



SCRUM

Fevereiro 2010

Scrum: Desenvolvido e mantido por Ken Schwaber e Jeff Sutherland

Agradecimentos

Geral

Scrum é baseado nas melhores práticas aceitas pelo mercado, utilizadas e provadas por décadas. Ele é definido então em uma teoria de processos empíricos. Como Jim Coplien certa vez observou para o Jeff, “Todos irão gostar de Scrum; isso é o que já fazemos quando nos pressionam contra a parede.”

Pessoas

Das milhares de pessoas que contribuíram para o Scrum, devemos destacar aquelas que contribuíram em seus primeiros dez anos. Primeiro havia o Jeff Sutherland, trabalhando com o Jeff McKenna, e Ken Schwaber com Mike Smith e Chris Martin. Scrum foi formalmente apresentado e publicado no OOPSLA 1995. Durante os cinco anos seguintes, Mike Beadle e Martine Devos fizeram contribuições significativas. E então todos os outros, que sem a sua ajuda o Scrum não teria sido aprimorado no que é atualmente.

História

A história do Scrum já pode ser considerada longa no mundo de desenvolvimento de software. Para honrar os primeiros lugares onde foi testado e aprimorado, honramos a Individual, Inc., Fidelity Investments e IDX (hoje GE Medical).

Tradução

Este guia foi traduzido da versão original em inglês, fornecida por Ken Schwaber e Jeff Sutherland. Em ordem alfabética, realizaram a tradução: Heitor Roriz Filho, Michel Goldenberg e Rafael Sabbagh. Realizaram a revisão os integrantes do grupo *scrumguide_br*, inicialmente formado por: Anderson Marcondes, Ânderson Quadros, Ari do Amaral Torres Filho, Marcos Garrido, Rafael Fuchs, Rafael Prikladnicki, Rodrigo de Toledo e Rafael Sabbagh (coordenação).

Propósito

Scrum vem sendo utilizado para o desenvolvimento de produtos complexos desde o início dos anos 90. Este guia descreve como usar o Scrum para desenvolver produtos. Scrum não é um processo ou uma técnica para o desenvolvimento de produtos. Ao invés disso, é um *framework* dentro do qual você pode empregar diversos processos e técnicas. O papel do Scrum é fazer transparecer a eficácia relativa das suas práticas de desenvolvimento para que você possa melhorá-las, **enquanto provê um *framework* dentro do qual produtos complexos podem ser desenvolvidos.**

Teoria do Scrum

Scrum, que é fundamentado na teoria de controle de processos empíricos, emprega uma abordagem iterativa e incremental para otimizar a previsibilidade e controlar riscos. Três pilares sustentam cada implementação de controle de processos empíricos.

O primeiro pilar é a transparência

A transparência garante que aspectos do processo que afetam o resultado devem ser visíveis para aqueles que gerenciam os resultados. Esses aspectos não apenas devem ser transparentes, mas também o que está sendo visto deve ser conhecido. Isto é, quando alguém que inspeciona um processo acredita que algo está pronto, isso deve ser equivalente à sua definição de pronto.

O segundo pilar é a inspeção

Os diversos aspectos do processo devem ser inspecionados com uma frequência suficiente para que variações inaceitáveis no processo possam ser detectadas. A frequência da inspeção deve levar em consideração que qualquer processo é modificado pelo próprio ato da inspeção. O problema acontece quando a frequência de inspeção necessária excede a tolerância do processo à inspeção. Felizmente, isso não parece ser verdade no desenvolvimento de software. Os

outros fatores são a habilidade e a aplicação das pessoas em inspecionar os resultados do trabalho.

O terceiro pilar é a adaptação

Se o inspetor determinar, a partir da inspeção, que um ou mais aspectos do processo estão fora dos limites aceitáveis e que o produto resultante será inaceitável, ele deverá ajustar o processo ou o material sendo processado. Esse ajuste deve ser feito o mais rápido possível para minimizar desvios posteriores.

Existem três pontos para inspeção e adaptação em Scrum. A Daily Scrum é uma reunião utilizada para inspecionar o progresso em direção à Meta da Sprint e para realizar adaptações que otimizem o valor do próximo dia de trabalho. Além disso, as reuniões de Revisão da Sprint e de Planejamento da Sprint são utilizadas para inspecionar o progresso em direção à Meta da Release e para fazer as adaptações que otimizem o valor da próxima Sprint. Finalmente, a Retrospectiva da Sprint é utilizada para revisar a Sprint passada e definir que adaptações tornarão a próxima Sprint mais produtiva, recompensadora e gratificante.

Conteúdo do Scrum

O *framework* Scrum consiste em um conjunto formado por **Times de Scrum** e seus papéis associados, **Time-Boxes** (eventos com duração fixa), **Artefatos** e **Regras**.

Times de Scrum são projetados para otimizar flexibilidade e produtividade. Para esse fim, eles são auto-organizáveis, multidisciplinares e trabalham em iterações. Cada Time de Scrum possui três papéis: 1) o **ScrumMaster**, que é responsável por garantir que o processo seja compreendido e seguido; 2) o **Product Owner**, que é responsável por maximizar o valor do trabalho que o Time de Scrum faz; e 3) o **Time**, que executa o trabalho propriamente dito. O Time consiste em desenvolvedores com todas as habilidades necessárias para transformar os requisitos do Product Owner em um pedaço potencialmente entregável do produto ao final da Sprint.

Scrum emprega eventos com duração fixa – os time-boxes – para criar regularidade. Entre os elementos do Scrum que têm duração fixa, temos a **Reunião de Planejamento da Release**, a **Reunião de Planejamento da Sprint**, a **Sprint**, a **Daily Scrum**, a **Revisão da Sprint** e a **Retrospectiva da Sprint**. O coração do Scrum é a **Sprint**, que é uma iteração de um mês ou menos, de duração consistente com o esforço de desenvolvimento. Todas as Sprints utilizam o mesmo modelo de Scrum e todas as Sprints têm como resultado um incremento do produto final que é potencialmente entregável. Cada Sprint começa imediatamente após a anterior.

Scrum utiliza quatro artefatos principais. O **Product Backlog** é uma lista priorizada de tudo que pode ser necessário no produto. O **Sprint Backlog** é uma lista de tarefas para transformar o Product Backlog, por uma Sprint, em um incremento do produto potencialmente entregável. Um *burndown* é uma medida do *backlog* restante pelo tempo. Um **Burndown de Release** mede o Product Backlog restante ao longo do tempo de um plano de release. Um **Burndown de Sprint** mede os itens do Sprint Backlog restantes ao longo do tempo de uma Sprint.

As **Regras** fazem o elo entre os time-boxes, os papéis e os artefatos do Scrum. Suas regras são descritas ao longo deste documento. Por exemplo, uma regra do Scrum diz que somente membros do Time - as pessoas comprometidas em transformar o Product Backlog em um incremento - podem falar durante uma Daily Scrum. Modos de implementar Scrum que não são regras, mas sim sugestões, estão descritas nas seções de "Dicas".

Dica

Quando as regras não são declaradas, espera-se que os usuários de Scrum descubram o que devem fazer. Não tente descobrir uma solução perfeita, porque geralmente o problema muda rapidamente. Ao invés disso, tente algo e veja como se sai. Os mecanismos de inspeção-e-adaptação inerentes à natureza empírica de Scrum irão lhe guiar.

Papéis no Scrum

O Time de Scrum é composto pelo ScrumMaster, pelo Product Owner e pelo Time. Os membros do Time de Scrum são chamados "porcos". O Product Owner é o "porco" do Product Backlog. O Time é o "porco" do trabalho da Sprint. O ScrumMaster é o "porco" do processo do Scrum. Qualquer outra pessoa é chamada de "galinha". "Galinhas" não podem dizer aos "porcos" como eles devem fazer seu trabalho. Galinhas e porcos vêm da seguinte história:

Uma galinha e um porco estão juntos quando a galinha diz: "Vamos abrir um restaurante!"

O porco reflete e então diz: "Como seria o nome desse restaurante?"

A galinha diz: "Presunto com Ovos!"

O porco diz: "Não, obrigado, eu estaria comprometido, mas você estaria apenas envolvida!"

Dica

O ScrumMaster trabalha com os clientes e a gerência para identificar e designar um Product Owner. O ScrumMaster ensina ao Product Owner como fazer seu trabalho. Espera-se dos Product Owners que eles saibam como conseguir otimizar valor através do Scrum. Se eles não souberem, consideramos o ScrumMaster responsável.

O ScrumMaster

O ScrumMaster é responsável por garantir que o Time de Scrum esteja aderindo aos valores do Scrum, às práticas e às regras. O ScrumMaster ajuda o Time de Scrum e a organização a adotarem o Scrum. O ScrumMaster educa o Time de Scrum treinando-o e levando-o a ser mais produtivo e a desenvolver produtos de maior qualidade. O ScrumMaster ajuda o

Dica

O ScrumMaster pode ser um membro do Time; por exemplo, um desenvolvedor realizando tarefas da Sprint. No entanto, isso frequentemente leva a conflitos quando o ScrumMaster deve escolher entre remover impedimentos e realizar tarefas. O ScrumMaster nunca deve ser o Product Owner.

Time de Scrum a entender e usar auto-organização e multidisciplinaridade. O ScrumMaster também ajuda o Time de Scrum a fazer o seu melhor em um ambiente organizacional que pode ainda não ser otimizado para o desenvolvimento de produtos complexos. Quando o ScrumMaster ajuda a realizar essas mudanças, isso é chamado de “remoção de impedimentos”. O papel de ScrumMaster é o de líder-servidor para o Time de Scrum.

O Product Owner

O Product Owner é a única pessoa responsável pelo gerenciamento do Product Backlog e por garantir o valor do trabalho realizado pelo Time. Essa pessoa mantém o Product Backlog e garante que ele está visível para todos. Todos sabem quais itens têm a maior prioridade, de forma que todos sabem em que se irá trabalhar.

O Product Owner é uma pessoa, e não um comitê. Podem existir comitês que aconselhem ou influenciem essa pessoa, mas quem quiser mudar a prioridade de um item, terá que convencer o Product Owner. Empresas que adotam Scrum podem perceber que isso influencia seus métodos para definir prioridades e requisitos ao longo do tempo.

Para que o Product Owner obtenha sucesso, todos na organização precisam respeitar suas decisões. Ninguém tem a permissão de dizer ao Time para trabalhar em um outro conjunto de prioridades, e os Times não podem dar ouvidos a ninguém que diga o contrário. As

Dica

Para desenvolvimento comercial, o Product Owner pode ser o Gerente de Produto. Para esforços internos de desenvolvimento, o Product Owner poderia ser o gerente da função de negócios em que se está trabalhando.

Dica

O Product Owner pode ser um membro do Time, trabalhando também em desenvolvimento. Mas essa responsabilidade adicional pode reduzir a sua habilidade de lidar com as partes interessadas. No entanto, o Product Owner nunca pode ser o ScrumMaster.

decisões do Product Owner são visíveis no conteúdo e na priorização do Product Backlog. Essa visibilidade requer que o Product Owner faça seu melhor, o que faz o papel de Product Owner exigente e recompensador ao mesmo tempo.

O Time

Times de desenvolvedores transformam o Product Backlog em incrementos de funcionalidades potencialmente entregáveis em cada Sprint. Times também são multidisciplinares: membros do Time devem possuir todo o conhecimento necessário para criar um incremento no trabalho. Membros do Time frequentemente possuem conhecimentos especializados, como programação, controle de qualidade, análise de negócios, arquitetura, projeto de interface de usuário ou projeto de banco de dados. No entanto, o conhecimento que os membros do Time devem compartilhar - isto é, a habilidade de trabalhar um requisito e transformá-lo em um produto utilizável - tendem a ser mais importantes do que aqueles que eles não dividem. As pessoas que se recusam a programar porque são arquitetas ou designers não se adaptam bem a Times. Todos contribuem, mesmo que isso exija aprender novas habilidades ou lembrar-se de antigas. Não há títulos em Times, e não há exceções a essa regra. Os Times também não contém subtimes dedicados a áreas particulares como testes ou análise de negócios.

Times também são auto-organizáveis. Ninguém - nem mesmo o ScrumMaster - diz ao time como transformar o Product Backlog em incrementos de funcionalidades entregáveis. O Time descobre por si só. Cada membro do Time aplica sua especialidade a todos os problemas. A sinergia que resulta disso melhora a eficiência e eficácia geral do Time como um todo.

O tamanho ótimo para um Time é de sete pessoas, mais ou menos duas pessoas. Quando há menos do que cinco membros em um Time, há menor interação e, como resultado, há menor ganho de produtividade. Mais do que isso, o time poderá encontrar limitações de conhecimento durante partes da Sprint e não ser capaz de entregar

uma parte pronta do produto. Se há mais do que nove membros, há simplesmente a necessidade de muita coordenação. Times grandes geram muita complexidade para que um processo empírico consiga gerenciar. No entanto, temos encontrado alguns Times bem-sucedidos que excederam os limites superior e inferior dessa faixa de tamanhos. O Product Owner e o ScrumMaster não estão incluídos nessa conta, a menos que também sejam porcos, trabalhando em tarefas do Sprint Backlog.

A composição do Time pode mudar ao final da Sprint. Toda vez que o Time muda, a produtividade ganha através da auto-organização é reduzida. Deve-se tomar cuidado ao mudar a composição do Time.

Time-Boxes

Os time-boxes no Scrum são a **Reunião de Planejamento da Release**, a **Sprint**, a **Reunião de Planejamento da Sprint**, a **Revisão da Sprint**, a **Retrospectiva da Sprint** e a **Daily Scrum**.

Reunião de Planejamento da Release

O propósito do planejamento da release é o de estabelecer um plano e metas que o Time de Scrum e o resto da organização possam entender e comunicar. O planejamento da release responde às questões: "Como podemos transformar a visão em um produto vencedor da melhor maneira possível? Como podemos alcançar ou exceder a satisfação do cliente e o Retorno sobre Investimento (ROI) desejados?" O plano da release estabelece a meta da release, as maiores prioridades do Product Backlog, os principais riscos e as características gerais e funcionalidades que estarão contidas na release. Ele estabelece também uma data de entrega e custo prováveis que devem se manter se nada mudar. A organização pode então inspecionar o progresso e fazer mudanças nesse plano da release a cada Sprint.

O planejamento da release é inteiramente opcional. Se um Time de Scrum iniciar o trabalho sem essa reunião, a ausência de seus artefatos aparecerá como um impedimento que deverá ser resolvido.

O trabalho para resolver o impedimento se tornará um item no Product Backlog.

Ao se utilizar Scrum, os produtos são construídos iterativamente, de modo que cada Sprint cria um incremento do produto, iniciando pelo de maior valor e maior risco. Mais e mais Sprints vão adicionando incrementos ao produto. Cada incremento é um pedaço potencialmente entregável do produto completo. Quando já tiverem sido criados incrementos suficientes para que o produto tenha valor e uso para seus investidores, o produto é entregue.

Muitas organizações já possuem um processo de planejamento de release, e na maior parte desses processos o planejamento é feito no início do trabalho da release e não é modificado com o passar do tempo. No planejamento de release do Scrum, são definidos uma meta geral e resultados prováveis. Esse planejamento geralmente não requer mais do que 15-20% do tempo que uma organização costumava utilizar para produzir um plano de release tradicional. No entanto, uma release com Scrum realiza planejamento no momento da execução de cada reunião de Revisão de Sprint e de Planejamento de Sprint, da mesma forma que realiza um planejamento diário no momento da execução de cada Daily Scrum. De forma geral, os esforços para uma release com Scrum provavelmente consomem ligeiramente mais esforço do que os esforços para um planejamento de release tradicional.

O planejamento da release requer estimar e priorizar o Product Backlog para a release. Há diversas técnicas para fazê-lo que estão fora do alcance do Scrum, mas que apesar disso são úteis quando usadas com ele.

A Sprint

A Sprint é uma iteração. Sprints têm duração fixa. Durante a Sprint, o ScrumMaster garante que não será feita nenhuma mudança que possa afetar a Meta da Sprint. Tanto a composição do time quanto as metas de qualidade devem permanecer constantes durante a Sprint. As Sprints contêm e consistem na reunião de Planejamento de Sprint, o trabalho de desenvolvimento, a Revisão da Sprint e a Retrospectiva da Sprint. As Sprints ocorrem uma após a outra, sem intervalos entre elas.

Um projeto serve para cumprir alguma função. Em desenvolvimento de *software*, o projeto é utilizado para desenvolver um produto ou sistema. Cada projeto consiste em uma definição do que será desenvolvido, um plano para desenvolvê-lo, o trabalho realizado de acordo com o plano e o produto resultante. Cada projeto possui um horizonte, que é o período de tempo para o qual o plano é válido. Se o horizonte for longo demais, a definição poderá ter mudado, variáveis demais poderão ter surgido, o risco poderá ser grande demais etc. Scrum é um *framework* para projetos cujo horizonte não é superior ao período de um mês, onde já há complexidade suficiente tal que um horizonte mais longo seria arriscado demais. A previsibilidade do projeto deve ser controlada pelo

Dica

Se o time sentir que se comprometeu com mais do que podia, ele se encontra com o Product Owner para remover ou reduzir o escopo do Product Backlog selecionado para a Sprint. Se o time sentir que sobrar tempo, ele pode trabalhar junto ao Product Owner para selecionar mais do Product Backlog.

Dica

Quando um Time começa com o Scrum, Sprints de duas semanas o permite aprender sem se afundar na incerteza. Sprints desse tamanho podem ser sincronizadas com outros Times adicionando-se dois incrementos juntos.

menos a cada mês, e o risco de que o projeto saia de controle ou se torne imprevisível é contido pelo menos a cada mês.

As Sprints podem ser canceladas antes que o prazo fixo da Sprint tenha acabado. Somente o Product Owner tem a autoridade para cancelar a Sprint, embora ele possa fazê-lo sob influência das partes interessadas, do Time ou do ScrumMaster. Sob que tipo de circunstâncias pode ser necessário cancelar uma Sprint? A gerência pode precisar cancelar uma Sprint se a Meta da Sprint se tornar obsoleta. Isso pode ocorrer se a empresa mudar de direção ou se as condições do mercado ou tecnologia mudarem. Em geral, uma Sprint deve ser cancelada se ela não fizer mais sentido dadas as circunstâncias atuais. Porém, por causa da curta duração das Sprints, raramente isso faz sentido.

Quando uma Sprint é cancelada, todos os itens do Product Backlog que estejam completados e "feitos" são revisados. Eles são aceitos se representarem um incremento potencialmente entregável. Todos os outros itens do Product Backlog são devolvidos ao Product Backlog com suas estimativas iniciais. Assume-se que qualquer trabalho realizado nesses itens é perdido. Cancelamentos de Sprints consomem recursos, já que todos têm que se reagrupar em outra reunião de Planejamento de Sprint para iniciar uma nova Sprint. Cancelamentos de Sprints são frequentemente traumáticos para o Time e são muito incomuns.

Reunião de Planejamento da Sprint

A Reunião de Planejamento da Sprint é o momento no qual a iteração é planejada. É fixada em oito horas de duração para uma Sprint de um mês. Para Sprints mais curtas, deve-se alocar proporcionalmente ao tamanho total da Sprint para essa reunião (por exemplo, para duas semanas teríamos uma Reunião de Planejamento da Sprint de quatro horas). A Reunião de Planejamento da Sprint consiste de duas partes. A primeira parte é o momento no qual é decidido o que será feito na Sprint. A segunda parte (com duração fixa de quatro horas para uma Sprint de um mês) é o momento no qual o Time entende como

desenvolverá essa funcionalidade em um incremento do produto durante a Sprint.

Há duas partes na Reunião de Planejamento da Sprint: a parte do “o quê?” e a parte do “como?”. Alguns Times de Scrum juntam as duas. Na primeira parte, o Time de Scrum trata da questão do “o quê?”. Aqui, o Product Owner apresenta ao Time o que é mais prioritário no Product Backlog. Eles trabalham em conjunto para definir qual funcionalidade deverá ser desenvolvida durante a próxima Sprint. As entradas para essa reunião são o Product Backlog, o incremento mais recente ao produto, a capacidade do Time e o histórico de desempenho do Time. Cabe somente ao Time a decisão de quanto do Backlog ele deve selecionar. Somente o Time pode avaliar o que ele é capaz de realizar na próxima Sprint.

Uma vez selecionado o Product Backlog, a Meta da Sprint é delineada. A Meta da Sprint é o objetivo que será atingido através da implementação do Product Backlog. Ela é uma descrição que fornece orientação ao Time sobre a razão pela qual ele está desenvolvendo o incremento. A Meta da Sprint é um subconjunto da Meta da Release.

O motivo para se ter uma Meta da Sprint é dar ao time alguma margem com relação à funcionalidade. Por exemplo, a meta para a Sprint acima poderia também ser: “Automatizar a funcionalidade de modificação de conta de clientes através de um *middleware* seguro capaz de recuperar transações.” Conforme o Time trabalha, ele mantém a meta em mente. Para satisfazer a meta, ele implementa a funcionalidade e a tecnologia. Se o trabalho se mostrar mais difícil do que o time esperava, o time então irá trabalhar em conjunto com o Product Owner e implementar a funcionalidade apenas parcialmente.

Na segunda parte da Reunião de Planejamento da Sprint, o Time trata a questão do “como?”. Durante a segunda parte da Reunião de Planejamento da Sprint (com duração fixa de quatro horas para Sprints de um mês), o Time define como transformará o Product Backlog selecionado durante a Reunião de Planejamento (o quê) em um incremento pronto. O Time geralmente começa projetando o

trabalho. Enquanto projeta, o time identifica tarefas. Essas tarefas são pedaços detalhados do trabalho necessário para converter o Product Backlog em *software* funcional. As tarefas devem ser decompostas para que possam ser feitas em menos de um dia. Essa lista de tarefas é chamada de Sprint Backlog. O time se auto-organiza para se responsabilizar pelo trabalho contido no Sprint Backlog, tanto durante a Reunião de Planejamento da Sprint quanto no próprio momento da execução da Sprint.

O Product Owner está presente durante a segunda parte da Reunião de Planejamento da Sprint para esclarecer o Product Backlog e para ajudar a efetuar trocas. Se o Time concluir que ele tem trabalho demais ou de menos, ele pode renegociar o Product Backlog escolhido com o Product Owner. O Time também

Dica

Geralmente, somente 60-70% do total do Sprint Backlog será definido na Reunião de Planejamento da Sprint. O restante é deixado para ser detalhado mais tarde ou são dadas estimativas grandes que serão decompostas mais tarde na Sprint

pode convidar outras pessoas a participarem da reunião para fornecerem conselhos técnicos ou sobre o domínio em questão. Geralmente, um Time novo percebe pela primeira vez se irá ganhar ou perder como um Time - não individualmente - nessa reunião. O Time percebe que deve contar consigo mesmo. Conforme ele percebe isso, ele começa a se auto-organizar para assumir as características e comportamento de um verdadeiro Time.

Revisão da Sprint

Ao final da Sprint, é feita a reunião de Revisão da Sprint. Para Sprints de um mês, essa é uma reunião com duração fixa de quatro horas. Para Sprints de durações mais curtas, deve-se alocar proporcionalmente menos do tamanho total da Sprint para essa reunião (por exemplo, para duas semanas deve-se ter uma Revisão da Sprint de duas horas). Durante a Revisão da Sprint, o Time de Scrum e as partes interessadas colaboram sobre o que acabou de ser feito.

Baseados nisso e em mudanças no Product Backlog feitas durante a Sprint, eles colaboram sobre quais são as próximas coisas que podem ser feitas. Essa é uma reunião informal, com a apresentação da funcionalidade, que tem a intenção de promover a colaboração sobre o que fazer em seguida.

A reunião inclui ao menos os seguintes elementos. O Product Owner identifica o que foi feito e o que não foi feito. O Time discute sobre o que correu bem durante a Sprint e quais problemas foram enfrentados, além de como esses problemas foram resolvidos. O Time então demonstra o trabalho que está pronto e responde a questionamentos. O Product Owner então discute o Product Backlog da maneira como esse se encontra. Ele faz projeções de datas de conclusão prováveis a partir de várias hipóteses de velocidade. Em seguida, o grupo inteiro colabora sobre o que foi visto e o que isso significa com relação ao que fazer em seguida. A Revisão da Sprint fornece entradas valiosas para as reuniões de Planejamento de Sprints seguintes.

Retrospectiva da Sprint

Após a Revisão da Sprint e antes da próxima reunião de Planejamento da Sprint, o Time de Scrum tem a reunião de Retrospectiva da Sprint. Esta é uma reunião com duração fixa de três horas para Sprints de um mês (deve-se alocar proporcionalmente ao tamanho total da Sprint para essa reunião). Nessa reunião, o ScrumMaster encoraja o Time a revisar, dentro do modelo de trabalho e das práticas do processo do Scrum, seu processo de desenvolvimento, de forma a torná-lo mais eficaz e gratificante para a próxima Sprint. Muitos livros documentam técnicas que são úteis em Retrospectivas.

A finalidade da Retrospectiva é inspecionar como correu a última Sprint em se tratando de pessoas, das relações entre elas, dos processos e das ferramentas. A inspeção deve identificar e priorizar os principais itens que correram bem e aqueles que, se feitos de modo diferente, poderiam ter deixado as coisas ainda melhores. Isso inclui a composição do Time de Scrum, preparativos para reuniões,

ferramentas, definição de “pronto”, métodos de comunicação e processos para transformar itens do Product Backlog em alguma coisa “pronta”. No final da Retrospectiva da Sprint, o Time de Scrum deve ter identificado medidas de factíveis melhoria que ele implementará na próxima Sprint. Essas mudanças se tornam a adaptação para a inspeção empírica.

Daily Scrum

Cada time se encontra diariamente para uma reunião de 15 minutos chamada Daily Scrum. Essa reunião é sempre feita no mesmo horário e no mesmo local durante as Sprints. Durante a reunião, cada membro explica:

1. O que ele realizou desde a última reunião;
2. O que ele vai fazer antes da próxima reunião; e
3. Quais obstáculos estão em seu caminho.

As Daily Scrums melhoram a comunicação, eliminam outras reuniões, identificam e removem impedimentos para o desenvolvimento, ressaltam e promovem a tomada rápida de decisões e melhoram o nível de conhecimento de todos acerca do projeto.

O ScrumMaster garante que o Time realize essa reunião. O Time é responsável por conduzir a Daily Scrum. O ScrumMaster ensina o time a manter a Daily Scrum com curta duração, reforçando as regras e garantido que as pessoas falem brevemente. O ScrumMaster também enfatiza a regra de que galinhas não estão autorizadas a falar e nem a interferir de modo algum na Daily Scrum.

A Daily Scrum não é uma reunião de status. Ela é só para as pessoas que estão transformando os itens do Product Backlog em um incremento (o Time), e para mais ninguém. O Time se comprometeu com uma Meta da Sprint, e a esses itens do Product Backlog. A Daily Scrum é uma inspeção do progresso na direção da Meta da Sprint (as três perguntas). Geralmente acontecem reuniões subsequentes para

realizar adaptações ao trabalho que está por vir na Sprint. A intenção é otimizar a probabilidade de que o Time alcance sua Meta. Essa é uma reunião fundamental de inspeção e adaptação no processo empírico do Scrum.

Artefatos do Scrum

Os artefatos do Scrum incluem o Product Backlog, o Burndown da Release, o Sprint Backlog e o Burndown da Sprint.

Product Backlog e Burndown da Release

Os requisitos para o produto que o(s) Time(s) está(ão) desenvolvendo estão listados no Product Backlog. O Product Owner é o responsável pelo Product Backlog, por seu conteúdo, por sua disponibilidade e por sua priorização. O Product Backlog nunca está completo. A seleção inicial para o seu desenvolvimento somente mostra os requisitos inicialmente conhecidos e melhor entendidos. O Product Backlog evolui à medida que o produto e o ambiente em que ele será usado evoluem. O Backlog é dinâmico, no sentido de que ele está constantemente mudando para identificar o que o produto necessita para ser apropriado, competitivo e útil. Enquanto existir um produto, o Backlog de Produto também existirá.

O Product Backlog representa tudo que é necessário para desenvolver e lançar um produto de sucesso. É uma lista de todas as características, funções, tecnologias, melhorias e correções de defeitos que constituem as mudanças que serão efetuadas no produto para releases futuras. Os itens do Product Backlog possuem os atributos de descrição, prioridade e estimativa. A prioridade é determinada por risco, valor e necessidade. Há diversas técnicas para dar valor a esses atributos.

Dica

Os itens do Product Backlog são geralmente representados como Estórias de Usuário. Casos de Uso também são apropriados, mas são mais adequados para desenvolvimento de softwares críticos em termos de vidas ou de desempenho.

O Product Backlog é ordenado por prioridade. O Product Backlog de mais alta prioridade leva a atividades de desenvolvimento imediatas. Quanto mais alta sua prioridade, mais urgente ele é, mais se pensou sobre ele e há mais consenso no que diz respeito ao seu valor. O Backlog de mais alta prioridade é mais claro e possui mais informações detalhadas do que o Backlog de prioridade mais baixa. Fazem-se melhores estimativas quando baseadas em uma maior clareza e em mais detalhes. Quanto mais baixa a prioridade, menor é o nível de detalhe, até que mal se consiga entender o item.

À medida que um produto é utilizado, que seu valor aumenta e que o mercado fornece *feedback*, o Product Backlog torna-se uma lista maior e mais aprofundada. Os requisitos nunca param de mudar. O Product Backlog é um documento vivo. Mudanças nos requisitos de negócios, condições do mercado, tecnologia e equipe causam mudanças no Product Backlog. Para minimizar o retrabalho, apenas os itens de maior prioridade precisam ser mais detalhados. Os itens do Product Backlog que ocuparão os Times de Scrum pelas várias Sprints seguintes deverão ter

Dica

Os Times Scrum geralmente gastam 10% de cada Sprint preparando o Product Backlog para adequá-lo à definição de Product Backlog feita acima. Quando estiverem adequados a esse nível de granularidade, os itens no topo do Product Backlog (de maior prioridade e de maior valor) são decompostos de forma que caibam em um Sprint. Eles foram analisados e se refletiu sobre eles durante esse processo de preparação. Quando ocorre a reunião de Planejamento de Sprint, esses itens de maior prioridade do Product Backlog estão bem entendidos e são facilmente selecionados.

Dica

Testes de aceitação são frequentemente utilizados como um outro atributo para o item do Product Backlog. Eles podem frequentemente substituir descrições textuais mais detalhadas com uma descrição testável do que o item do Product Backlog deve fazer uma vez que esteja completo.

granularidade mais fina, tendo sido decompostos de forma tal que cada um dos itens possa ser feito dentro da duração da Sprint.

Frequentemente, múltiplos Times de Scrum trabalham juntos no mesmo produto. Um único Product Backlog é usado para descrever o trabalho a ser realizado no produto. Então, um atributo que agrupe os itens é aplicado no Product Backlog. O agrupamento pode ocorrer por conjuntos de características, por tecnologia ou por arquitetura, e ele é frequentemente utilizado como uma forma de se organizar o trabalho por Time de Scrum.

O gráfico de Burndown da Release registra a soma das estimativas dos esforços restantes do Product Backlog ao longo do tempo. O esforço estimado deve estar em qualquer unidade de medida de trabalho que o Time de Scrum e a organização tenham decidido usar. As unidades de tempo geralmente são Sprints.

As estimativas dos itens do Product Backlog são inicialmente calculadas durante o Planejamento da Release, e posteriormente à medida que itens forem sendo criados. Durante a preparação do Backlog de Produto, os itens são revistos e revisados. Entretanto, eles podem ser atualizados em qualquer momento. O Time é responsável por todas as estimativas. O Product Owner

Dica

Em algumas organizações, acrescenta-se mais trabalho ao Backlog do que se realiza. Isso pode criar uma linha de tendência plana ou até com inclinação crescente. Para compensar isso e manter a transparência, um novo piso pode ser criado quando se adiciona ou quando se subtrai trabalho. O piso deve adicionar ou remover somente mudanças significativas e deve ser bem documentado.

Dica

A linha de tendência pode não ser confiável pelas duas ou três primeiras Sprints de uma release, a menos que os times já tenham trabalhado juntos anteriormente, conheçam bem o produto e entendam a tecnologia envolvida.

pode influenciar o Time, ajudando-o a entender e a escolher as mudanças, mas a estimativa final é feita pelo Time. O Product Owner mantém o Product Backlog e o Burndown do Backlog da Release atualizados e publicados todo o tempo. Uma linha de tendência pode ser traçada baseada na mudança do trabalho restante.

Sprint Backlog e Burndown da Sprint

O Sprint Backlog consiste nas tarefas que o time executa para transformar itens do Product Backlog em um incremento “pronto”. Muitas delas são elaboradas durante a Reunião de Planejamento da Sprint. O Sprint Backlog é todo trabalho que o Time identifica como necessário para alcançar a Meta da Sprint. Os itens do Sprint Backlog devem ser decompostos. A decomposição deve ser suficiente para que mudanças no progresso possam ser entendidas na Daily Scrum. Um dia ou menos é um tamanho comum para um item do Sprint Backlog no qual se está trabalhando.

O Time modifica o Sprint Backlog no decorrer da Sprint, bem como surge Sprint Backlog durante a Sprint. Quando chega às tarefas individuais, o Time pode descobrir que mais ou menos tarefas serão necessárias, ou que uma determinada tarefa levará mais ou menos tempo do que era esperado. À medida que novo trabalho surge, o Time o adiciona ao Sprint Backlog. À medida que se trabalha nas tarefas ou que elas são completadas, o trabalho restante para cada tarefa é atualizado. Quando se verifica que determinadas tarefas são desnecessárias, elas são removidas. Somente o Time pode modificar o seu Sprint Backlog durante uma Sprint. Somente o Time pode mudar o seu conteúdo ou as suas estimativas. O Sprint Backlog é um retrato em tempo real altamente visível do trabalho que o Time planeja efetuar durante a Sprint, e ele pertence unicamente ao Time.

O Burndown do Sprint Backlog é um gráfico da quantidade restante de trabalho do Sprint Backlog em uma determinada Sprint ao longo do tempo da Sprint. Para criar esse gráfico, determine quanto trabalho

resta somando as estimativas do Backlog a cada dia da Sprint. A quantidade de trabalho restante para uma Sprint é a soma do trabalho restante para tudo do Sprint Backlog. Acompanhe essas somas diariamente e utilize-as para criar um gráfico que mostre o trabalho restante ao longo do tempo. Traçando uma linha através dos pontos no gráfico, o Time pode gerenciar seu progresso em completar o trabalho de uma Sprint. A duração não é considerada em Scrum. O trabalho restante e a data são as únicas variáveis de interesse.

Dica

Sempre que possível, desenhe o gráfico de burndown à mão em uma folha grande de papel exibida na área de trabalho do Time. É mais provável que o Time veja um gráfico grande e visível do que um gráfico de Burndown da Sprint no Excel ou em alguma ferramenta.

Uma das regras do Scrum está ligada ao objetivo de cada Sprint, que é ter como resultado incrementos de funcionalidade potencialmente entregáveis que estejam de acordo com uma definição de “pronto” operacional.

Pronto

Scrum exige que os Times desenvolvam um incremento de funcionalidade do produto a cada Sprint. Esse incremento deve ser potencialmente entregável, pois o Product Owner pode optar por implantar a funcionalidade imediatamente. Para isso ser possível, o incremento deve ser um pedaço completo do produto. Ele deve estar “pronto” (ou “*done*”, em inglês). Cada incremento deve ser adicionado a todos os incrementos anteriores e exaustivamente testado, garantindo que todos os incrementos funcionem juntos.

No desenvolvimento de produtos, afirmar que a funcionalidade está pronta pode levar alguém a presumir que ela está pelo menos bem codificada, refatorada, que tenha passado por testes unitários, que tenha sido feito o *build* e que tenha passado por testes de aceitação. Outros podem presumir que apenas o código tenha sido desenvolvido. Se ninguém sabe qual a definição de “pronto”, os outros dois pilares

do controle de processos empíricos não funcionam. Quando alguém descreve algo como “pronto”, todos devem entender o que “pronto” significa.

“Pronto” define o que o Time quer dizer quando se compromete a “aprontar” um item de Product Backlog em uma Sprint. Alguns produtos não contêm documentação, de forma que sua definição de “pronto” não inclui documentação. Um incremento completamente “pronto” inclui toda a análise, projeto, refatoramento, programação, documentação e testes para o incremento e todos os itens do Product Backlog no incremento. Os testes incluem testes de unidade, de sistema, de usuário e de regressão, bem como testes não-funcionais como de performance, de estabilidade, de segurança e de integração. “Pronto” inclui também qualquer internacionalização necessária. Alguns Times ainda não são capazes de incluir em sua definição de “pronto” tudo o que é necessário para a implantação. Isto deve estar claro para o Product Owner. Esse trabalho restante deverá ser feito antes que o produto possa ser implantado e utilizado.

Dica

Trabalho "não pronto" é geralmente acumulado em um item do Product Backlog chamado "Trabalho Não Pronto" ou "Trabalho de Implantação". À medida que esse trabalho acumula, o Burndown do Product Backlog se mantém mais preciso do que se esse trabalho não fosse acumulado.

Considerações Finais

Algumas organizações são incapazes de desenvolver um incremento completo dentro de uma Sprint. Elas podem ainda não ter infraestrutura automatizada de testes para completar todos os testes. Nesse caso, duas categorias são criadas para cada incremento: o trabalho “pronto” e o trabalho “não pronto”. O trabalho “não pronto” é a porção de cada incremento que terá que ser completada mais tarde. O Product Owner sabe exatamente o que ele está inspecionando ao final da Sprint, porque o incremento atende à definição de “pronto” e o Product Owner entende essa definição. O trabalho “não pronto” é adicionado a um item do Product Backlog

chamado "trabalho não pronto", de forma que ele se acumula e isso é refletido corretamente no gráfico de Burndown da Release. Essa técnica cria transparência no progresso em direção à entrega. A inspeção e a adaptação na Revisão da Sprint serão tão precisas quanto essa transparência for.

Por exemplo, se um Time não é capaz de realizar os testes de performance, regressão, estabilidade, segurança e integração para cada item do Product Backlog, a proporção desse trabalho em relação ao trabalho que pode ser pronto (análise, projeto, refatoramento, programação, documentação, testes unitário e testes de usuário) é calculada. Vamos supor que essa proporção é de seis peças de "pronto" e quatro peças de "não pronto". Se o Time terminar um item de Backlog de Produto de seis unidades de trabalho (o Time está estimando baseado no que ele sabe como "aprontar"), quatro unidades serão acrescentadas ao item do Product Backlog denominado "trabalho não pronto" quando eles tiverem terminado.

Sprint a Sprint, o trabalho "não pronto" de cada incremento vai se acumulando e deve ser tratado antes da entrega do produto. Esse trabalho é acumulado linearmente, embora haja algum tipo de acúmulo exponencial que é dependente das características de cada organização. Sprints de Release são adicionados ao final de cada release para completar o trabalho "não pronto". O número de Sprints é imprevisível, visto que o acúmulo de trabalho "não pronto" não é linear.