



Um guia leve e rápido sobre a teoria e prática do Scrum
Versão 2.0

Pete Deemer

GoodAgile

www.goodagile.com

Gabrielle Benefield

Evolve

www.evolvebeyond.com

Craig Larman

www.craiglarman.com

Bas Vodde

Odd-e

www.odd-e.com



InfoQ
ENTERPRISE SOFTWARE
DEVELOPMENT SERIES

Um recado para os leitores: Existem muitas descrições concisas do Scrum disponíveis on-line e essa cartilha tem como objetivo fornecer um próximo nível de detalhes sobre estas práticas. Não pretende-se que seja a etapa final de uma educação em Scrum; as equipes que estão pensando em adotar o Scrum são aconselhadas a equipar-se com o livro *Agile Project Management with Scrum* de Ken Schwaber, ou com o Mike Cohn's *Succeeding with Agile* e aproveitar as excelentes opções de treinamento e coaching do Scrum disponíveis; Todos os detalhes estão em scrumalliance.org. Nossos agradecimentos vão para Ken Schwaber, Dr. Jeff Sutherland e Mike Cohn por suas contribuições generosas.

© 2012 Pete Deemer, Gabrielle Benefield, Craig Larman, Bas Vodde

Tradução e adaptação para o português:

Fernando N. Leal

Nota do tradutor: no universo da gestão de projetos ágeis, existem variações em diversas terminologias decorrentes da cultura regional (cidades, estados e países) bem como da cultural organizacional (setorizada ou não). Para fins de padronização e acessibilidade para todos os interessados em pesquisas nesta área, a terminologia padrão em inglês foi preservada. Exemplo: o termo Team foi preservado mediante a diversas possibilidades na substituição deste mesmo termo por *time*, *equipe*, *equipe de desenvolvimento*, *time scrum*, *development team*, *scrum team*, etc. Algumas destas terminologias também são retratadas por diferentes valores, dependendo do autor da obra, uma vez que o termo scrum team é utilizado para retratar todos inclusos na equipe Scrum mas, também, como apenas aos membros da equipe de desenvolvimento. Outra razão para preservar a terminologia em inglês é evitar mesclar duas línguas diferentes, o que traz um pouco de confusão, visto que ora utilizamos uma terminologia sem tradução (exemplo: ScrumMaster) ora utilizamos com tradução (exemplo: Time de Desenvolvimento), além dos momentos em que os termos são mesclados entre terminologia original e traduzida (Backlog do Produto). A tradução foi realizada com o mínimo de adaptações possíveis, visando manter o método de discurso do conteúdo e exemplificações dos autores.

Além do desenvolvimento tradicional

Desenvolvimento tradicional com grupos de função única, loops de feedback atrasados ou fracos, carregados de planejamento preditivo e um fluxo sequencial de análise direto para testes não é muito bem-sucedido no mundo volátil de hoje. Essa abordagem retarda o feedback, o aprendizado e o potencial retorno do investimento devido à ausência de softwares reais em funcionamento até o final do game, causando falta de transparência, falta de capacidade de melhoria, redução da flexibilidade e aumento dos riscos comerciais e técnicos. Uma alternativa: cross-functional Teams com desenvolvimento iterativo. Eles existem há décadas, mas não foi tão amplamente utilizado quanto o modelo tradicional.

O Scrum agrupa conceitos comprovados de desenvolvimento de produtos em uma estrutura simples, incluindo: equipes reais, cross-functional Teams, equipes de autogerenciamento, loops de feedback iterativo ao longo de todo o ciclo e redução do custo de mudança. Esses conceitos aumentam a agilidade e o feedback, permitem um ROI antecipado e reduzem os riscos em geral.

Visão geral

O Scrum é uma estrutura de desenvolvimento em que os cross-functional Teams desenvolvem produtos ou projetos de maneira iterativa e incremental. Esta estrutura de desenvolvimento é caracterizada em ciclos de trabalho chamados Sprints. Essas iterações não duram mais de quatro semanas (a mais comum é de duas semanas) e ocorrem uma após a outra em sequência, sem pausa. Os Sprints são timeboxed, ou seja, eles terminam em uma data específica, mesmo se o trabalho foi concluído ou não, e nunca são prorrogados. Geralmente, as equipes do Scrum escolhem um tempo de duração para o Sprint e utilizam-no para todos os seus Sprints seguintes, até que o Team melhore e possa usar um ciclo mais curto. No início de cada Sprint, um cross-functional Team (de cerca de sete pessoas) seleciona itens (requisitos do cliente) de uma lista com os itens já priorizados. O Team concorda com um objetivo coletivo do que eles acreditam que podem fornecer até o final do Sprint, algo que é tangível e que será realmente "feito". Durante o Sprint, nenhum novo item pode ser adicionado. O Scrum adota a mudança para o próximo Sprint, enquanto o Sprint atual deve se concentrar em um objetivo pequeno, claro e relativamente estável. Todos os dias o Team se reúne brevemente para inspecionar seu progresso e ajustar as próximas etapas necessárias para concluir o trabalho restante. No final do Sprint, o Team analisa o Sprint com os Stakeholders e demonstra o que foi construído. O Team obtém feedbacks que podem ser incorporados no próximo Sprint. O Scrum enfatiza o produto de trabalho no final do Sprint, que é realmente "feito"; no caso de software, isso significa um sistema que é integrado, totalmente testado, documentado pelo usuário final e potencialmente utilizável. As principais funções, artefatos e eventos estão resumidos na *Figura 1*.

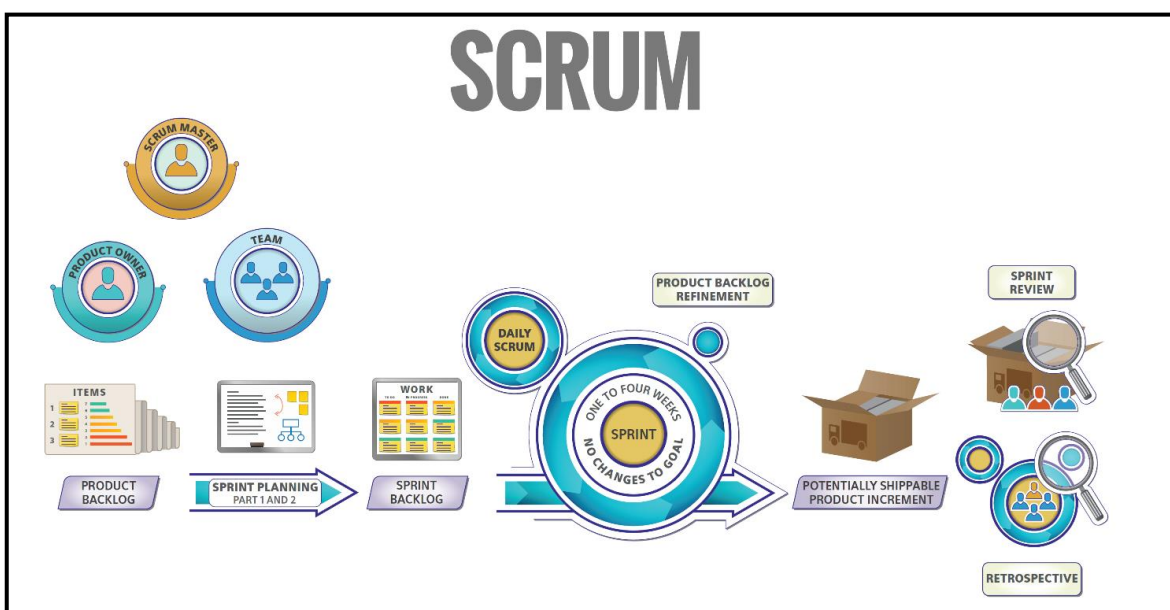


Figura 1: visão geral do Scrum.

Um tema importante no Scrum é inspecionar e adaptar. Como o desenvolvimento envolve, inevitavelmente, aprendizado, inovação e surpresas, o Scrum enfatiza um pequeno passo de desenvolvimento, inspecionando o produto resultante e a eficácia das práticas atuais, adaptando as metas do produto e práticas ao longo do processo. A chave é manter este ciclo contínuo e ininterrupto.

Scrum Roles

No Scrum, existem três funções: Product Owner, Team e ScrumMaster. Juntos, estes são conhecidos como o Scrum Team. O Product Owner é responsável por maximizar o retorno do investimento (ROI), identificando os recursos do produto, traduzindo-os em uma lista de itens priorizados conhecidos como Product Backlog, decidindo qual item deve estar no topo da lista para o próximo Sprint e continuamente priorizando e refinando esta lista. O Product Owner tem responsabilidade de ganhos e perdas para o produto em relação ao seu valor, assumindo que é um produto comercial. No caso de um desenvolvimento de um produto interno na Organização, o Product Owner não é responsável pelo ROI no sentido de um produto comercial (que irá gerar receita), mas ele ainda é responsável por maximizar o ROI no sentido de escolher, em cada Sprint, os itens com maior prioridade que agreguem valor. Na prática, 'valor' é um termo nebuloso e a priorização pode ser influenciada pelo desejo de satisfazer os clientes-chave, alinhamento com objetivos estratégicos, riscos de ataque, melhoria e outros fatores. Em alguns casos, o Product Owner e o cliente são a mesma pessoa; isso é comum em desenvolvimento de produtos internos na Organização. Em outros, os clientes podem ser milhões de pessoas com uma variedade de necessidades e, nesse caso, a função de Product Owner é semelhante à posição gerente de produto ou gerente de marketing de produto na maioria das Organizações. No entanto, o Product Owner é um pouco diferente do gerente de produto tradicional, pois ele interage ativamente e regularmente com o Team, prioriza o trabalho com todos os Stakeholders e analisa os resultados de cada Sprint, em vez de delegar decisões de desenvolvimento a um gerente de projeto. É importante notar que, no Scrum, existe uma e apenas uma pessoa que serve como, e tem a autoridade final, Product Owner e, ele ou elas é responsável pelo valor do trabalho, embora essa pessoa não precise trabalhar sozinha.

O Team (também chamada de Equipe de Desenvolvimento) cria o produto que o Product Owner indica: o aplicativo ou site, por exemplo. O Team no Scrum é “cross-funcional” pois inclui toda a expertise necessária para entregar o produto potencialmente utilizável de cada Sprint e é auto-organizável (autogerenciamento), com um grau muito alto de autonomia e responsabilidade. O Team decide quantos itens (do conjunto oferecido no Product Backlog pelo Product Owner) escolherão para compilar em um Sprint e qual o melhor caminho para atingirem a meta traçada.

Cada membro do Team é apenas um membro do Team. Observe que não há títulos de especialistas fixos em um grupo que adote o Scrum. Não existe um analista de negócios, nenhum DBA, nenhum arquiteto, nenhum líder de equipe, nenhum designer de interação / UX, nenhum programador, etc. Eles trabalham juntos durante cada Sprint de qualquer maneira que seja apropriada para atingir o objetivo que eles estabeleceram para si mesmos.

Como existem apenas membros do Team, os integrantes não são apenas cross-funcional, mas também demonstram a capacidade de aprendizado múltiplo: cada membro certamente tem pontos fortes especiais, mas também continua a aprender outras especialidades. Cada membro terá habilidades primárias, secundárias e até terciárias, e destina-se a ir aonde o trabalho está. Os indivíduos realizam tarefas em áreas menos familiares para ajudar a completar um item. Por exemplo, uma pessoa cuja habilidade principal é o design de interação poderia ter uma habilidade secundária em testes automatizados, assim como alguém com habilidade primária em redação técnica também pode ajudar na análise e programação.

O Team é composto de sete ou mais membros e, para um produto de software, o Team pode incluir pessoas com habilidades em análise, desenvolvimento, teste, design de interface, design de banco de dados, arquitetura, documentação e assim por diante. O Team desenvolve o produto e fornece ideias para o Product Owner sobre como tornar o produto ótimo. No Scrum, o Team é mais produtivo e eficaz se todos os membros forem 100% dedicados ao trabalho de um produto durante o Sprint. O Team evita a multitarefa em vários produtos ou projetos, para evitar a drenagem dispendiosa de atenções divididas e trocas de contexto. Teams estáveis estão associados a maior produtividade, portanto, evite realizar mudanças dos membros do Teams. Quando houver muitas pessoas então organiza-se vários Teams, cada um focado em diferentes recursos para o produto, com uma estreita coordenação de seus

esforços. Como o Team geralmente faz todo o trabalho, (planejamento, análise, programação e teste) para um recurso completo centrado no cliente, os Teams também são conhecidos como *Feature Teams*.

O ScrumMaster ajuda os Teams a aprenderem e aplicar o Scrum para alcançar o valor comercial. O ScrumMaster faz o que estiver ao seu alcance para ajudar o Team, o Product Owner e a organização a serem bem sucedidos. O ScrumMaster não é o gerente dos membros do Team, nem é gerente de projetos, líder do Team ou representante do Team. Em vez disso, o ScrumMaster serve o Team. Ele ajuda a remover impedimentos, protege os membros do Team de interferências externas e ajuda o Team a adotar práticas modernas de desenvolvimento. Ele educa, treina e orienta o Product Owner, o Team e o resto da organização no uso hábil do Scrum. O ScrumMaster é um treinador e professor. O ScrumMaster garante que todos (incluindo o Product Owner e os gerentes) compreendam os princípios e práticas do Scrum, e ajudam a liderar a organização através da mudança, muitas vezes difíceis e necessárias para alcançar o sucesso com o desenvolvimento ágil. Como o Scrum torna visíveis muitos impedimentos e ameaças à eficácia do Team e do Product Owner, é importante ter um ScrumMaster engajado trabalhando energeticamente para ajudar a resolver esses problemas, ou o Team ou o Product Owner acharão difícil ter sucesso. Deve haver um ScrumMaster dedicado em tempo integral, embora um Team menor possa ter um membro dentro do Team desempenhando esse papel (carregando uma carga mais leve de trabalho regular por esta razão). O bom ScrumMaster pode vir de qualquer experiência ou disciplina: Engenharia, Design, Teste, Gerenciamento de Produto, Gerenciamento de Projeto ou Gerenciamento de Qualidade.

O ScrumMaster e o Product Owner não podem ser o mesmo indivíduo, pois seu foco é tão diferente e combiná-los, muitas vezes, leva a confusão e conflito. Um resultado infeliz e comum da combinação dessas funções é um Product Owner de microgerenciamento, o oposto dos Teams de autogerenciamento que o Scrum exige. Ao contrário de um gerente tradicional, o ScrumMaster não diz às pessoas o que fazer ou atribuir tarefas, uma vez que ele deve facilitar o processo, apoiando o Team enquanto ele organiza-se e gerencia-se. Se o ScrumMaster estava anteriormente em uma posição de gerenciamento de Teams, eles precisarão mudar significativamente sua mentalidade e estilo de interação para o Team ter sucesso com o Scrum.

Nota: não há papel de gerente de projeto no Scrum. Isto é porque nenhum é necessário. As responsabilidades tradicionais de um gerente de projeto foram divididas e redistribuídas entre os três papéis do Scrum e, principalmente, para o Team e o Product Owner, e não para o ScrumMaster. Praticar Scrum com o acréscimo de um gerente de projetos indica um mal-entendido fundamental do Scrum, o que normalmente resulta em responsabilidades conflitantes, autoridade pouco clara e resultados abaixo do ideal. Às vezes, um (ex-) gerente de projetos pode assumir o papel de ScrumMaster, mas o sucesso dessa abordagem depende muito do indivíduo e o quanto ele entende a diferença fundamental entre os dois papéis, tanto no dia-a-dia quanto nas responsabilidades e na mentalidade necessária para ser bem sucedido. Uma boa maneira de entender completamente o papel do ScrumMaster e começar a desenvolver as principais habilidades necessárias para o sucesso é participar do treinamento Certified ScrumMaster da Scrum Alliance®.

Além dessas três funções, há outros stakeholders que contribuem para o sucesso do produto, incluindo gerentes, clientes e usuários finais. Alguns stakeholders, como gerentes funcionais (por exemplo, um gerente de engenharia), podem achar que seu papel, ainda que valioso, muda ao adotar o Scrum. Por exemplo:

- eles apoiam o Team respeitando as regras e o espírito do Scrum
- eles ajudam a remover os impedimentos identificados pelo Team e pelo Product Owner
- eles disponibilizam seus conhecimentos e experiências

No Scrum, esses indivíduos substituem o tempo que anteriormente passaram desempenhando o papel de “babá” (atribuindo tarefas, obtendo relatórios de status e outras formas de microgestão) com o tempo como “guru” e “servo” do Team (mentoring, coaching, ajudando a remover obstáculos, ajudando na solução de problemas, fornecendo informações criativas e orientando o desenvolvimento de habilidades dos membros do Team). Nesta mudança, os gerentes podem precisar mudar seu estilo de

gerenciamento, por exemplo, usando o questionamento socrático para ajudar o Team a descobrir a solução para um problema, em vez de simplesmente decidir uma solução e atribuí-la ao Team.

Product Backlog

Quando um grupo está planejando fazer a transição para o Scrum, antes que o primeiro Sprint possa começar, eles precisam de um Product Backlog, uma lista priorizada (ordenada 1, 2, 3, ...) de recursos a serem desenvolvidos centrados nas necessidades do cliente. O Product Backlog existe (e evolui) durante a vida útil do produto e é o roteiro do produto (Figura 2 e Figura 3). Durante todo o ciclo Scrum, o Product Backlog é a visão única e definitiva de tudo o que poderia ser feito pelo Team, sempre em ordem de prioridade. Apenas um único Product Backlog existe para um produto, isso significa que o Product Owner é obrigado a tomar decisões de priorização em todo o espectro, representando os interesses dos Stakeholders (incluindo o Team).

	Novas estimativas no Sprint ...								
Prioridade	Item	Detalhes (wiki URL)	Tamanho inicial Estimativa	1	2	3	4	5	6
1	Como comprador, quero colocar um livro em um carrinho de compras (Esboços da interface do usuário na página da wiki)	...	5						
2	Como comprador, quero remover um livro em um carrinho de compras	...	2						
3	Melhore o desempenho do processamento de transações (consulte o destino métricas de desempenho no wiki)	...	13						
4	Investigue soluções para acelerar a validação do cartão de crédito (veja as métricas de desempenho alvo no wiki)	...	20						
5	Atualize todos os servidores para o Apache 2.2.3	...	13						
6	Diagnosticar e corrigir os erros de script de processamento de pedidos (bugzilla Identificação 14823)	...	3						
7	Como comprador, quero criar e salvar uma lista de desejos	...	40						
8	Como comprador, quero adicionar ou deletar itens da minha lista de desejos.	...	20						

Figura 2: Product Backlog.

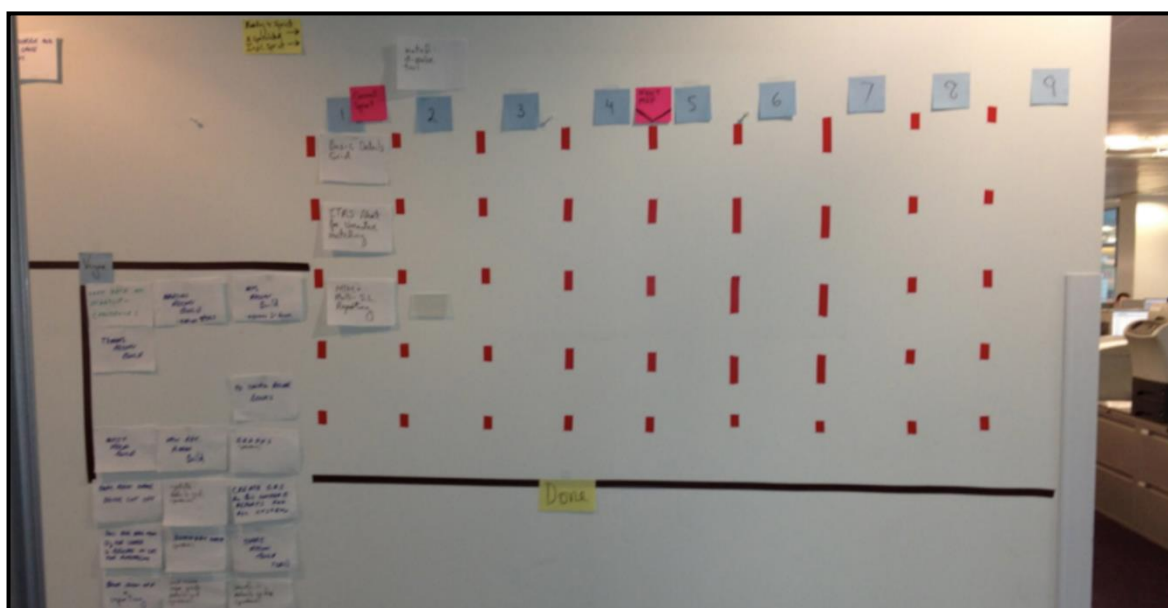


Figura 3: Gestão Visual: Items do Product Backlog no mural.

O Product Backlog inclui uma variedade de itens, principalmente os novos recursos do cliente (“permite que todos os usuários coloquem livros no carrinho de compras”), mas também grandes metas de melhoria de engenharia (por exemplo, “reescrever o sistema de C++ para Java”), metas de melhorias internas (por exemplo, “acelere nossos testes”), trabalhos de pesquisas (“investigar soluções para acelerar a validação de cartão de crédito”) e, possivelmente, defeitos conhecidos (“diagnosticar e corrigir erros de script de processamento de pedidos”) se houver apenas alguns problemas. (Um sistema com muitos defeitos geralmente possui um sistema separado de rastreamento destes defeitos).

Os itens do Product Backlog são articulados de qualquer maneira clara e sustentável. Ao contrário do mal-entendido popular, o Product Backlog não contém User Stories, mas contém itens. Esses itens podem ser expressos como User Stories, casos de uso ou qualquer outra abordagem de requisitos que o grupo julgue útil. Mas seja qual for a abordagem, a maioria dos itens deve se concentrar em fornecer valor aos clientes. Um bom Product Backlog é DEEP... ou seja: detalhado, aprofundado e consistente. O itens devem conter as seguintes características:

Detalhados apropriadamente: os itens de prioridade máxima são mais refinados e detalhados do que os itens prioritários, uma vez que o primeiro será trabalhado mais cedo do que o segundo. Por exemplo, os 10% superiores do backlog podem ser compostos de itens muito pequenos e bem analisados, já os outros 90% ainda não possuem uma análise completa e ainda precisam ser ajustados adequadamente.

Estimativa: os itens para o release atual precisam ter estimativas e, além disso, devem ser considerados para re-estimativa de cada Sprint, devido as novas informações que serão adquiridas e ao processo de autoaprendizado do Team e dos demais Stakeholders. A equipe fornece ao proprietário do produto estimativas de esforço necessário para cada item do Product Backlog e, talvez, também estimativas técnicas de risco. O Product Owner e outros Stakeholders da empresa fornecem informações sobre o valor das solicitações do produto, que podem incluir receita obtida, custos reduzidos, riscos do negócio, nível de importância para Stakeholders, dentre outras informações.

Emergente: em resposta ao aprendizado e às transformações necessárias, o Product Backlog é regularmente refinado. Em cada Sprint itens podem ser adicionados, removidos, modificados, divididos e alterados conforme a sua prioridade. Assim, o Product Backlog é continuamente atualizado pelo Product Owner para refletir as mudanças nas necessidades do cliente, novas ideias ou insights, movimentos da concorrência, obstáculos técnicos que surgem no decorrer do ciclo, etc.

Priorizado: os itens no topo do Product Backlog são priorizados e ordenados em uma ordem 1-N. Em geral, os itens de maior prioridade devem oferecer o maior retorno para seu investimento: muito retorno (valor de negócio) para baixo custo (custo geral). Outra motivação para aumentar a prioridade de um item é lidar com os altos riscos o quanto antes, antes que os riscos ataquem você.

Normalmente, o desenvolvimento tradicional não enfatiza a entrega de acordo com o mais alto custo, mas esse é um tema do Scrum e, portanto, o Product Owner precisará aprender como avaliar o impacto do valor comercial. O ScrumMaster pode ajudar o Product Owner neste processo de aprendizado. O que significa "valor de negócio"? Alguns Product Groups usam uma estimativa de ponto de valor relativo simples para cada item do Product Backlog que sintetiza uma estimativa de fatores, incluindo ganho de receita, redução de custos, preferências dos Stakeholders, diferenciação de mercado e assim por diante.

Alguns financiam um item específico por um ou mais clientes que pagam pelo seu desenvolvimento e, portanto, usam a receita exata (de curto prazo) desse item como um proxy para o valor. Para outros grupos, tal estimativa de valor específica de item é muito desfocada ou granular. Eles aplicam uma abordagem mais ampla baseada em resultados de negócios ("aumentar as assinaturas em 10% até 1º de setembro"), na qual o valor só é fornecido quando vários itens que contribuem para o resultado são entregues juntos. Nesse caso, o Product Owner precisa definir o próximo incremento do *Minimum Viable Product*.

Para estimativas de esforço, uma técnica comum é estimar em termos de tamanho relativo (fatoração em esforço, complexidade e incerteza) usando uma unidade de *Story Points* ou simplesmente *Points*. Estas são apenas sugestões. O Scrum não define a técnica para expressar ou priorizar itens no Product Backlog e não define a técnica ou unidades de estimativa. Uma técnica comum usada no Scrum é rastrear quanto trabalho o Team completa em cada Sprint, por exemplo: com uma média de 26 pontos concluídos por Sprint. Com essas informações é possível projetar uma data de lançamento para concluir todos os recursos ou quantos recursos podem ser concluídos até uma data fixa, se a média continuar e

nada mudar. Essa média é chamada de *velocity*. A *velocity* é expressa nas mesmas unidades que as estimativas de tamanho dos itens do Product Backlog.

Os itens no Product Backlog podem variar significativamente em tamanho ou esforço. Os maiores são divididos em itens menores durante o workshop de refinamento do Product Backlog ou no Sprint Planning Meeting, onde itens menores serão consolidados. Os itens do Product Backlog para os próximos Sprints seguintes devem ser pequenos e detalhados o suficiente para que sejam compreendidos pelo Team, permitindo que as previsões feitas na reunião de Sprint Planning sejam significativas: isso é chamado de *actionable size*.

Grandes melhorias de engenharia que consomem muito tempo e dinheiro devem estar no Product Backlog, uma vez que podem ser um investimento comercial opcional, em última análise, a ser feito pelo Product Owner orientado para os negócios. Note que no Scrum, o Team tem autoridade independente referente a quantos itens do Product Backlog eles decidem levar para um Sprint, então eles são independentemente livres para assumir pequenos trabalhos de melhoria de engenharia, pois eles podem ser considerados parte do custo normal de fazer negócios e o que seja necessário para que um desenvolvedor faça seu trabalho corretamente. Dito isso, em cada Sprint, a maioria do tempo de um Team geralmente deve estar focados nos objetivos do proprietário do produto, e não em tarefas internas de engenharia.

Um dos mitos sobre o Scrum é que ele impede que você escreva especificações detalhadas quando, na realidade, cabe ao Product Owner e ao Team decidir a quantidade necessária de detalhes, e isso variará de um item do Product Backlog para o seguinte, dependendo da percepção do Team e de outros fatores. Indique o que é importante na menor quantidade de espaço necessário. Em outras palavras, não descreva todos os detalhes possíveis de um item, apenas deixe claro o que é necessário para que ele seja entendido e desenvolva-o com o diálogo contínuo entre o Team, o Product Owner e os Stakeholders. Os itens do Product Backlog de baixa prioridade, que não serão trabalhados por algum tempo, são geralmente “brutos” (grandes, com requisitos menos detalhados). Os itens do Product Backlog de alta prioridade e refinados, que serão implementados em breve, tendem a ter mais detalhes.

Definition of Done (DoD)

O output de cada Sprint é oficialmente chamado de *Potentially Shippable Product Increment*. Antes de iniciar o primeiro Sprint, o Product Owner, o Team e o ScrumMaster precisam revisar o que é necessário para que um item do Product Backlog seja potencialmente utilizável. Todas as atividades necessárias para enviar o produto devem ser incluídas na definição de Potentially Shippable e, portanto, devem ser feitas durante o Sprint.

Infelizmente, quando as equipes começam a usar o Scrum, elas geralmente não conseguem atingir o objetivo de entregar um Potentially Shippable Product Increment a cada Sprint. Muitas vezes isso ocorre porque o Team não tem automação ou não é cross-funcional o suficiente (por exemplo, os desenvolvedores técnicos ainda não estão incluídos no cross-funcional Team). Ao longo do tempo, o Team precisará melhorar para poder entregar um *Potentially Shippable Product Increment* a cada Sprint mas, para começar, eles precisarão criar uma linha de base de seus recursos existentes. Isso é registrado como *Definition of Done*.

Antes do primeiro Sprint, o Product Owner e o Team precisam concordar com um Definition of Done, que é um subconjunto das atividades necessárias para criar um Potentially Shippable Product Increment. O Team planejará seu trabalho do Sprint de acordo com este Definition of Done.

Um bom Product Owner sempre desejará que o Definition of Done seja o mais próximo possível de Potentially Shippable, pois isso aumentará a transparência no desenvolvimento e diminuirá o atraso e risco. Se o Definition of Done não for igual a Potentially Shippable então o trabalho realizado não é aceito, é adiado e retornado a liberação que causa esse risco e atraso para ser reavaliado. Este trabalho atrasado é às vezes chamado de *undone work*. O Team deve melhorar continuamente, o que se reflete em estender sua Definição de Feito.

Sprint Planning

Resumo: uma reunião para preparar o Sprint, normalmente dividida em duas partes (a primeira parte é "o quê" e a segunda parte é "como").

Participantes: primeira parte: Product Owner, Team e ScrumMaster. Segunda parte: Team, ScrumMaster e Product Owner (este último é opcional, mas deve ser acessível para consultas).

Duração: cada fase é *timeboxed* a uma hora por semana de Sprint.

No início de cada Sprint, o Sprint Planning Meeting ocorre. Ele é dividido em duas sub-reuniões distintas, a primeira delas é chamada de Sprint Planning Part One. No Sprint Planning Part One, o Product Owner e o Team analisam os itens de alta prioridade do Product Backlog que o Product Owner está interessado em implementar no Sprint. Normalmente, esses itens terão sido bem analisados em um Sprint anterior (durante o refinamento do Product Backlog), de modo que nessa reunião haverá apenas pequenas questões de esclarecimento de última hora. Na reunião, o Product Owner e o Team discutem os objetivos e o contexto desses itens de alta prioridade no Product Backlog, fornecendo ao Team uma visão da interpretação do Product Owner. A Part One se concentra em entender o que o Product Owner quer e por que o que ele quer é necessários. No final da Part One o (sempre ocupado) Product Owner pode sair, embora deva estar disponível (por exemplo, por telefone) durante a Part Two da reunião. Na Part One, o Team e o Product Owner também podem elaborar o Sprint Goal. Esta é uma declaração resumida do objetivo do Sprint, que idealmente tem um tema coeso. O Sprint Goal também dá flexibilidade ao escopo do Team em relação ao que eles podem realmente fornecer, porque embora possam ter que remover algum item (já que o Sprint é *timeboxed*), eles devem comprometer-se a entregar algo tangível e “pronto” que esteja no espírito do Sprint Goal. Quão grandes devem ser os itens que são levados para um Sprint? Cada item deve ser dividido o suficiente para que seja estimado, exigindo consideravelmente menos do que todo o Sprint. Uma diretriz comum é que um item é estimado pequeno o suficiente para ser concluído em um quarto ou menos de um Sprint por todo o Team.

Sprint Planning Part Two foca em como implementar os itens que o Team decidiu assumir. O Team prevê a quantidade de itens que podem ser concluídos até o final do Sprint, começando na parte superior do Product Backlog (em outras palavras, iniciando com os itens que são a prioridade mais alta para o Product Owner) e trabalhando na lista em ordem. Esta é uma prática chave no Scrum: O Team decide quanto trabalho será concluído, em vez de ser atribuído a eles pelo Product Owner. Isso traz uma previsão mais confiável porque o Team está fazendo isso com base em sua própria análise e planejamento. Embora o Product Owner não tenha controle sobre quantos itens o Team irá selecionar para a Sprint, ambos sabem que os itens são retirados da parte superior do Product Backlog ou, em outras palavras, dentre os itens que ele classificou como mais importantes. O Team tem a capacidade de fazer lobby para itens mais abaixo na lista e isso geralmente acontece quando o Team e o Product Owner percebem que algo de menor prioridade se encaixa de maneira fácil e apropriada, nos itens de alta prioridade.

A Sprint Planning Meeting geralmente dura várias horas, mas não mais do que quatro horas para um Sprint de duas semanas. O Team está fazendo uma previsão séria para concluir o trabalho, e isso requer uma reflexão cuidadosa para ter êxito. A Part One e a Part Two têm comprimentos iguais de *timebox*. Para um Sprint de duas semanas cada parte é de duas horas, no máximo. O Scrum não define exatamente como fazer o Sprint Planning Part Two. Alguns Teams usam sua *velocity* dos Sprints anteriores para orientar o objetivo desejado. Outros Teams usarão uma abordagem mais refinada para calcular primeiro a sua *capacity*. Ao usar a abordagem de *capacity*, o Team, no Sprint Planning Part Two, calcula quanto tempo cada um dos seus membros possui para o trabalho relacionado ao Sprint. A maioria dos Teams presume que os seus membros só podem se concentrar no trabalho relacionado ao Sprint por 4-6 horas por dia, o resto do tempo será para e-mail, almoço, facebook, reuniões e tomar café. Depois que a *capacity* é determinada, o Team precisa descobrir quantos itens do Product Backlog podem ser concluídos nesse período e como eles serão concluídos. Isso geralmente começa com uma discussão de design em um quadro branco. Uma vez que o design geral seja entendido, o Team decompõe os itens do Product Backlog em um trabalho de baixa granularidade. Antes de selecionar os itens do Product Backlog, o Team pode se concentrar em gerar tarefas para uma meta de melhoria criada no Sprint Retrospective anterior. Em seguida, o Team seleciona o primeiro item do Product Backlog (o item de maior prioridade do Product Owner) e trabalha-o até que ele esteja completamente decomposto em *tasks*. Para cada item, o Team cria uma lista de trabalho que consiste em itens do Product Backlog decompostos desta mesma forma. Quando o item do Product Backlog for pequeno o suficiente para o Team desenvolvê-lo e implementá-lo em apenas algumas horas, este mesmo item do Product Backlog pode ser considerado como *task*. Esta lista de trabalho desenvolvida durante o Sprint é chamada de Sprint Backlog (Figura 4 e Figura 5).

Novas estimativas de esforço restantes ao final do dia									
Items do product Backlog	Sprint tasks	Voluntário	Estimativa inicial de esforço	1	2	3	4	5	6
Como um comprador, eu quero colocar um livro no carrinho de compras.	modificar banco de dados		5						
	criar página da web (UI)		8						
	criar página da web (lógica JavaScript)		13						
	escreva testes de aceitação automatizados		13						
	atualizar página de ajuda do comprador		3						
	...								
Melhorar o desempenho do processamento das transações	mesclar código DCP e concluir nível da camada de testes		5						
	Completar ordem de máquina para pRank		8						
	altere o DCP e o leitor para usar o pRank http API		13						

Figura 4: Exemplo possível para criar um Sprint Backlog.

No final da Sprint Planning Meeting, o Team define uma meta realista para o que eles acreditam que podem entregar até o final do Sprint. Tradicionalmente, isso era chamado de Sprint Commitment, onde o Team compromete-se a fazer o melhor que puder para alcançar seu objetivo. Infelizmente, isso às vezes era mal interpretado como uma promessa escrita em sangue, em vez do Team “ir em frente” seriamente. Para evitar essa confusão, o Sprint Goal agora é chamado de forecast e é comunicado ao Product Owner.

O Scrum incentiva trabalhadores multi-qualificados ao invés de apenas “trabalhar para o cargo adquirido”, como um “testador” apenas fazendo testes. Em outras palavras, os membros do Team “vão para onde o trabalho está” e ajudam o máximo possível. Se houver muitas tarefas de teste, todos os membros do Team podem ajudar. Isso não implica que todos sejam generalistas pois, sem dúvida, algumas pessoas são especialmente habilitadas em testes (e assim por diante), mas os membros do Team devem trabalhar juntos e aprenderem novas habilidades uns dos outros. Consequentemente, durante a geração e estimativa de tarefas no Sprint Planning, não é necessário, e nem apropriado, que as pessoas se voluntariem para todas as tarefas que “elas podem fazer melhor”. Em vez disso, é melhor se voluntariarem para uma tarefa de cada vez. É hora de escolher uma nova tarefa e considerar a escolha de tarefas que, de propósito, envolvam o aprendizado (talvez por meio de um par de trabalho com um especialista). Esta é uma das razões para não pré-atribuir tarefas durante o Sprint Planning, em vez disso, isso deve ser feito “conforme necessário” durante o Sprint.

Tudo isso dito, há raros momentos em que, por exemplo, um membro chamado John possa fazer uma tarefa em particular, porque levaria muito tempo ou seria impossível para os outros aprenderem. Talvez John seja a única pessoa com qualquer habilidade artística para desenhar imagens. Outros membros do Team não poderiam desenhar um “homem de palitinho” se sua vida dependesse disso. Neste caso raro, e se não for raro e não ficar mais raro conforme o Team aprende, há algo errado. Pode ser necessário perguntar se o total de tarefas de desenho planejadas que devem ser feitas por John é viável dentro do curto Sprint.

Muitos Teams têm um Sprint Backlog na forma de um painel de tarefas do tamanho de uma parede (geralmente chamado de Scrum Board) onde as tarefas (escritas em Post-It Notes) migram durante o Sprint através de colunas To Do, Work in Progress e Done. Veja a Figura 5.



Figura 5: Gerenciamento Visual – Sprint Backlog Tasks no mural

Um dos pilares do Scrum: uma vez que o Team define seu Sprint Goal, quaisquer adições ou alterações devem ser adiadas até o próximo Sprint. Isso significa que se no meio do Sprint o Product Owner decidir que há um novo item no qual ele gostaria que o Team trabalhasse, ele não pode fazer a alteração até o início do próximo Sprint. Se alguma circunstância externa significar uma alteração nas prioridades do Sprint Goal de tal maneira que o Team perderia o seu tempo se continuasse a rodar o Sprint, o Product Owner ou o Team podem encerrar o Sprint. O Team encerra o seu trabalho e um novo Sprint Planning Meeting inicia, para organizar um novo Sprint. A interrupção de fazer isso geralmente é ótima pois isso serve como um desincentivo para o Product Owner ou Team recorrer a essa decisão dramática futuramente.

Há uma poderosa influência positiva que vem do Team sendo protegido de mudanças de metas durante o Sprint. Primeiro, o Team começa a trabalhar sabendo com absoluta certeza que sua meta não mudará, o que reforça o foco do Team em garantir a conclusão. Segundo, ele disciplina o Product Owner para realmente pensar nos itens que ele prioriza no Product Backlog e o oferece ao Team para o Sprint. Seguindo estas regras Scrum, o Product Owner ganha duas coisas. Primeiro, ele ou ela tem a confiança de saber que o Team se comprometeu a fazer o melhor para completar um conjunto de trabalho realista e claro que escolheu. Com o tempo, um Team pode se tornar bastante habilidoso nas escolhas e entregas de uma previsão realista. Em segundo lugar, o Product Owner pode fazer as alterações que deseja no Product Backlog antes do início do próximo Sprint. Nesse ponto, adições, exclusões, modificações e revalorizações são possíveis e aceitáveis. Embora o Product Owner não seja capaz de fazer alterações nos itens selecionados em desenvolvimento durante o Sprint atual, ele tem apenas uma duração de Sprint ou menos para fazer as alterações desejadas. Foi-se o estigma em torno da mudança como mudança de direção, mudança de requisitos ou simplesmente mudar de ideia. Pode ser por esse motivo que os Product Owners são tão entusiasmados com o Scrum quanto qualquer um.

Daily Scrum

Resumo: atualização e coordenação entre os membros do Team.

Participantes: o Team é necessária e o Product Owner é opcional. O ScrumMaster geralmente está presente, mas garante, de qualquer maneira, que a Daily Scrum ocorra.

Duração: duração máxima de 15 minutos.

Uma vez que o Sprint tenha começado, o Team se envolve em outra das principais práticas do Scrum: o Daily Scrum. Esta é uma reunião curta (15 minutos ou menos) que ocorre em todos os dias de trabalho em um horário determinado. Todos do Team participam. Para ser breve, é recomendável que todos permaneçam de pé. É a oportunidade do Team sincronizar seu trabalho e relatar um ao outro sobre os obstáculos. No Daily Scrum, um por um, cada membro do Team relata três coisas aos seus outros membros: (1) O que foi realizado desde a última reunião ?; (2) O que será feito antes da próxima reunião ? (3) Quais obstáculos estão no caminho? Observe que o Daily Scrum não é uma reunião de status para relatar a um gerente, mas é um momento para o Team auto-organizado compartilhar uns com os outros o que está acontecendo, para ajudá-los a coordenar. Alguém toma nota dos impedimentos e o ScrumMaster é responsável por ajudar os membros da equipe a resolvê-los. Há pouca ou nenhuma discussão aprofundada durante o Daily Scrum, o tema é foado no relato das respostas para as três perguntas básicas. Se a discussão é necessária, ela ocorre imediatamente após a Daily Scrum em uma ou mais reuniões de acompanhamento paralelas, embora no Scrum ninguém precise comparecer a elas. Uma reunião de acompanhamento é um evento comum em que alguns ou todos os membros do Team se adaptam às informações que ouviram no Daily Scrum. Em outras palavras: outro ciclo de inspeção e adaptação. Para novos Teams praticantes do Scrum, geralmente não é recomendado ter gerentes ou outros em cargos de autoridade percebidos participando do Daily Scrum. Isso pode fazer com que o Team se sinta “monitorado”, sob pressão para relatar um grande progresso todos os dias (uma expectativa irreal) e inibido para relatar possíveis problemas. Isso tende a minar o autogerenciamento do Team e convidá-los à microgestão. Seria mais útil para um stakeholder, em vez disso, entrar em contato com o Team após a reunião e oferecer ajuda para os impedimentos que estão retardando o progresso do Team.

Acompanhando o progresso durante o Sprint

O Team no Scrum é autogerenciável e, para fazer isso com sucesso, deve saber como tudo está indo. Todos os dias, os membros do Team atualizam sua estimativa de esforço restante para concluir seu trabalho atual do Sprint Backlog (Figura 6). Também é comum alguém adicionar o esforço restante para o Team como um todo e traçá-lo no Sprint Burndown Chart (Figura 7 e Figura 8). Este gráfico mostra, a cada dia, uma nova estimativa de quanto trabalho restante o Team ainda possui para ser concluído. Idealmente, este é um gráfico inclinado para baixo que está em uma trajetória para alcançar o "esforço zero restante" no último dia do Sprint. Por isso, é chamado de Burndown Chart. Enquanto às vezes o progresso parece bom, muitas vezes ele não está. Essa é a realidade do desenvolvimento de produtos. O importante é que isso mostra ao Team o seu progresso em direção ao seu objetivo, não em termos de quanto tempo foi gasto no passado (um fato irrelevante em termos de progresso), mas em termos de quanto trabalho ainda resta no futuro e separa o Team do seu objetivo. Se a linha de Burndown não estiver em ritmo adequado nas proximidades do final do Sprint, então o Team precisa se ajustar, de modo a reduzir o escopo do trabalho ou encontrar uma maneira de trabalhar com mais eficiência, mantendo um ritmo sustentável. Enquanto o Sprint Burndown Chart pode ser criado e exibido usando uma planilha, muitos Teams acham que é mais eficaz mostrá-lo em um papel na parede, um cartaz, na sua área de trabalho, com atualizações com caneta. Essa solução de “baixa tecnologia / alto contato” é rápida, simples e geralmente mais visível do que um gráfico de computador.

Novas estimativas de esforço Remin g no final de Da y ...										
Backlog produto Item	Tarefa de sprint	Voluntário	Inicial Estimativa de Esforço	1	2	3	4	5	6	
colocar um livro em um carrinho de compras	modificar banco de dados	Sanjay	5	4	3	0	0	0		
	criar página da web (UI)	Jing	3	3	3	2	0	0		
	escreva testes de aceitação automatizados	Sarah	5	5	5	5	5	0		
	atualizar página de ajuda do comprador	Sanjay e Jing	3	3	3	3	3	0		
Melhorar transação em processamento desempenho	mesclar código DCP e concluir testes em nível de camada		5	5	5	5	5	5		
	ordem de máquina completa para pRank		3	3	8	8	8	8		
Total			50	49	48	44	43	34		

Figura 6: Atualizações diárias do trabalho restante no Sprint Baklog

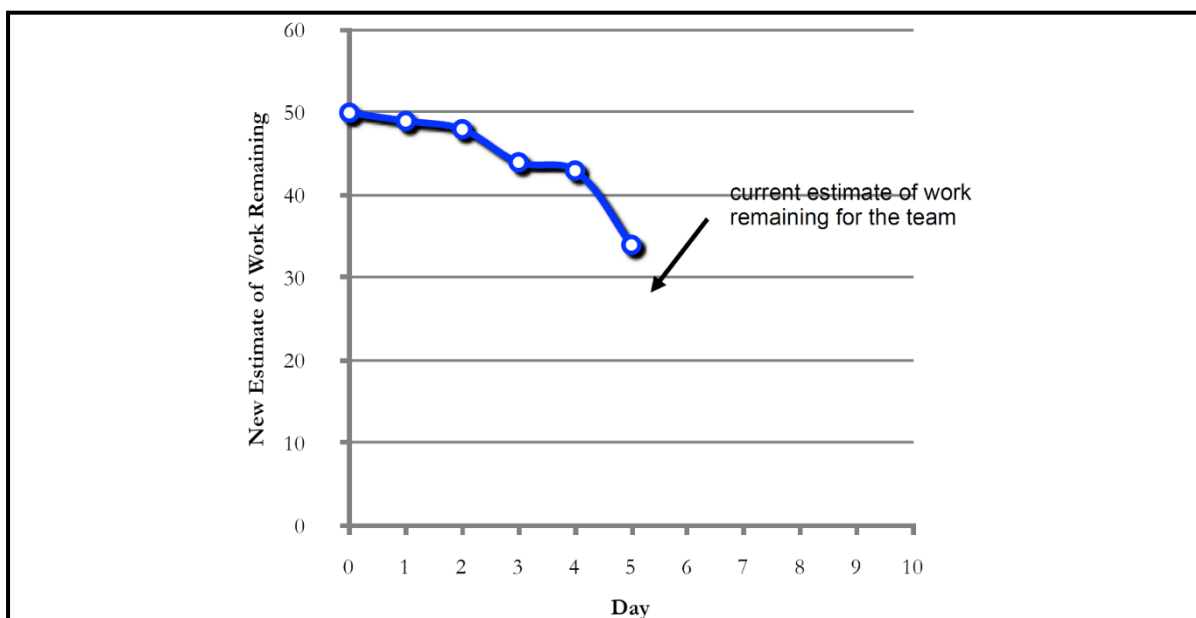


Figura 7: Sprint Burndown Chart

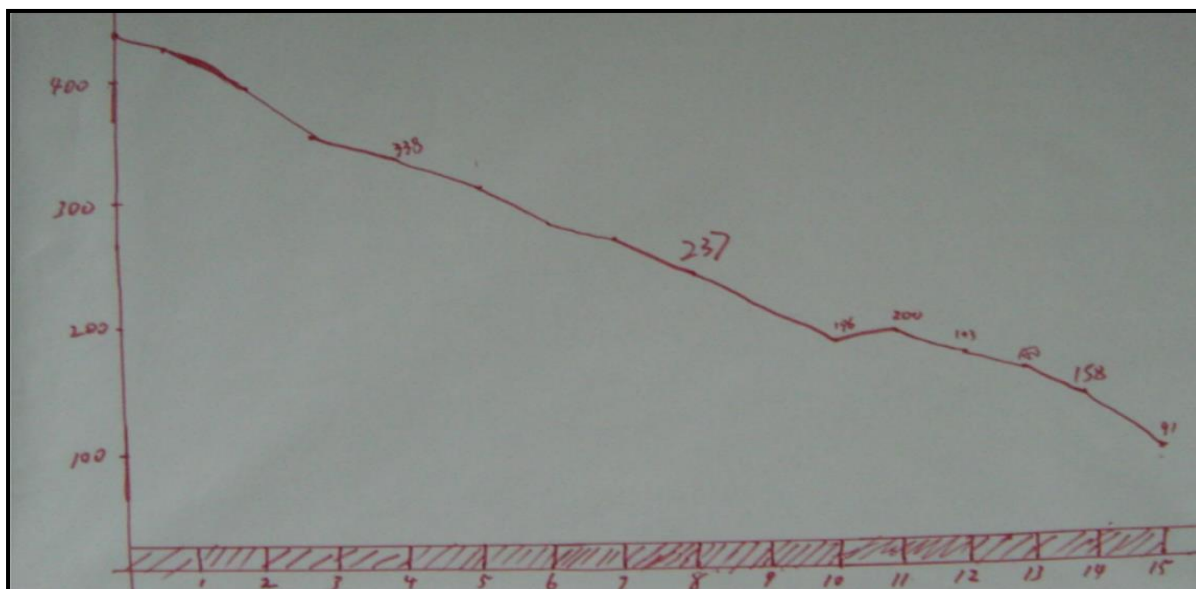


Figura 8: Gerenciamento Visual – Sprint Burndown Chart desenhado manualmente

Refinamento do Product Backlog

Resumo: divide itens grandes, analisa itens, reestima e prioriza para Sprints futuros.

Participantes: Team e o Product Owner que, por sua vez, participará de toda a atividade se for o especialista que pode ajudar com o refinamento detalhado, caso contrário, poderá atender apenas a um subconjunto para definir a direção ou redefinir as prioridades. Outros que compreendem os requisitos e podem ajudar o Team também podem participar. O ScrumMaster irá participar durante as sessões iniciais para treinar o grupo em serem eficazes, caso contrário, não poderá comparecer.

Duração: normalmente, não mais que 10% da capacity do Team para o Sprint, embora possa ser mais longa para itens de “análise pesada”. Por exemplo, em um Sprint de duas semanas, talvez um dia seja gasto em refinamento.

Uma das diretrizes menos conhecidas, mas valiosa no Scrum, é que uma porcentagem de cada Sprint deve ser dedicada por todo o Team para refinar (ou “preparar”) o Product Backlog para suportar futuros Sprints. Isso inclui análise detalhada de requisitos, divisão de itens grandes em itens menores, estimativa de novos itens e reestimativa de itens existentes. O Scrum é omissivo sobre como este trabalho é feito, mas uma técnica frequentemente utilizada é um workshop focado próximo do meio ou fim do Sprint, para que o Team e o Product Owner, assim como os Stakeholders, possam se dedicar a esse trabalho sem interrupções. Esta atividade de refinamento não é para itens selecionados para o Sprint atual, mas sim para futuros Sprints, provavelmente para um ou dois Sprints futuros. Com essa prática, o Sprint Planning se torna relativamente simples porque o Product Owner e a Team do Scrum iniciam o planning com um conjunto de itens claros, bem analisados e cuidadosamente estimados. Um sinal de que este workshop de refinamento não está sendo feito (ou que não está sendo bem feito) é que o Sprint Planning envolve questões significativas, descobertas ou confusão e parece incompleto. Planejar o trabalho, em seguida, muitas vezes transborda para o próprio Sprint, o que normalmente não é desejável.

Sprint Review

Resumo: inspeção e adaptação relacionados ao product increment.

Participantes: Team, Product Owner, ScrumMaster e Stakeholders. Este último conforme o apropriado, convidadas pelo Product Owner.

Duração: Timeboxed a uma hora por semana do Sprint.

Após o término do Sprint há o Sprint Review, onde as pessoas revisam o Sprint. Os presentes nesta reunião são o Product Owner, os membros do Team e o ScrumMaster, além de clientes, usuários, Stakeholders, especialistas, executivos e qualquer outra pessoa que esteja interessada. Para um Sprint de duas semanas utiliza-se um período máximo de duas horas. Qualquer um presente é livre para fazer perguntas e colaborar com inputs. A revisão é frequentemente rotulada erroneamente como “demo”, mas isso não captura a verdadeira intenção desta reunião. Uma ideia chave no Scrum é inspecionar e adaptar. Observar e aprender o que está acontecendo e depois evoluir com base no feedback, repetindo os ciclos. O Sprint Review é uma atividade de inspeção e adaptação do produto. É hora do Product Owner aprender o que está acontecendo com o produto e com o Team (ou seja, uma revisão do Sprint), bem como o Team deverá compreender o que está acontecendo entre o Product Owner e o mercado. Consequentemente, um elemento crítico do Sprint Review é uma conversa aprofundada entre o Team e o Product Owner para saber sobre a situação atual, obter orientações e assim por diante. A revisão definitivamente inclui o uso do software real ao vivo que o Team construiu durante o Sprint, mas se o foco do Sprint Review for apenas olhar para o produto em vez de ter uma conversa sobre ele, haverá um desequilíbrio.

A parte “live software” da Sprint Review não é uma “apresentação” que o Team dará: não há slideware. O objetivo é ser uma inspeção prática do software real em execução, por exemplo, em um ambiente de desenvolvimento sandbox. Haverá um ou mais computadores na sala de revisão em que as pessoas podem inspecionar e usar o software ao vivo. Prefira uma sessão ativa na qual os usuários reais e o Product Owner façam interação prática com o software, em vez de uma demonstração de sessão passiva do Team. Não gaste mais de 30 minutos preparando-se para o Sprint Review, caso contrário, isso sugere que algo está errado.

Sprint Retrospective

Resumo: inspeção e adaptação relacionados ao processo e ambiente do Scrum.

Participantes: Team, ScrumMaster, Product Owner (opcional). Outros Stakeholders somente podem participar se forem convidados pelo Team.

Duração: Timeboxed de 45 minutos por semana de Sprint.

A Sprint Review envolve inspecionar e adaptar em relação ao produto. A retrospectiva do Sprint, que acompanha o Sprint Review, envolve inspecionar e adaptar em relação ao processo e ao ambiente do Scrum. É uma oportunidade para o Team discutir o que está funcionando e o que não está funcionando e concordar com as mudanças a serem feitas. Às vezes, o ScrumMaster pode atuar como um facilitador efetivo para o Sprint Retrospective, mas pode ser melhor encontrar um outsider neutro para facilitar a reunião. Uma boa abordagem é que o ScrumMaster facilite as retrospectivas de cada um, o que permite a polinização cruzada entre os Teams. Existem muitas técnicas para conduzir uma Retrospectiva da Sprint, e o livro Agile Retrospectives (Derby, Larsen 2006) fornece um catálogo útil de técnicas.

Muitos Teams realizam Sprints Retrospectives apenas com foco em problemas, e isso é muito ruim. Isso pode levar as pessoas a pensarem nestes momentos como eventos deprimentes ou negativos. Em vez disso, certifique-se de que cada Sprint Retrospective também se concentre em pontos positivos ou pontos fortes. Existem vários livros sobre investigação apreciativa que oferecem dicas mais detalhadas. Sprints Retrospectives que sempre usam a mesma técnica de análise podem se tornar entediantes, portanto introduza várias técnicas ao longo do tempo.

Começando o próximo Sprint

Após a Revisão do Sprint, o Product Owner pode atualizar o Product Backlog com qualquer novo insight: adicionando novos itens, removendo os obsoletos ou revisando os existentes. O Product Owner é responsável por garantir que essas alterações sejam refletidas no Product Backlog. Veja na Figura 9 um exemplo do Product Backlog atualizado.

Novas estimativas do Sprint								
Prioridade	Item	Detalhes (wiki URL)	Inicial Estimativa	1	2	3	4	5
1	Como comprador, quero colocar um livro em um carrinho de compras (Esboços da interface do usuário na página da wiki)	...	5	0	0	0		
2	Como comprador, quero remover um livro em um carrinho de compras	...	2	0	0	0		
3	Melhore o desempenho do processamento de transações (consulte o destino 'métricas de desempenho' no wiki)	...	13	13	0	0		
4	Investigue soluções para acelerar a validação do cartão de crédito (veja as métricas de desempenho alvo no wiki)	...	20	20	20	0		
5	Atualize todos os servidores para o Apache 2.2.3	...	13	13	13	13		
6	Diagnosticar e corrigir os erros do script de processamento de pedidos (bugzilla ID 14823)	...	3	3	3	3		
7	Como comprador, quero criar e salvar uma lista de desejos	...	40	40	40	40		
8	Como comprador, quero adicionar ou excluir itens no meu desejo (lista)	...	20	20	20	20		
...				537	580	570	500	

Não há tempo de inatividade entre os Sprints. Os Teams normalmente saem de um Sprint Retrospective uma tarde para o próximo Sprint Planning na manhã seguinte (ou após o final de semana). Um dos princípios do desenvolvimento ágil é o “ritmo sustentável” e somente trabalhando em horários regulares em um nível razoável, para que assim os Teams possam continuar este ciclo indefinidamente. A produtividade cresce com o tempo, através da evolução das práticas do Team e da remoção de

impedimentos à produtividade do Team, e não por excesso de trabalho ou comprometimento da qualidade.

Os Sprints continuam até que o Product Owner decida que o produto está pronto para ser lançado. A visão de perfeição do Scrum é que o produto é potencialmente utilizável no final de cada Sprint, o que implica que não há trabalho de finalização necessário, como teste ou documentação. A implicação é que tudo está completamente terminado a cada Sprint, que você poderia enviá-lo ou implementá-lo imediatamente após o Sprint Review. No entanto, muitas organizações têm práticas de desenvolvimento, ferramentas e infra-estrutura frágeis e não conseguem atingir essa visão de perfeição e, portanto, haverá a necessidade de um “Release Sprint” para lidar com esse trabalho restante. Quando um Release Sprint é necessário, é considerado um mal necessário, onde o trabalho da organização é melhorar suas práticas para que isso não seja mais necessário.

Gerenciando Lançamentos

Uma pergunta que às vezes é feita é como, em um modelo iterativo, o Release Planning a longo prazo pode ser feito. Há dois casos a serem considerados: (1) um novo produto em sua primeira versão e (2) um produto existente em uma versão posterior. No caso de um novo produto ou de um produto existente que apenas adote o Scrum, há a necessidade de fazer o refinamento inicial do Product Backlog antes do primeiro Sprint, onde o Product Owner e o Team formam um Scrum Product Backlog apropriado. Isso pode levar alguns dias ou uma semana e envolve um workshop (às vezes chamado de Criação de Initial Product Backlog ou Release Planning), algumas análises detalhadas de requisitos e a estimativa de todos os itens identificados para o primeiro release.

Surpreendentemente, no Scrum, no caso de um produto estabelecido com um Product Backlog estabelecido, não deve haver a necessidade de nenhum planejamento de lançamento especial ou extensivo para o próximo lançamento. Por quê? Porque o Product Owner e o Team devem fazer o refinamento do Product Backlog em cada Sprint (cinco ou dez por cento de cada Sprint), preparando-se continuamente para o futuro. Esse modo contínuo de desenvolvimento de produtos elimina a necessidade de fases dramáticas pontuadas de preparação-execução-conclusão que se observam no desenvolvimento tradicional do ciclo de vida sequencial.

Durante um workshop inicial de refinamento do Product Backlog e durante o refinamento contínuo do Backlog de cada Sprint, o Team e o Product Owner farão o Release Planning, refinando as estimativas, as prioridades e o conteúdo conforme eles aprendem. Alguns lançamentos são orientados por datas, por exemplo: “vamos lançar a versão 2.0 do nosso projeto em uma feira de negócios em 10 de Novembro.” Nesta situação, o Team concluirá o maior número possível de Sprints (e criará tantos recursos) quanto possível no tempo disponível. Outros produtos exigem que determinados recursos sejam criados antes que possam ser chamados de concluídos e que o produto não seja lançado até que esses requisitos sejam atendidos, independentemente do tempo que isso leve. Como o Scrum enfatiza a produção de código potencialmente utilizável para cada Sprint, o Product Owner pode optar por começar a fazer liberações intermediárias, para permitir que o cliente obtenha os benefícios do trabalho concluído mais cedo. Como eles não podem saber tudo de antemão, o foco está em criar e refinar um plano para dar uma ampla orientação ao release e esclarecer como as decisões de tradeoff serão tomadas (escopo versus cronograma, por exemplo). Pense nisso como o roteiro que o orienta em direção ao seu destino final: quais as estradas exatas que você toma e as decisões que você toma durante a viagem podem ser determinadas no caminho. O destino é mais importante que a jornada.

A maioria dos Product Owners escolhem uma abordagem de lançamento. Por exemplo, eles decidirão uma data de lançamento e trabalharão com o Team para estimar os itens do Product Backlog que podem ser concluídos até essa data. Os itens que estão previstos para a versão atual às vezes são chamados de Release Items. Em situações onde um compromisso de “preço fixo / data fixa / entrega fixa” é necessário (por exemplo, desenvolvimento de contratos) um ou mais desses parâmetros devem ter um buffer embutido para permitir a incerteza e a mudança. A este respeito, o Scrum não é diferente de outras abordagens.

Aplicação ou foco no produto

Para aplicações ou produtos, sejam para o mercado ou para uso interno dentro de uma organização, o Scrum leva os grupos do antigo modelo de projeto para um modelo contínuo de desenvolvimento de

aplicativos / produtos . Não há mais um projeto com início, meio e fim e, portanto, nenhum gerente de projetos tradicional. Em vez disso, há simplesmente um Product Owner estável e um Team de autogerenciamento de longa duração que colaboram em uma série “infinita” de Sprints de comprimento fixo, até que o produto ou o aplicativo seja retirado. Todo o trabalho de gerenciamento de “projeto” necessário é tratado pelo Team e pelo Product Owner, que é um cliente comercial interno ou da Gerência de Produto. Não é gerenciado por um gerente de TI ou por alguém de um escritório de gerenciamento de projetos.

O Scrum também pode ser usado para projetos verdadeiros que são iniciativas únicas (em vez de trabalhar para criar ou desenvolver aplicativos de longa duração); ainda assim, neste caso, o Team e o Product Owner fazem o gerenciamento do projeto. E se houver uma quantidade de novos trabalhos insuficiente de um ou mais aplicativos existentes para garantir um Team dedicado, de longa duração, para cada aplicativo? Nesse caso, um Team estável e de longa duração pode receber itens de um aplicativo em um Sprint e, em seguida, itens de outro no próximo Sprint. Nesta situação, os Sprints são frequentemente curtos, como uma semana. Ocasionalmente, há um novo trabalho insuficiente até mesmo para a solução anterior, e o Team pode aceitar itens de várias aplicações durante o mesmo Sprint. No entanto, tenha cuidado com essa solução, pois ela pode se transformar em multitarefa improdutiva em vários aplicativos. Um tema básico de produtividade no Scrum é o foco do Team em um produto ou aplicativo para um único Sprint.

Desafios Comuns

O Scrum não é apenas um conjunto concreto de práticas e, mais importante, é uma estrutura que fornece transparência e um mecanismo que permite “inspecionar e adaptar”. O Scrum trabalha tornando visíveis as disfunções e impedimentos que estão impactando o Product Owner e a eficácia do Team, para que possam ser abordados. Por exemplo, o Product Owner pode não conhecer realmente o mercado, os recursos ou como estimar seu valor comercial relativo, ou o Team pode ser inábil na estimativa de esforço ou no trabalho de desenvolvimento.

O framework Scrum irá revelar rapidamente essas fraquezas. O Scrum não resolve os problemas do desenvolvimento, mas torna-os dolorosamente visíveis e fornece uma estrutura para que as pessoas explorem maneiras de resolver problemas em ciclos curtos e com pequenas experiências de melhoria. Suponha que o Team não consiga entregar o que eles prevêem no primeiro Sprint devido à falta de análise de tarefas e habilidade de estimativa. Para o Team, isso parece um fracasso. Mas, na realidade, essa experiência é o primeiro passo necessário para se tornar mais realista e consciente sobre suas previsões. Esse padrão do Scrum, ajudando a tornar visível a disfunção, permitindo que o Team faça algo a respeito, é o mecanismo básico que produz os benefícios mais significativos para os Teams, utilizando a experiência gerada pelo Scrum.

Um erro comum cometido, quando apresentado a uma prática de Scrum que é desafiadora, é mudar o Scrum. Por exemplo, os Teams com problemas de entrega podem decidir tornar a duração da Sprint extensível, de modo que ela nunca fique sem tempo e, no processo, assegure que ela nunca precise aprender como fazer um trabalho melhor de estimar e gerenciar seu tempo. Desta forma, sem o coaching e o suporte de um ScrumMaster experiente, as organizações podem transformar o Scrum em apenas uma imagem espelhada de suas próprias fraquezas e disfunções, minando o benefício real que o Scrum oferece: tornar visíveis os pontos bons e os ruins, dando a organização a escolha de elevar-se a um nível superior. Outro erro comum é assumir que uma prática é desencorajada ou proibida apenas porque o Scrum não exige especificamente isso. Por exemplo, o Scrum não exige que o Product Owner defina uma estratégia de longo prazo para seu produto, nem exige que os engenheiros busquem conselhos de engenheiros mais experientes sobre problemas técnicos complexos. O Scrum deixa os indivíduos envolvidos para tomar a decisão certa e, na maioria dos casos, essas duas práticas (junto com muitas outras) são bem aconselhadas. Outra coisa para ser cauteloso sobre os gerentes impondo o Scrum em seus Teams. O Scrum é sobre dar espaço e ferramentas para gerenciar a si mesmo, e ter isso ditado de cima não é uma receita para o sucesso. Uma abordagem melhor pode começar com um Team aprendendo sobre Scrum de um colega ou gerente, sendo educado de forma abrangente na formação profissional e, em seguida, tomando uma decisão como um Team para seguir fielmente as práticas por um período definido. No final desse período, o Team avaliará sua experiência e decidirá se deve continuar. A boa notícia é que, embora o primeiro Sprint seja geralmente muito desafiador para o Team, os benefícios do Scrum tendem a ser visíveis até o final, levando muitos novos Teams do Scrum a

exclamar: “Scrum é difícil, mas com certeza é um todo muito melhor do que o que estávamos fazendo antes!”

Apêndice A: Leitura Adicional

Há muito material publicado sobre o Scrum. Nesta seção de referência, gostaríamos de destacar alguns materiais on-line adicionais e alguns livros.

Materiais on-line:

- The Lean Primer - An introduction to Lean Thinking, an important influence to Scrum.
<http://www.leanprimer.com>
- The Distributed Scrum Primer - Additional tips for teams who aren't co-located.
<http://www.goodagile.com/distributedscrumprimer/>
- The ScrumMaster Checklist - A list of question that good ScrumMasters use.
<http://www.scrummasterchecklist.org/>
- Feature Team Primer - Scaling Scrum with Feature Teams,
<http://www.featureteams.org>
- The Agile Atlas - Core Scrum. ScrumAlliance description of Scrum.
<http://agileatlas.org/atlas/scrum>
- Scrum Guide - Scrum.org description of Scrum.
<http://www.scrum.org/Scrum-Guides>
- Agile Contracts Primer - How to make Scrum-friendly contracts.
<http://www.agilecontracts.org/>

Livros:

- Leading Teams - Richard Hackman
- Scaling Lean & Agile Development: Thinking and Organizational Tools for Large-Scale Scrum - Craig Larman, Bas Vodde
- Practices for Scaling Lean & Agile Development: Large, Multisite, and Offshore Product Development with Large-Scale Scrum - Craig Larman, Bas Vodde
- Agile Project Management with Scrum - Ken Schwaber
- Succeeding with Agile: Software Development using Scrum - Mike Cohn

Apêndice B: Terminologia

Burn Down

A tendência de trabalho restante ao longo do tempo em um Sprint, um Release ou um produto. A origem dos dados brutos é o Sprint backlog e o Product Backlog, com o trabalho remanescente rastreado no eixo vertical e os períodos de tempo (dias de um Sprint ou Sprints) rastreados no eixo horizontal.

Daily Scrum

Uma breve reunião realizada diariamente por cada Team, durante a qual os membros do Team inspecionam seu trabalho, sincronizam seu trabalho e progresso e relatam e impedimentos ao ScrumMaster que precisam ser removidos. Reuniões subsequentes para adaptar os próximos trabalhos para otimizar o Sprint podem ocorrer após as reuniões do Daily Scrum.

Development Team

Outro nome para o papel Team.

Done

Completo como mutuamente acordado por todas as partes e que está em conformidade com os padrões, convenções e diretrizes de uma organização. Quando algo é relatado como "concluído" na reunião de Sprint Review ele deve estar em conformidade com essa definição acordada.

Estimated Work Remaining (Sprint Backlog items)

O número de horas que um membro do Team estima permanecer trabalhando em qualquer tarefa. Esta estimativa é atualizada no final de cada dia, quando a tarefa do Sprint Backlog é trabalhada. A estimativa é o esforço total estimado restante, independentemente do número de membros que executem o trabalho.

Increment

Funcionalidade do produto que é desenvolvida pelo Team durante cada Sprint e é potencialmente utilizável pelas partes interessadas do Product Owner.

Increment of Potentially Shippable Product Functionality

Uma fatia completa do produto ou sistema geral que poderia ser usado pelo Product Owner ou pelos Stakeholders se eles escolhessem implementá-la.

Product Backlog

Uma lista priorizada de requisitos com tempos estimados para transformá-los em funcionalidades completas do produto. As estimativas são mais precisas quanto mais alto um item estiver na prioridade do Product Backlog. A lista surge, mudando conforme as condições de negócios ou as mudanças tecnológicas.

Product Backlog Item

Requisitos funcionais, requisitos não funcionais e problemas, priorizados em ordem de importância para os negócios e dependências, além de estimados. A precisão da estimativa depende da prioridade e granularidade do item no Product Backlog com os itens de maior prioridade que podem ser selecionados no próximo Sprint sendo muito granulares e precisos.

Product Owner

A pessoa responsável pelo gerenciamento do Product Backlog para maximizar o valor do produto. O Product Owner é responsável por representar os interesses de todos que têm participação no projeto e seu produto resultante.

Scrum

Não é um acrônimo, mas um mecanismos no jogo de rugby para recuperar uma bola fora de jogo.

ScrumMaster

A pessoa responsável pelo processo Scrum, sua correta implementação e a maximização de seus benefícios.

Sprint

Uma iteração, ou um ciclo repetitivo de trabalho semelhante, que produz o Product Increment ou sistema. Não mais do que um mês e geralmente mais de uma semana. A duração é fixada durante todo o trabalho geral e todas os Teams que trabalham no mesmo sistema ou produto usam o mesmo ciclo de duração.

Sprint Backlog

Uma lista do trabalho do Team para um Sprint. Isso é frequentemente decomposto em um conjunto de tarefas mais detalhadas. A lista surge durante o Sprint Planning e pode ser atualizada pelo Team durante o Sprint, com itens sendo removidos ou novas tarefas sendo adicionadas conforme o necessário. Cada tarefa do Sprint Backlog será rastreada durante o Sprint e mostrará o esforço estimado restante.

Sprint Backlog Task

Uma das tarefas que o Team ou um membro do Team define como necessário para transformar itens confirmados do Product Backlog na funcionalidade do sistema.

Sprint Planning meeting

Um horário de reunião com quatro horas (para um Sprint de duas semanas) que inicia cada Sprint. A reunião é dividida em dois segmentos de duas horas, cada um com o time box. Durante a primeira parte, o Product Owner apresenta o Product Backlog de maior prioridade para o Team. O Team e o Product Owner colaboram para ajudar o Team a determinar quanto do Product Backlog pode se transformar em funcionalidade durante o próximo Sprint. Durante a segunda parte, o Team planeja como conseguir isso projetando e decompondo o trabalho para que eles entendam como atingirão o objetivo da Sprint.

Sprint Retrospective meeting

Uma reunião facilitada pelo ScrumMaster na qual o Team completa discute o recém-concluído Sprint e determina o que poderia ser alterado para tornar o próximo Sprint mais agradável ou produtivo.

Sprint Review meeting

Uma reunião de duas horas com time box (para um Sprint de duas semanas) no final de cada Sprint, em que o Team colabora com o Product Owner e os Stakeholders e inspeciona o output do Sprint. Isso geralmente começa com uma revisão dos itens concluídos do Product Backlog, uma discussão de oportunidades, restrições e riscos, além de uma discussão sobre quais podem ser as melhores coisas para fazer a seguir (potencialmente resultando em alterações do Product Backlog). Somente funcionalidades completas do produto pode ser demonstrada.

Stakeholders

Alguém com interesse no resultado de um projeto, seja porque o financiou, vai usá-lo ou será afetado por ele.

Team

Um grupo cross-funtional de pessoas que é responsável por gerenciar-se para desenvolver um Product Increment a cada Sprint.

Time box

Um período de tempo que não pode ser excedido e dentro do qual um evento ou reunião ocorre. Por exemplo, uma reunião diária do Scrum é marcada com quinze minutos e termina no final de quinze minutos, independentemente disso. Para reuniões, pode durar menos. Para Sprints, dura exatamente esse comprimento.