necessidade do usuário (ou seja, de um “requisito”)
- ▶ Utiliza o ponto de vista do usuário
- ▶ Sua estrutura padrão é QUEM, O QUE POR QUÊ
 - QUEM – define quem é o usuário
 - O QUÊ – define qual é a necessidade
 - POR QUÊ – define o benefício esperado

User Story (exemplos)

- ▶ *“Como um professor <quem>, eu posso fazer backup do meu HD <o que> para garantir a integridade dos meus arquivos <por quê>”*
- ▶ *“Como um estudante <quem>, eu posso comprar um Passe Estudantil <o que> para ir à escola <por quê>”*

Exercícios [14]

(MPE/CE – FCC 2013)

O Scrum é um modelo ágil para a gestão de projeto de software. No Scrum,

- a) o scrum team é a equipe de desenvolvimento com 6 a 10 pessoas, necessariamente dividida em papéis como analista, designer e programador.
- b) o scrum master é um gerente e um líder como nos modelos prescritivos, já que as equipes não são autoorganizadas.
- c) o product backlog precisa ser completo desde o início do projeto, contemplando todas as funcionalidades.
- d) as funcionalidades a serem implementadas em cada projeto (requisitos ou histórias de usuário) são mantidas em uma lista chamada de product backlog.
- e) o product owner define quais são os requisitos mais importantes a serem tratados em cada sprint, porém, não é o responsável pelo ROI (Return Of Investment), nem por avaliar as necessidades dos clientes.

Sprint Backlog

- ▶ O Sprint Backlog é uma lista de itens (tarefas) para o desenvolvimento do Incremento do Produto
- ▶ Também envolve um plano sobre como o trabalho será realizado
 - Normalmente expresso por uma lista de tarefas
- ▶ Pertence ao Time de Desenvolvimento, que é responsável pelo seu uso

Sprint Backlog (exemplo)

Item de Backlog

PBI	Todo	In Progress	Done

Fonte: Scrum.org

Tarefas

Definition of Ready (“Preparado”)

- ▶ Um item de backlog será considerado para desenvolvimento apenas quando ele estiver “preparado”
- ▶ A preparação de um item de backlog ocorre durante sessões de Refinamento de Backlog
- ▶ A preparação é importante para diminuir os riscos de uma Sprint mal planejada

Definition of Ready (exemplo)

- ▶ Um item de backlog será considerado “preparado” quando:
 - Sua descrição especificar claramente o que será realizado, por quem e por quê (razão de negócio)
 - Suas principais regras de negócio tiverem sido especificadas
 - Seus critérios de aceitação forem claros e objetivos
 - O time de desenvolvimento considerá-lo factível de implementação em uma Sprint

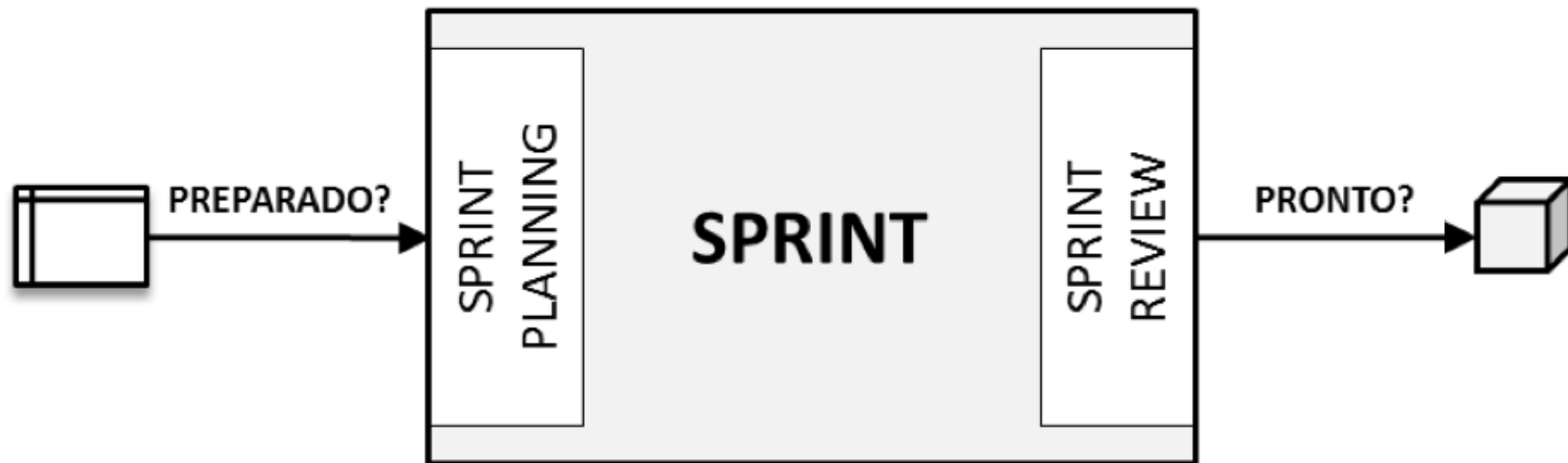
Definition of Done (“Pronto”)

- ▶ A definição de “pronto” é um acordo formal entre PO e Time de Des.
- ▶ Define quando um trabalho realizado na Sprint realmente está pronto
- ▶ São critérios acordados de antemão para garantir a transparência
- ▶ A definição é criada antes do desenvolvimento, mas pode evoluir durante o projeto

Definition of Done (exemplo)

- ▶ Um item de backlog ou incremento de produto será considerado “pronto” quando estiver:
 - Com toda a documentação técnica e de negócio atualizadas
 - Aderente a todas as métricas de qualidade definidas para o projeto
 - Atendendo todos os testes unitários e de aceitação

Ready versus Done



Fonte: Rafael Sabbagh

Exercícios [15]

(EBC – CESPE 2011)

No Scrum, o taskboard é um grande painel onde podem ser colocadas informações importantes para o acompanhamento do sprint. Por meio do taskboard, as informações acerca das atividades não iniciadas (sprint backlog), das que estão em andamento e das concluídas ficam sempre visíveis e disponíveis para todos os interessados no projeto.

Exercícios [16]

(EBC – CESPE 2011)

SCRUM é um framework para gerenciar o desenvolvimento de produtos complexos. Com relação a essa metodologia, assinale a alternativa correta.

- a) O SCRUM Master é o responsável primário pelo gerenciamento do Backlog do produto.
- b) O SCRUM Master é responsável por ordenar (priorizar) os itens do Backlog do Produto para alcançar melhor as metas e missões.
- c) O Sprint Backlog é uma lista de todas as tarefas que o Time de Desenvolvimento se compromete a fazer ao longo do desenvolvimento do produto.
- d) O Sprint Backlog é uma lista contendo todas as funcionalidades desejadas para o produto, sendo seu conteúdo definido pelo SCRUM Master e pelo contratante do projeto.
- e) Uma equipe SCRUM é composta por um Product Owner (Dono do Produto), por um SCRUM Master e pelo Time de Desenvolvimento.

Incremento do Produto

- ▶ O Incremento do Produto é o resultado de trabalho de uma Sprint
 - Todos os itens entregues devem estar “prontos”
- ▶ Espera-se que o incremento seja implantável, mas o PO pode decidir esperar antes de realizar um lançamento (release)

Meta da Sprint

- ▶ Uma Meta de Negócio no Scrum não é um valor financeiro ou uma determinada quantidade de trabalho
- ▶ Na verdade, é uma necessidade a ser satisfeito por meio de um trabalho
- ▶ Funcionam como guias ou motivadores, e são definidas pelo Product Owner juntamente com o Time de Desenvolvimento

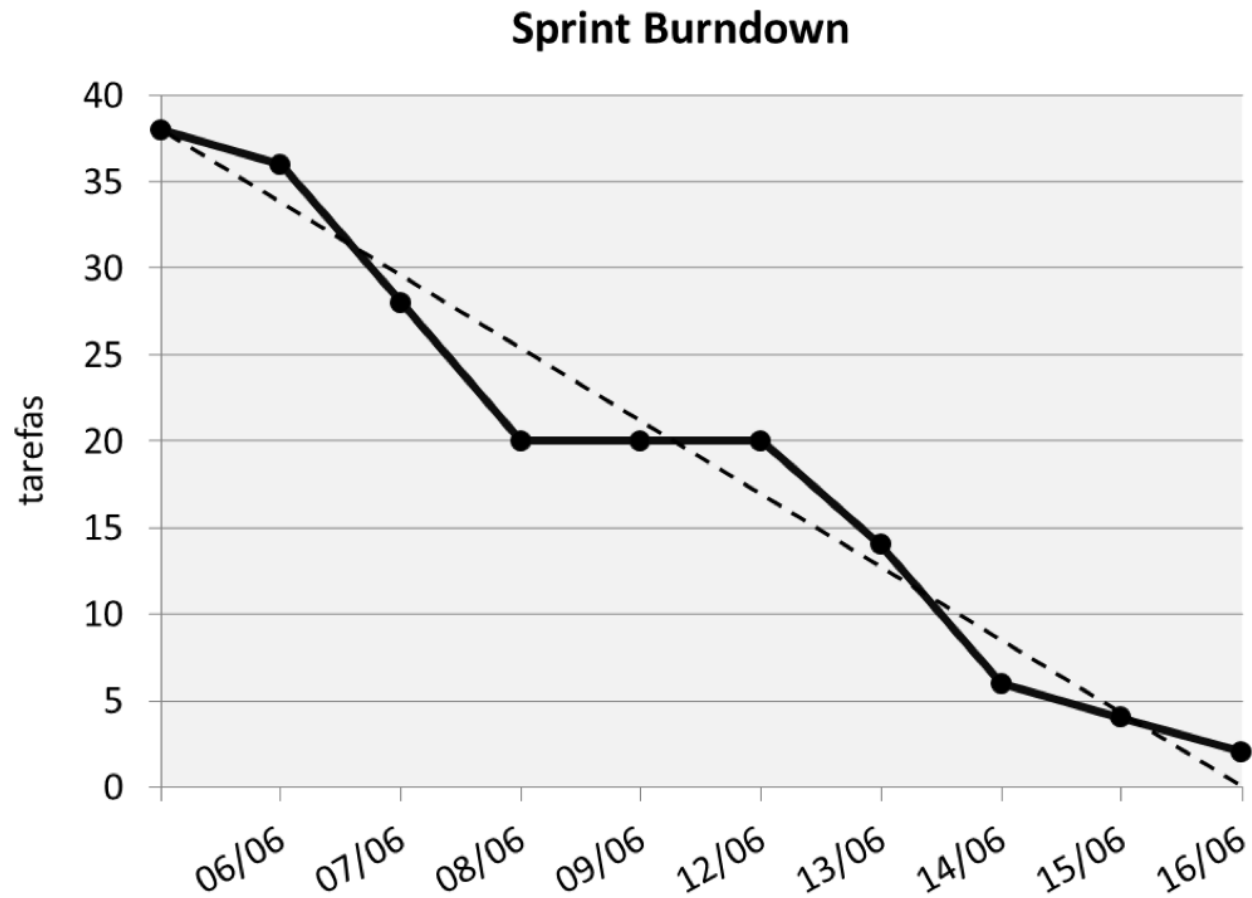
Meta da Sprint (cont.)

- ▶ Definida no Sprint Planning
- ▶ Não é numérica ou um conjunto de itens, mas um objetivo de negócio
- ▶ Não deve mudar durante a Sprint, uma vez que já foi estabelecida
- ▶ O Time de Desenvolvimento não precisa, necessariamente, implementar todos os itens de uma Sprint – ele precisa alcançar a Meta

Gráficos de Acompanhamento

- ▶ Os Gráficos de Acompanhamento são ferramentas para visualizar o progresso do trabalho
- ▶ O gráfico mais utilizado por equipes ágeis é o de “Sprint Burndown Chart” (cuidado: não é parte integrante do Scrum)

Sprint Burndown Chart



Fonte: Rafael Sabbagh

Exercícios [17]

(SERPRO – QUADRIX 2014)

No modelo ágil Scrum, durante o sprint, cabe ao product owner manter o sprint backlog atualizado, indicando as tarefas já concluídas e aquelas ainda por concluir, preferencialmente mostradas em um gráfico atualizado diariamente e à vista de todos. A cada dia podese avaliar o andamento das atividades, contando a quantidade de atividades por fazer e a quantidade de atividades terminadas, o que vai produzir o diagrama:

- a) sprint schedule
- b) sprint burndown
- c) daily work.
- d) daily burndown.
- e) dailyscrum.

Cerimônias (eventos)

Cerimônias

Scrum

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

Extra Scrum

- Release

Sprint

- ▶ Objetivo:
 - Atingir a meta da Sprint
- ▶ Quando:
 - Durante todo o projeto, de maneira contínua
- ▶ Duração:
 - Fixa, de uma a quatro semanas
- ▶ Participantes:
 - PO, SM, DevTeam
- ▶ Saída:
 - Incremento do Produto “pronto”

Sprint (características)

- ▶ É o ciclo de desenvolvimento do Scrum
- ▶ Ocorrem uma atrás da outra, sem intervalos
 - Tudo o que acontece no Scrum é trazido para dentro de uma Sprint
- ▶ Pode ser cancelada apenas pelo PO
- ▶ São como “miniprojetos”, onde ocorre tudo o que é necessário ser feito
- ▶ A duração fixa cria um ritmo regular

Exercícios [1 8]

(TRE/RS – FCC 2010)

Os princípios Scrum são usados para guiar as atividades de desenvolvimento dentro de um processo que incorpora as seguintes atividades de arcabouço: requisitos, análise, projeto, evolução e entrega. Em cada atividade de arcabouço, as tarefas de trabalho ocorrem dentro de um padrão de processo chamado

- a) pendência.
- b) iterator.
- c) demo.
- d) história de usuário.
- e) sprint.

Sprint Planning

- ▶ **Objetivo:**
 - Planejar o ciclo de desenvolvimento que se inicia
- ▶ **Quando:**
 - No primeiro dia da Sprint
- ▶ **Duração:**
 - Máxima, proporcional a 8 horas para uma Sprint de quatro semanas
- ▶ **Participantes:**
 - PO, SM, DevTeam
- ▶ **Saída:**
 - Meta da Sprint e Sprint Backlog

Sprint Planning (características)

Ocorre em duas etapas:

- ▶ Primeiro Tópico: “O que pode ser feito nesta Sprint?”
- ▶ Segundo Tópico: “Como o trabalho será feito?”

Sprint Planning (“O quê”)

- ▶ O PO apresenta os itens mais importantes e responde dúvidas sobre cada um deles
- ▶ Boa prática: apenas itens “preparados” (ready) devem ser considerados
- ▶ O PO propõe a Meta da Sprint e o Time de Desenvolvimento se compromete a alcançá-la

Sprint Planning (“Como”)

- ▶ O Time de Desenvolvimento traça um “plano” para implementar os itens de backlog
 - Itens + Plano = Sprint Backlog
- ▶ Normalmente o plano é expresso em termos de tarefas
 - Os itens são quebrados em tarefas necessárias para sua implementação
- ▶ A participação do PO na segunda etapa não é obrigatória, apesar de recomendada

Exercícios [19]

(TRT1 – FCC 2011)

No SCRUM, o processo de desenvolvimento inicia com uma reunião de planejamento na qual o Product Owner e a equipe decidem, em conjunto, o que deverá ser implementado do Product Backlog. Assim, a equipe planeja seu trabalho, definindo o Sprint Backlog, na

- a) primeira parte da Sprint Planning Meeting.
- b) segunda parte da Sprint Planning Meeting.
- c) terceira parte da Sprint Planning Meeting.
- d) Sprint.
- e) Sprint Burndown.

Daily Scrum

- ▶ Objetivo:
 - Planejar o próximo dia de desenvolvimento
- ▶ Quando:
 - Diariamente, durante a Sprint
- ▶ Duração:
 - Máxima de 15 minutos
- ▶ Participantes:
 - DevTeam
- ▶ Saída:
 - Feedback para o próximo dia de trabalho

Daily Scrum (características)

- ▶ É uma reunião curta feita diariamente pelo Time de Desenvolvimento
- ▶ Ocorrem basicamente três perguntas:
 - O que foi feito desde o último Daily Scrum?
 - O que pretendemos fazer até o próximo Daily Scrum?
 - Quais impedimentos estão em nosso caminho?
- ▶ A presença do Scrum Master NÃO é obrigatória

Daily Scrum (características)

A Daily Scrum NÃO é:

- ▶ O único momento onde são informados impedimentos ao Scrum Master
 - Isso deve acontecer assim que são identificados!
- ▶ Uma reunião de trabalho
 - A Daily Scrum não é o momento para tirar dúvidas técnicas ou de negócio
- ▶ Uma prestação de contas
 - O foco não reside em cobrar tarefas

Exercícios [20]

(TRE/CE – FCC 2012)

Um dos pontos da metodologia Scrum é o Daily Scrum, que consiste em uma reunião diária com aproximadamente 15 minutos de duração onde são tratados assuntos relacionados ao projeto. Nessa reunião são feitas 3 perguntas a cada membro do time de desenvolvimento, constando o que foi feito desde a última reunião, o que será feito até a próxima reunião e qual

- a) modelo de testes está sendo utilizado pela tarefa atual.
- b) o tempo restante para finalização da tarefa.
- c) a relação da tarefa atual com o outro membro da equipe.
- d) a tarefa que está sendo executada no momento.
- e) obstáculo impede o desenvolvedor de prosseguir com a tarefa.

Sprint Review

- ▶ Objetivo:
 - Inspecionar o incremento do produto
- ▶ Quando:
 - No último dia de cada Sprint
- ▶ Duração:
 - Máxima, de 4 horas proporcional a uma Sprint de quatro semanas
- ▶ Participantes:
 - PO, SM e DevTeam
- ▶ Saída:
 - Visibilidade sobre o incremento de produto

Sprint Review (características)

- ▶ A Sprint Review nada mais é do que uma “demo” funcional do produto
- ▶ A demonstração é informal e funcional (nada de PowerPoint)
- ▶ É importante alinhar as expectativas com todas as partes interessadas
- ▶ O PO confirma se cada item do backlog está sendo entregue “Pronto” (Done)

Sprint Retrospective

- ▶ Objetivo:
 - Melhoria contínua dos processos de trabalho
- ▶ Quando:
 - No último dia de cada Sprint, após a Sprint Review
- ▶ Duração:
 - Máxima, de 3 horas proporcional a Sprints de quatro semanas
- ▶ Participantes:
 - PO, SM e DevTeam
- ▶ Saída:
 - Propostas de melhorias nos processos de trabalho

Sprint Retrospective (características)

- ▶ Buscam melhorar o processo utilizado pela equipe
- ▶ Ao contrário da reunião *post-mortem* tradicional, a Retrospectiva permite melhoria durante o projeto
- ▶ São inspecionados: comportamentos, práticas, dinâmicas, ferramentas, etc.
- ▶ NÃO se deve:
 - Identificar melhorias no produto (Sprint Review)
 - Buscar culpados (realizar acusações)

Exercícios [21]

(ALEPE – FCC 2014)

O Scrum define reuniões e eventos que devem ser realizados de forma a oferecer oportunidades formais para inspeção e adaptação, cujos tempos de duração são referenciais máximos recomendados. Considere:

- I. É uma Sprint de um mês, para inspecionar o incremento e adaptar o Backlog do Produto, se necessário.
- II. É uma reunião timeboxed de 3 horas para uma Sprint de um mês, sendo uma oportunidade para o Time Scrum inspecionar a si próprio e criar um plano para melhorias a serem aplicadas na próxima Sprint.
- III. É um evento timeboxed de 15 minutos, para que a Equipe de Desenvolvimento possa sincronizar as atividades e criar um plano para as próximas 24 horas.
- IV. É um timebox de 8 horas para uma Sprint de um mês de duração.

Exercícios [21]

(ALEPE – FCC 2014)

Estão de acordo com as definições I, II, III e IV, respectivamente, as denominações:

- a) planejamento da Sprint revisão da Sprint daily Scrum retrospectiva da Sprint
- b) revisão da Sprint retrospectiva da Sprint daily Scrum planejamento da Sprint
- c) revisão da Sprint planejamento da Sprint 15 min break retrospectiva da Sprint
- d) retrospectiva da Sprint planejamento da Sprint short meeting revisão da Sprint
- e) planejamento da Sprint retrospectiva da Sprint daily Scrum revisão da Sprint

Release

- ▶ **Objetivo:**
 - Implantar o produto e disponibilizá-lo aos usuários finais
- ▶ **Quando:**
 - O Product Owner julgar que existe valor suficiente para o produto ser utilizado
- ▶ **Participantes:**
 - PO e DevTeam
- ▶ **Saída:**
 - Produto implantado e funcional

Release (características)

- ▶ A estratégia de lançamento (release) é de responsabilidade do PO
- ▶ A estratégia deve estabelecer quais são os objetivos a serem alcançados
- ▶ Podemos classificar a release pela sua Frequência e Público-Alvo

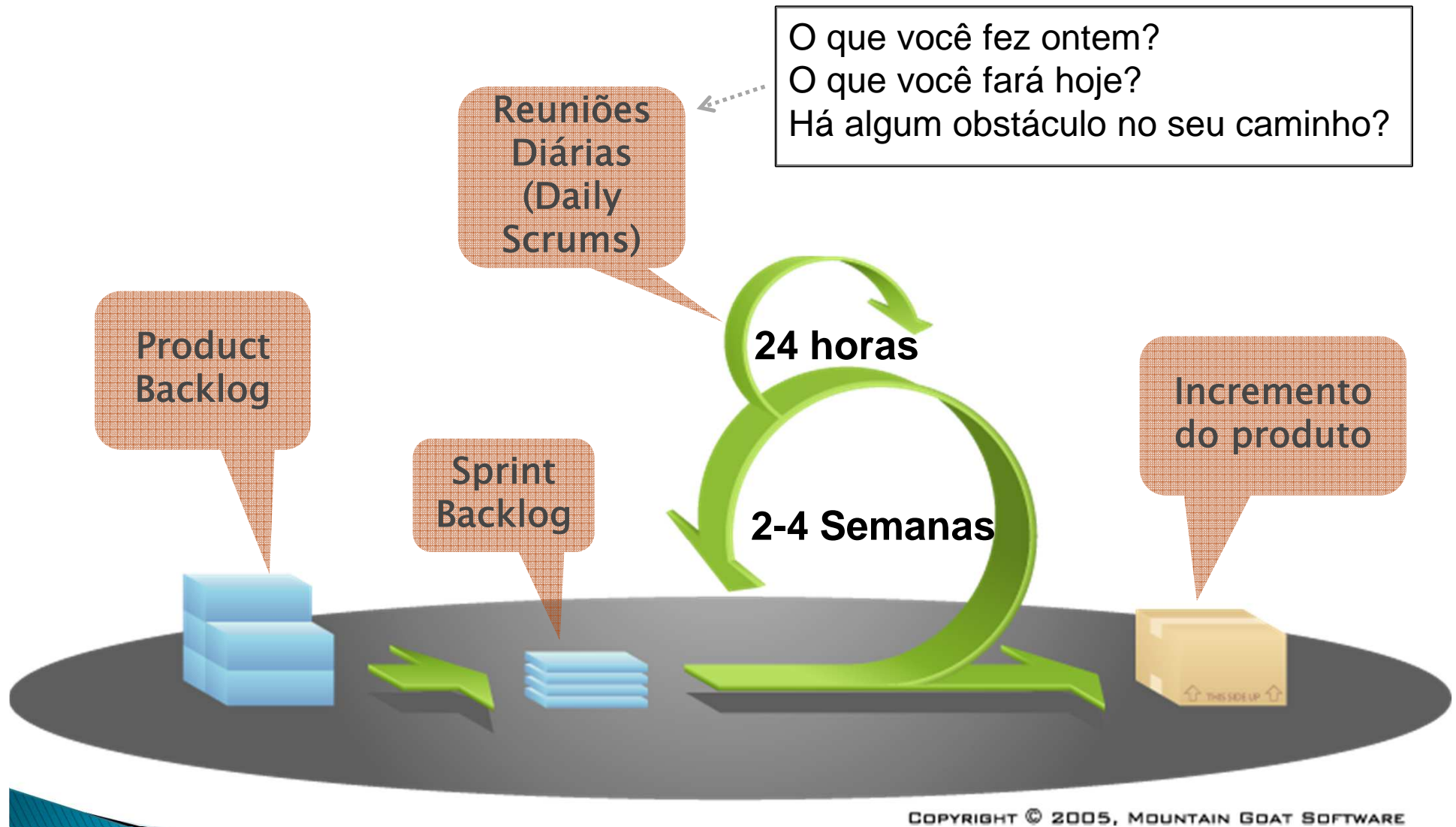
Release (frequência)

- ▶ Release por Valor
 - Quando o PO julgar que o produto já possui valor suficiente
- ▶ Release por Sprint
- ▶ Release por Item de Backlog
- ▶ Release por Meta (plano)
 - Ocorre quando é estabelecido um plano para alcançar uma meta
 - Normalmente abrange várias sprints

Release (público-alvo)

- ▶ Release para usuários finais
- ▶ Release para usuários intermediários
 - Internos, especialistas no negócio, usabilidade ou qualquer outra pessoa capacitada a gerar feedback
- ▶ Release para usuários selecionados
 - Normalmente um subgrupo de usuários finais (Beta Testers)

Visão Geral



Gabarito dos Exercícios

- ▶ [1] C, E, E
- ▶ [2] E
- ▶ [3] C, E
- ▶ [4] E
- ▶ [5] E, E, C
- ▶ [6] A
- ▶ [7] E
- ▶ [8] E
- ▶ [9] C, C
- ▶ [10] C, E, E
- ▶ [11] A
- ▶ [12] A
- ▶ [13] E
- ▶ [14] D
- ▶ [15] C
- ▶ [16] E
- ▶ [17] B
- ▶ [18] E
- ▶ [19] B
- ▶ [20] E
- ▶ [21] B

FIM

