

## SCRUM

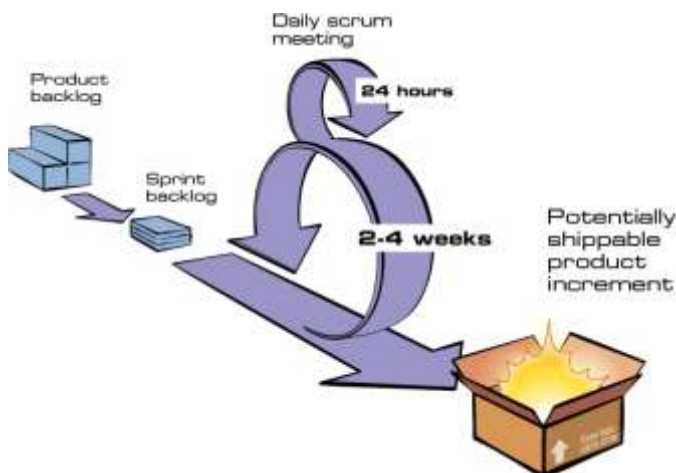
Scrum é uma metodologia ágil para gestão e planejamento de projetos de software.

No Scrum, os projetos são divididos em ciclos (tipicamente mensais) chamados de Sprints. O Sprint representa um Time Box dentro do qual um conjunto de atividades deve ser executado. Metodologias ágeis de desenvolvimento de software são iterativas, ou seja, o trabalho é dividido em iterações, que são chamadas de Sprints no caso do Scrum.

As funcionalidades a serem implementadas em um projeto são mantidas em uma lista que é conhecida como Product Backlog. No início de cada Sprint, faz-se um Sprint Planning Meeting, ou seja, uma reunião de planejamento na qual o Product Owner prioriza os itens do Product Backlog e a equipe seleciona as atividades que ela será capaz de implementar durante o Sprint que se inicia. As tarefas alocadas em um Sprint são transferidas do Product Backlog para o Sprint Backlog.

A cada dia de uma Sprint, a equipe faz uma breve reunião (normalmente de manhã), chamada Daily Scrum. O objetivo é disseminar conhecimento sobre o que foi feito no dia anterior, identificar impedimentos e priorizar o trabalho do dia que se inicia.

Ao final de um Sprint, a equipe apresenta as funcionalidades implementadas em uma Sprint Review Meeting. Finalmente, faz-se uma Sprint Retrospective e a equipe parte para o planejamento do próximo Sprint. Assim reinicia-se o ciclo. Veja a ilustração abaixo:



O que é Scrum:

Scrum é uma metodologia usada para a gestão dinâmica de projetos, sendo muitas vezes aplicada para o desenvolvimento ágil de um software.

O scrum é uma ferramenta que permite controlar de forma eficaz e eficiente o trabalho, potencializando as equipes que trabalham em prol de um objetivo em comum.

Esta metodologia é essencial para muitas empresas atualmente, porque não apenas facilita a definição de objetivos, como também ajuda a cumprir os prazos estabelecidos.

No scrum se trabalha com o chamado product backlog, um registro que contém as áreas do produto que devem ser desenvolvidas. Do product backlog é criado o release backlog, que é a junção dos requisitos do product backlog que vão ser trabalhados, de acordo com a prioridade de cada um. O release backlog é um ponto para a criação do sprint backlog, que representa o espaço de tempo em que uma tarefa (chamada de user story) vai ser concluída.

O tamanho de cada sprint é adequado à empresa em questão e aos seus projetos. A sprint pode demorar entre uma a quatro semanas. O processo de scrum costuma ser controlado em um quadro, onde é possível ver as tarefas que estão em desenvolvimento, as que foram trabalhadas, mas que ainda precisam ser verificadas ou testadas, e as que são consideradas concluídas.

Alguns dos elementos que fazem parte do processo do Scrum são:

**Product owner:** é o dono do produto ou projeto que vai ser trabalhado, sendo responsável pela direção a seguir, definindo quais requisitos vão fazer parte do product backlog e quais devem ser abordados pela equipe. Representa os usuários ou clientes do produto em questão;

**Scrum Master:** é o elemento que faz a ligação entre o product owner e a equipe. Tem a responsabilidade de organizar reuniões, fazer o acompanhamento do trabalho e se certificar que cada integrante da equipe tem as ferramentas necessárias para cumprir a sua função da melhor maneira possível.

**Team (equipe):** É a equipe que trabalha para o desenvolvimento do projeto ou produto.

Outro conceito relevante nesta área é o daily scrum, ou scrum diário, que consiste em uma reunião organizada pelo Scrum Master. Todos os elementos estão em pé, para que a reunião seja de curta duração (máximo 15 minutos). Esta reunião é uma forma de comprovar que cada elemento está cumprindo o seu papel.

Outra reunião importante no âmbito do scrum é a de planejamento da próxima sprint, onde é definido quanto tempo vai durar cada tarefa. O standard para a medição do tempo de cada tarefa pode ser atribuição de pontos ou tamanhos de camiseta (XL, L, M, S, XS), sendo que uma tarefa que demora mais tem mais pontos. Desta forma, é possível somar os pontos no fim da sprint e averiguar a velocidade de trabalho da equipe.

A monitorização do progresso de cada sprint é feita através da burndown chart (tabela burndown), uma das características que torna o scrum tão popular.

Consiste em uma tabela que permite controlar se um projeto está se desenvolvendo da forma programada. Ela apresenta uma medição diária da quantidade de trabalho que ainda não foi feito em cada sprint ou release. Esta tabela também permite fazer uma estimativa do tempo em que a sprint vai ser concluída. Assim, é possível saber se o projeto está progredindo de acordo com o tempo estimado ou se vai sofrer algum atraso. Essa informação pode ser usada pela equipe para fazer alguns ajustes no seu trabalho, impedindo que o atraso se verifique realmente.

A origem do termo scrum vem do esporte rúgbi, onde scrum define a aglomeração dos jogadores, muitas vezes vista como "formação ordenada". No scrum, 8 jogadores de cada time estão frente a frente e têm que fazer um esforço para recuperar a bola que se encontra no meio do "aglomerado".

A metodologia Agile, como o próprio nome já diz, é utilizada para tornar os processos empresariais mais ágeis, sobretudo o desenvolvimento de sistemas. Dentro das metodologias ágeis, o framework Scrum é um dos mais difundidos e utilizados. Juntos permitem controlar de forma eficiente as atividades realizadas, incentivando as equipes a trabalharem com foco em um objetivo comum.

Além de otimizar a definição de metas, o Scrum assegura a geração de valor em um projeto, uma forma de trabalho inovadora que tem sido adotada por grandes empresas.

### **O Que É Metodologia Agile**

A metodologia Agile, ou ágil em português, se consolidou nos últimos anos como uma alternativa para atender às demandas de clientes e projetos de forma dinâmica, flexível e com grande aumento de produtividade.

No desenvolvimento ágil, é utilizada uma abordagem de planejamento iterativa. Enquanto no método tradicional todas as etapas do projeto são documentadas detalhadamente, desde o início até o fim do projeto, esse processo no método ágil é realizado em etapas curtas, chamadas iterações.

### **Fundamentos Da Metodologia Agile**

A metodologia Agile, nos moldes como é conhecida atualmente, foi concebida no início de 2001.

Um grupo de 17 conceituados desenvolvedores de software se reuniu para aprimorar conceitos e metodologias ágeis existentes e formular o "Manifesto para o Desenvolvimento Ágil de Software". Assinado pelos 17 desenvolvedores, ele reúne quatro valores e 12 princípios.

Os valores determinam o que deve ser priorizado:

Os indivíduos e as interações entre eles mais que os processos e as ferramentas;

O software funcionando mais do que uma documentação completa e abrangente;

A colaboração com e dos clientes mais do que as negociações de contratos; e

Respostas às mudanças mais do que seguir o plano inicial.

Já os princípios são:

A maior prioridade é satisfazer o cliente com a entrega adiantada e contínua de software de valor;

Aceitar mudanças de requisitos, mesmo no fim do desenvolvimento. Processos ágeis se adequam a mudanças para que o cliente possa tirar vantagens competitivas;

Entregar software funcionando com frequência, preferencialmente em semanas;

Cooperação diária entre pessoas que entendem do "negócio" e desenvolvedores;

Projetos surgem por meio de indivíduos motivados, entre os quais existe relação de confiança.

A maneira mais eficaz e eficiente de transmitir informações é por conversas frente a frente;

Softwares funcionais são a principal medida de progresso do projeto;

Processos ágeis promovem um ambiente sustentável. Os patrocinadores, desenvolvedores e usuários devem ser capazes de manter, indefinidamente, passos constantes;

Contínua atenção à excelência técnica e bom design aumenta a agilidade;

Simplicidade é essencial. Cultivar a arte de maximizar a quantidade de trabalho que não precisou ser feito;

As melhores arquiteturas, requisitos e projetos emergem de equipes auto-organizadas;

Em intervalos regulares, o time reflete sobre como se tornar mais efetivo e, então, se ajustam e otimizam seu comportamento de acordo.

O funcionamento do framework Scrum

Entre as diversas metodologias ágeis usadas por desenvolvedores, o framework Scrum é uma das mais difundidas, especialmente pelo formato dinâmico como as etapas dos projetos são desenvolvidas.

Em um projeto realizado utilizando o Scrum, a execução acontece em iterações, as chamadas sprints.

Elas são ciclos de desenvolvimento que começam em uma reunião de planejamento (Sprint Planning) e terminam com outros dois eventos: a revisão da sprint (Sprint Review), em que o produto é demonstrado; e a retrospectiva da sprint (Sprint Retrospective), em que o time afia seus processos ao refletir sobre os aprendizados que tiveram naquele ciclo. O desenvolvimento é acompanhado por reuniões diárias em pé, as chamadas Daily Scrum.

Uma sprint pode durar qualquer unidade de tempo, mas a recomendação é que as mais curtas levem uma semana e as mais longas, um mês. Quando um time define um padrão de duração da Sprint, ele costuma ser usado em todas as iterações do projeto. As sprints são sucessivas e, assim, a seguinte somente é iniciada ao fim da anterior.

No Scrum, três papéis são definidos para a divisão dos membros envolvidos num projeto:

**Product Owner:** atua como o "dono" do projeto. Será o responsável por definir prioridades a serem desenvolvidas em cada sprint e fará a intermediação entre a área de negócios e a equipe de scrum.

**Scrum Master:** atua como um líder-servo, blindando os demais membros e assegurando que a equipe siga a metodologia do Scrum sem interrupções externas. É também o responsável por remover

obstáculos que possam prejudicar o desenvolvimento realizado pela equipe e ajudá-la a cumprir suas tarefas com a melhor performance possível.

**Scrum Team:** a equipe de desenvolvimento. Todos devem se comprometer em realizar as entregas estabelecidas dentro de uma Sprint. Para tal, é necessário que haja maturidade suficiente de cada membro para a execução de sua função e, quando preciso, solicite auxílio. Para a eficiência do Scrum, é recomendável que essas equipes sejam compostas por membros multidisciplinares e não envolvam muitos participantes.

#### Sprint Planning

O planejamento da sprint é realizado nessa reunião, em que será definido como o trabalho da equipe será feito dentro do período estabelecido. Nesse momento, devem ser priorizadas as atividades do product backlog — descrições de todas as funcionalidades desejadas para o produto — que passarão a integrar o sprint backlog.

É importante ressaltar que o backlog é composto pelo repositório de projetos e ações da empresa. Em cada sprint, seleciona-se, de acordo com a prioridade e dificuldade do projeto, os projetos que sairão do backlog e comporão a sprint.

#### Daily Scrum

Diariamente, a equipe de desenvolvimento deve se reunir para discutir aquilo que tem sido desenvolvido dentro da sprint. Essa reunião deve ser rápida e, preferencialmente, não durar mais que 15 minutos. Nela, serão respondidas três perguntas por cada membro da equipe:

O que foi feito ontem para ajudar a equipe de desenvolvimento a atender a meta da sprint?

O que será feito hoje para ajudar a equipe de desenvolvimento a atender a meta da sprint?

Há algum obstáculo que impeça o atendimento da meta da sprint?

Se houver uma resposta afirmativa para a última pergunta, o Scrum Master deverá buscar rapidamente uma solução para que o desenvolvimento da sprint siga conforme o planejado. Por isso é importante que ele tenha autonomia na organização para corrigir problemas no processo.

#### Sprint Meeting Review

Ao final da sprint, uma reunião de revisão é realizada para a discussão daquilo que foi desenvolvido naquele ciclo. Caberá ao Product Owner analisar se cada tarefa foi concluída conforme o esperado ou se alguma delas deverá retornar ao Product Backlog para a inclusão em uma nova sprint.

#### Sprint Retrospective

A reunião de retrospectiva da sprint é realizada após a reunião de revisão e anteriormente à reunião de planejamento, para que seja discutido um plano de melhorias. O objetivo dessa reunião é promover a colaboração entre os membros da equipe de desenvolvimento para corrigir possíveis desvios de rota e aprimorar aquilo que já está nos trilhos.

Por que as empresas utilizam o Scrum?

Em qualquer gerenciamento de projeto, os pilares buscados são a diminuição de custos, entregas mais rápidas e com maior qualidade. Em suma, as empresas que recorrem ao Scrum miram esses três objetivos numa única metodologia.

De forma geral, os projetos são desenvolvidos para a criação ou melhoria de um produto, pensando em necessidades de mercado que podem se transformar rapidamente, de acordo com o segmento de cada cliente. Com as entregas parciais a cada sprint, o Scrum permite que o projeto seja construído aos poucos e novas funcionalidades possam ser integradas.

Dessa forma, a metodologia permite que os produtos sejam lançados, testados e validados no mercado rapidamente, reduzindo o “time-to-market”.

Uma vez que o Product Owner priorizará atividades mais importantes para a realização do projeto, haverá a segurança de que os requisitos de maior valor para o negócio terão sua entrega privilegiada pelo time de desenvolvimento. Ou seja, aquilo que é primordial para o cliente receberá atenção especial, enquanto funcionalidades secundárias tendem a levar mais tempo para entrar em uma sprint. Por consequência, o Scrum agrega maior valor ao negócio e em menor tempo, o que diminui os custos de operação e potencializa o ROI desses projetos.

Da mesma forma em que são estabelecidas prioridades para o desenvolvimento no projeto, o Scrum abre espaço para atender à necessidade de mudanças no decorrer desse processo, na medida em que o desenvolvimento é realizado em curtas etapas, com entregas parciais.

Enquanto as metodologias tradicionais não proporcionavam a liberdade para que alterações fossem implementadas durante o projeto — era necessário aguardar a entrega final para testar suas aplicações —, os desenvolvedores estão mais capacitados a absorver a necessidade de mudanças com a adoção de metodologias ágeis e podem modificar o escopo do projeto.

Nesse sentido, há um claro aumento de competitividade das empresas que recorrem ao Scrum. Com as entregas parciais daquilo que é desenvolvido e os testes realizados ao fim de cada sprint, há a possibilidade de o projeto ser modificado para atender a novas demandas do cliente e as prioridades serem alteradas no backlog já no próximo ciclo de produção.

Metodologia Scrum e Agile têm se consolidado como alternativas para o desenvolvimento de sistemas com rapidez e eficiência, permitindo não apenas a redução de tempo, como também de gastos. Todavia, o gestor de projetos deverá conhecer diferentes técnicas, a fim de definir aquela que será a mais adequada para as necessidades de sua empresa. Nesse sentido, é possível aplicar Scrum e outras metodologias ágeis em soluções híbridas.

### **Desenvolvimento Ágil Com Scrum**

Considerando o cotidiano da maioria das organizações na atualidade, é inegável a importância desempenhada por sistemas computacionais dos mais variados tipos na condução de operações rotineiras. Esse suporte não apenas contribui para uma maior eficácia na execução de atividades do dia-a-dia, como também foi um fator determinante para um incremento substancial na produtividade de diversos processos de negócio.

A evolução da área de sistemas foi acompanhada pelo surgimento de diversas metodologias, com estas últimas normalmente englobando um conjunto de diretrizes e conceitos criados com o intuito de nortear o processo de construção de um software. Como de praxe, não há uma fórmula mágica para se chegar ao resultado final, ou seja, uma aplicação que atenda às expectativas dos usuários e seja capaz de funcionar dentro de uma série de parâmetros considerados como aceitáveis. Partindo de um conjunto de técnicas e diretrizes com eficácia já comprovada, os profissionais envolvidos empreendem esforços no sentido de adaptar um modelo para a realidade na qual se encontram inseridos.

É importante frisar sempre que desenvolver softwares não é uma tarefa das mais simples. Uma ampla gama de variáveis influencia no modo como uma aplicação será construída, somando-se ainda a isto influências como a pressão pela entrega do produto em um prazo muito curto, mudanças motivadas por alterações na legislação vigente, dificuldades dos usuários que solicitam um sistema em descrever de forma clara e concisa aquilo que realmente necessitam (com prováveis pedidos de modificações ao longo do projeto), dentre outros aspectos.

A noção de mudança também representa um elemento central na elaboração de aplicações. Por mais que todo um esforço procurando contemplar um número extenso de situações seja levado a cabo, é praticamente impossível construir um software que em determinado momento não precise passar por alterações. Aliás, é bastante comum que exista a necessidade de modificações durante a construção da aplicação, comprometendo assim uma parcela do tempo e de recursos que já haviam sido alocados para a implementação daquela solução.

Em um cenário de transformações constantes, a escolha pelo paradigma de desenvolvimento mais adequado a um determinado contexto é, sem sombra de dúvidas, um fator crucial para o sucesso do projeto. As primeiras técnicas formais voltadas à construção de sistemas possuíam um enfoque direcionado à elaboração de soluções com uma estrutura mais rígida e, portanto, eram menos suscetíveis a mudanças. A necessidade de uma rápida adaptação diante de situações imprevistas é um dos aspectos



que caracteriza o mundo corporativo atual; procurando atender a este tipo de demanda, surgiriam práticas mais flexíveis para a criação de aplicações, com diversas destas sendo conhecidas como “metodologias ágeis”.

O objetivo deste artigo é abordar, em linhas gerais, como métodos ágeis podem ser empregados em atividades relacionadas com o desenvolvimento de sistemas. Isto será feito, basicamente, através de um estudo mais detalhado de um conjunto de práticas com uma grande aceitação na área de software atualmente: trata-se da metodologia Scrum.

### **Metodologias De Desenvolvimento De Software**

O modelo em cascata (em inglês “waterfall”) foi, ainda na década de 1970, um dos primeiros padrões a fornecer diretrizes para processos voltados ao desenvolvimento de software. Esta metodologia é caracterizada por fases que ocorrem dentro de uma sequência rígida, com o início das atividades de uma etapa acontecendo imediatamente após o término daquela que a precedeu. A implementação de um projeto que segue este modelo é geralmente dividida nas seguintes fases: análise de requisitos, projeto da aplicação, implementação, testes de validação, implantação e manutenção.

De certa forma, esta abordagem é bastante semelhante à da linha de montagem clássica do mundo fabril. Considerando o cenário atual, em que muitas organizações se veem às voltas de transformações profundas e muitas vezes repentinas, a maneira linear que este modelo impõe à atuação dentro de um projeto de software revela-se como um fator deveras limitante.

Após uma fase inicial em que se apresentam as expectativas e em que se estabelece o escopo da aplicação a ser gerada, a área que solicitou um sistema apenas terá uma visão do resultado ao término do projeto. Mudanças eventuais em requisitos ou, mesmo, em regras que definem o comportamento destes podem prejudicar todo um esforço de meses. Tais alterações podem ainda comprometer um orçamento previamente acordado, além de não ser raro que o produto resultante sequer chegue a ser utilizado (por estar distante daquilo que se aguardava).

Conforme já frisado neste artigo, o cotidiano dinâmico de muitas organizações acabaria impondo o surgimento de um novo enfoque para a construção de softwares. Com base em uma série de parâmetros pré-estipulados, mas que também se encontram sujeitos a mudanças ao longo de um projeto, outras abordagens foram desenvolvidas de forma a tornar o desenvolvimento de aplicações mais flexível e produzindo resultados mais próximos do esperado.

Procurando fazer um contraponto às deficiências do modelo cascata, a metodologia RUP (Rational Unified Process) foi criada para permitir um desenvolvimento incremental e com entregas sucessivas de partes do software combinado. Amparando-se em conceitos da Orientação a Objetos, além de representação de estruturas de sistemas por meio da linguagem UML.

RUP é uma boa alternativa para projetos de grande porte que exigem um processo bem estruturado e que prime por uma documentação rica em detalhes. Esta é uma demanda bastante comum em organizações sujeitas a rigorosos procedimentos de auditoria. Este modelo está dividido em fases de análise (Concepção), modelagem e arquitetura do sistema a ser entregue (Elaboração), implementação (Construção) e Transição (implantação), sendo possível ainda que as diferentes atividades destas etapas possam vir a acontecer de maneira paralela em determinados momentos.

Ainda sobre o RUP, esta metodologia é na verdade uma implementação de um processo conhecido como Unified Process, representando uma solução específica da IBM para a condução de atividades voltadas ao desenvolvimento de software. É importante destacar também que existe uma série de ferramentas disponibilizadas para a geração dos diversos artefatos previstos por este modelo.

Nem sempre um processo trabalhoso e extremamente detalhado como o RUP pode ser a melhor alternativa em projetos de software. Equipes de tamanho reduzido, projetos que não são extensos e requisitos mudando constantemente motivariam a busca por novas formas de se controlar o processo de desenvolvimento de uma aplicação. Essas necessidades levariam, conseqüentemente, ao desenvolvimento de uma série de padrões que compõem um agrupamento de técnicas conhecidas como metodologias ágeis.

Em 2001 seria publicado, a partir do trabalho conjunto de diversos especialistas da área de sistemas, o Manifesto para Desenvolvimento Ágil de Software. Tal declaração enfatiza a entrega de software

operável, atuando em conjunto com o solicitante do mesmo, além de frisar a necessidade de uma interação adequada entre os diversos envolvidos num projeto e a flexibilidade diante de mudanças. São doze os princípios que norteiam o Manifesto Ágil (ou em inglês “Agile Manifesto”):

Satisfação das expectativas da área solicitante/cliente;

Uma postura mais positiva diante da necessidade de mudanças, procurando inclusive obter benefícios e explorar oportunidades não vislumbradas antes;

Entregas mais frequentes, a fim de possibilitar que o cliente possa acompanhar melhor a evolução do projeto, visando assegurar que os requisitos esperados realmente têm sido atendidos;

A colaboração entre todos aqueles que participam do projeto, quer sejam pessoas da área que solicitou a aplicação ou, mesmo, desenvolvedores focados em atividades de implementação, de forma a se agir como um time coeso;

Assegurar que existam condições para que todos possam trabalhar motivados;

Transmissão de informações através de conversas face-a-face;

Garantir que o software é entregue funcionando;

Manter um ritmo constante de trabalho, de forma que o processo como um todo continue de um modo sustentável, sem grandes percalços;

Forte ênfase na qualidade do que é produzido;

Foco na simplicidade;

Equipes auto-organizáveis, capazes de tomar decisões técnicas que viabilizem a evolução contínua do processo em que se encontram inseridas;

Uma reflexão constante dos rumos que estão sendo trilhados, a fim de se aumentar a eficácia da equipe, bem como ajustar o comportamento da mesma para com metas estabelecidas previamente.

Além do Scrum que será descrito em maiores detalhes nas próximas seções, outros modelos ágeis que desfrutaram de uma boa aceitação na área de desenvolvimento de software são as metodologias XP e Lean IT.

XP (abreviação do inglês “Extreme Programming”) é uma metodologia de desenvolvimento de software que se caracteriza por uma forte ênfase na elaboração e execução contínua de testes unitários, bem como pela codificação em pares: uma dupla de desenvolvedores participa da implementação de uma ou mais funcionalidades, sendo que isto acontece em torno de um único computador, com a pessoa responsável por escrever o código sendo auxiliada pelo parceiro que a observa e a orienta conforme a evolução das atividades. Esta abordagem em que dois profissionais interagem em conjunto procura diminuir a incidência de falhas, assim como possibilitar uma melhor qualidade do código resultante.

Lean IT é uma metodologia ágil baseada no modelo de gestão de produção da montadora japonesa Toyota. Este método prioriza a organização das atividades em uma maneira na qual se elimine ou reduza a perda de recursos (sobretudo tempo), de forma a tornar a aumentar a eficiência dos processos e atender mais rapidamente às necessidades das áreas que solicitaram um projeto.

Independentemente da opção escolhida para gerenciar projetos dentro da área de Tecnologia, é necessário sempre ter em mente que cada padrão pode ser adaptado ou, até mesmo, combinado a outra abordagem, de maneira a se adaptar melhor às necessidades da organização.

Métodos ágeis não estão restritos a uma tecnologia (.NET, Delphi, Java etc.); na verdade, as diversas técnicas existentes procuram ser a base para uma mudança de postura rumo a um melhor gerenciamento das atividades cotidianas.

Desenvolvimento ágil com Scrum

Scrum é uma metodologia ágil voltada ao desenvolvimento de software.

Surgido ainda na década de 1990, este modelo é resultado dos esforços conjuntos de especialistas da área de sistemas, com destaque especial para Ken Schwaber e Jeff Sutherland.

O termo “scrum” é originário do meio esportivo: no jogo de rugby esta palavra de língua inglesa refere-se ao reinício de uma partida logo após uma infração leve. Na Figura 1 é apresentada a representação esquemática de um processo baseado nesta metodologia, com os diferentes itens citados sendo descritos nas próximas seções.

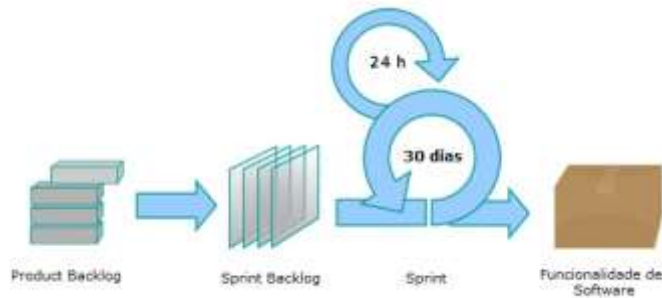


Figura 1: Visão geral dos prRepresentação lúdica do comprometimento com um projeto Scrum

Assim como outros métodos com um enfoque similar, a proposta do Scrum é fornecer subsídios para o gerenciamento de atividades muitas vezes complexas, porém de uma forma flexível e que facilite a adaptação do projeto diante das inevitáveis mudanças. São três as ideias principais em que a metodologia Scrum se ampara:

Transparência;

Inspeção;

Adaptação.

A noção de transparência deve ser compreendida como a existência de um consenso mútuo entre todos os participantes de um projeto, considerando para isto termos específicos de negócio, regras que caracterizam processos e outros aspectos. Inclusive o significado de "pronto", algo tão controverso em projetos de software que apresentam deficiências, está contemplado dentro do conceito de transparência.

Outro ponto importante da metodologia Scrum é a atividade de inspeção. A verificação contínua do que é produzido é fator determinante para que se tenha a certeza de que o projeto caminha dentro do estipulado, bem como para diagnosticar desvios indesejáveis e atuar de forma corretiva sobre estes últimos.

Já o conceito de adaptação vai de encontro a um dos principais objetivos dos métodos ágeis: a flexibilidade diante de mudanças. Não serão raras as ocasiões em que alterações em demandas de clientes ou ainda, regulamentações governamentais e de ordem setorial, acarretarão desvios de rotas pré-estabelecidas. Diante disto, ajustes se fazem necessários, atenuando desse modo outros efeitos indesejáveis em momentos posteriores.

Os diferentes papéis de atuação na metodologia Scrum

Um conjunto de papéis bem definidos determina, através de atribuições, como será a atuação dos diversos profissionais que participarão de um projeto baseado na metodologia Scrum:

Product Owner: representante da área cliente/solicitante;

Scrum Master: o líder que gerenciará o projeto;

Equipe de Desenvolvimento/Time: os profissionais responsáveis pela criação do produto esperado.



O Product Owner é o ponto que centraliza a interação com a área/grupo de usuários que solicitou a execução de um projeto. A partir do mesmo são definidas prioridades, o que deverá ou não ser implementado e a validação dos diferentes resultados ao longo do processo de desenvolvimento.

Já o Scrum Master corresponde ao papel do Gerente de Projetos tradicional. Além de ser um facilitador removendo impedimentos e um mediador em prováveis conflitos, este profissional garante que a equipe sob sua supervisão siga de maneira adequada as práticas de Scrum.

Por fim a Equipe de Desenvolvimento (ou Time) elabora estimativas, estipula tarefas, implementa o produto dentro de níveis de qualidade pré-estipulados e cuida da apresentação do mesmo ao cliente/solicitante. Conforme já mencionado anteriormente, espera-se que tal equipe conte com um caráter auto-gerenciável, com o comprometimento e uma postura multifuncional dos membros representando um fator crucial para o sucesso do projeto.

Além dos três papéis já descritos, certamente também existirão envolvidos com o projeto, mas que não desempenham um papel direto na sua execução. Estes elementos podem englobar usuários, gerentes, diretores ou departamentos que possuem interesses (neste caso são conhecidos dentro da Gerência de Projetos como “stakeholders”) ou ainda, são afetados pelos resultados do produto final.

Considerando tudo isto, criou-se uma forma lúdica de representar o grau de comprometimento de uma pessoa em um projeto que segue as práticas do Scrum; para isto, foi utilizada uma sequência de histórias em que interagem um porco e um frango (Figura 2). Porcos representariam profissionais realmente engajados/comprometidos com o sucesso do projeto (Product Owner, Scrum Master, Time), ao passo que frangos seriam pessoas relacionadas indiretamente e sem uma maior disposição para com o mesmo (usuários comuns, gerentes ou áreas afetadas).



Figura 2: Representação lúdica do comprometimento com um projeto Scrum

**OBSERVAÇÃO:** No site “Implementing Scrum” (endereço indicado na seção de Links deste artigo) podem ser encontradas outras tiras que descrevem de forma bem humorada a metodologia Scrum, assim como problemas do dia-a-dia e os esforços na condução de projetos sob este padrão.

#### Eventos possíveis em Scrum

Como outros métodos ágeis, Scrum é uma metodologia que prima pelo desenvolvimento iterativo e incremental de software. Em termos práticos, isto significa que ciclos contendo um conjunto de específico de atividades são repetidos continuamente ao longo de um projeto; por incremental, deve-se ter em mente a idéia de sucessivas entregas de funcionalidade, acrescentando aquilo que se espera do software em intervalos constantes de tempo.

Antes de prosseguir com a descrição dos diferentes eventos de Scrum, faz-se necessário conceituar as idéias de Time-Box e Sprint. Uma Time-Box nada mais é do que a quantidade de tempo estipulado para a realização de uma iteração. Esta última recebe o nome de Sprint, sendo que essa estimativa de tempo não pode ser alterada, a fim de com isto garantir a entrega sem atrasos e facilitar o planejamento. Diante da possibilidade de erros na estimativa, deve-se proceder com uma redução do escopo, sem contudo afetar substancialmente as metas da Sprint ou obrigar a um aumento na quantidade de horas ou dias planejados. O comum é que uma Sprint dure de duas a quatro semanas.

A partir destas explanações, são possíveis dentro de Scrum os seguintes eventos:

Reunião de Planejamento da Sprint;

Reunião Diária;

Revisão da Sprint;

Retrospectiva da Sprint.

A Reunião de Planejamento da Sprint é uma atividade com duração de 8 horas e da qual participam todos os profissionais comprometidos com o projeto (Product Owner, Scrum Master, Equipe de Desenvolvimento). Esta reunião é formada por duas etapas: num primeiro momento o Product Owner define a prioridade de funcionalidades a serem implementadas (a partir do Backlog); posteriormente, a Equipe montará sua própria lista de trabalho (Sprint Backlog) com base no que foi exposto pelo Product Owner.

Já a Reunião Diária é uma curta sessão de 15 minutos, em que membros da Equipe e o Scrum Master comentam a situação atual (muitas vezes em pé dentro de um recinto). Isto normalmente envolve uma rápida explanação do que foi feito no dia anterior, o que será realizado na data atual e uma discussão de prováveis impedimentos que foram encontrados.

A Revisão da Sprint é uma reunião de normalmente 4 horas, em que estão presentes o Product Owner, o Scrum Master e a Equipe (pode acontecer ainda de outras pessoas serem convidadas para participar). As atividades desempenhadas durante a Sprint são apresentadas, procedendo-se com a entrega do software funcionando (conforme pregado pela metodologia).

Por fim, há ainda a Retrospectiva da Sprint. Trata-se de uma reunião de geralmente 3 horas entre Equipe e Scrum Master. Nesta discussão aborda-se o que deu certo e aquilo que falhou, além de se estudarem formas para se melhorar num próximo ciclo (Sprint).

Artefatos de Scrum

Por mais que como uma metodologia ágil Scrum priorize a entrega do software em detrimento de uma extensa e trabalhosa documentação, a elaboração e a consequente manipulação de alguns artefatos neste modelo é de fundamental importância para o controle das atividades rotineiras. A seguir estão listados tais documentos:

Backlog do Produto (ou em inglês “Product Backlog”);

User Story;

Sprint Backlog;

Gráfico de Burn Down.

O Product Backlog é uma listagem que contempla todas as funcionalidades desejadas para o software que se está implementando. Além disso, as informações contidas neste tipo de controle podem conter ainda uma ordem de prioridade, sendo incumbência do Product Owner criar e controlar o status dos diferentes elementos do Backlog.

Uma User History é uma pequena história que descreve as características esperadas para uma funcionalidade constante no Backlog do Produto. Constam no documento que representa tal história um título, uma descrição clara do que se necessita, bem como é possível se indicar ainda uma prioridade para execução da tarefa.

A Sprint Backlog é uma relação de tarefas elaborada pelo Time de Desenvolvimento durante a segunda etapa da Reunião de Planejamento da Sprint. Trata-se de algo que está em conformidade com o conceito de equipes auto-gerenciáveis, uma vez que os profissionais responsáveis por isto planejam como será o dia-a-dia de desenvolvimento a partir das prioridades apontadas pelo Product Owner.

O Gráfico de Burn Down é uma ferramenta de gerenciamento. Este artefato costuma ser atualizado diariamente, servindo de base para a comparação entre o que foi planejado e aquilo que realmente se realizou. Pode ser considerado um instrumento para tomada de decisão, uma vez que fornece embasamento para ações em prol de uma maior produtividade ou ainda, a fim de atenuar riscos iminentes.

