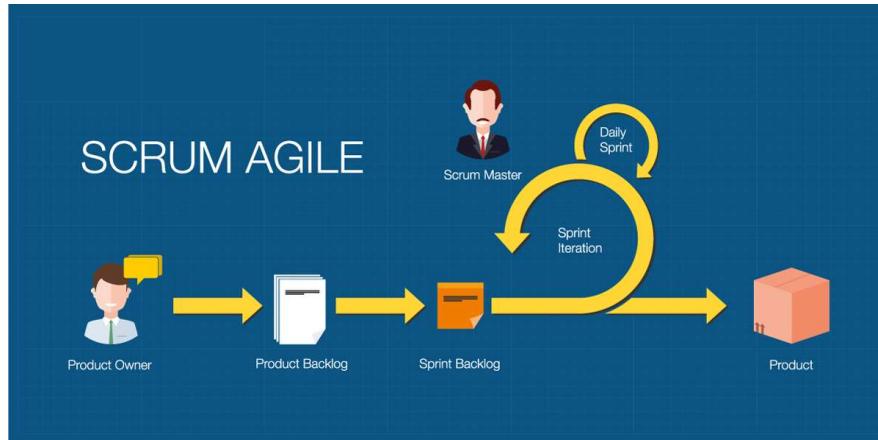


NÃO PODE FALTAR

## DESENVOLVIMENTO DE MÉTODOS ÁGEIS UTILIZANDO SCRUM

Izabelly Soares de Morais



Fonte: Shutterstock.

### Deseja ouvir este material?

Áudio disponível no material digital.

### PRATICAR PARA APRENDER

Prezado aluno, é bem provável que, em seu dia a dia, você execute atividades diversas, que vão desde atividades pessoais até profissionais. Outro fator provável é que você deve fazer uso de softwares para exercer algumas dessas atividades, correto? Você já reparou que, a cada dia que passa, alguns serviços básicos, como o e-mail, acabam recebendo novas funcionalidades? Para sua melhor compreensão, antes, o serviço de e-mail era apenas o de receber e enviar mensagens de texto, porém, devido às novas necessidades dos usuários, ao surgimento de novos tipos de dados, de arquivos, entre outros recursos, as empresas que disponibilizam esses serviços adicionaram envio de fotos, vídeos, formatação de texto do e-mail, inclusão de assinatura etc. Se formos citar todas as funcionalidades, passaremos um bom tempo nessa conversa.

Antigamente, o processo de desenvolvimento de software era bem mais burocrático e lento em comparação ao que temos hoje, mas com o decorrer do tempo, as metodologias evoluíram. Atualmente, temos uma diversidade bem interessante, que adota os princípios da metodologia ágil, como o *Scrum*.

Devido à alta demanda por softwares cada vez mais eficazes, é importante que você conheça as particularidades do *Scrum*, que carrega consigo alguns pilares que podem auxiliar nesta nova necessidade social, que é a de tornar o ciclo do software algo mais rápido, porém garantindo a qualidade do software final e de todo seu processo de desenvolvimento.

E para proporcionar o que foi dito, o *Scrum* determina atividades específicas para componentes específicos do projeto, como *Scrum Master*, *Product Owner* (dono do produto) e *Time Scrum*, além disso, existem as cerimônias, como sprint, que representa o ciclo iterativo, a reunião diária e o *backlog*, que é a lista das

funcionalidades que devem ser priorizadas conforme a necessidade do cliente e que passa a fazer parte da *Sprint*. Essa seleção facilita a decisão sob a aprovação ou não do que foi produzido, dessa forma, todas essas ações são sempre voltadas à qualidade do sistema e ressaltam a importância da agilidade neste momento.

Uma rede de *fast food* está no mercado alimentício há mais de 10 anos. Ela foi fundada no Japão e, posteriormente, expandiu-se para todo o mundo, abrindo várias filiais até mesmo no Brasil. No ano de 2020, os executivos da empresa decidiram abrir um escritório em São Paulo e você foi contratado pelo setor de tecnologia para fazer parte da equipe.

Na empresa, são desenvolvidos diversos aplicativos voltados às ações do e-commerce, que atua mundialmente, fazendo com que suas aplicações atendam ao seu público-alvo, independentemente da sua localização.

Seu desafio é implantar o *Scrum*, assumindo o papel de *Scrum Master*, sendo uma das suas principais funções garantir que o *Scrum* seja aplicado de forma correta. Dessa forma, terá que expor a sua equipe o papel de cada um dentro dela e seus respectivos cargos, ou seja, terá que definir o dono do produto, já que você atuará como *Scrum Master*. Aproveite e deixe claro suas funções dentro das cerimônias e diante dos artefatos gerados, pois você poderá trabalhar desenvolvendo aplicações para vários setores da empresa. Além disso, escolha uma ferramenta para acompanhamento do projeto. Como exemplo, você poderá utilizar o Azure DevOps.

Como artefato gerado por essa atividade, deverá ser desenvolvido um relatório. Posteriormente, compartilhe seu relatório na nuvem (*Google Drive*, *One drive*, *dropbox*, entre outras opções). O objetivo é que os demais colegas possam visualizar suas soluções para a situação apresentada e contribuir com suas ideias para melhorar sua solução. Além disso, você também pode elaborar uma apresentação, simulando uma situação real de apresentação de projeto aos executivos da empresa.

Após ter sido apresentada a situação-problema, reforce seus conhecimentos sobre cada papel dentro do escopo da metodologia Scrum e reflita sobre a melhor forma de solucioná-la.

Agora, aproveite este momento fantástico para conhecer as particularidades do *Scrum* e imaginar como você aplicaria esses conhecimentos para desenvolver softwares!

## CONCEITO-CHAVE

### METODOLOGIAS DE DESENVOLVIMENTO DE SOFTWARE

As primeiras metodologias de desenvolvimento de software surgiram em meados dos anos 1970. Naquela época, era difícil imaginar que seria possível utilizar um software nos mais variados tipos de dispositivos para desempenhar a diversidade de ações disponíveis na atualidade, uma vez que o seu uso era mais restrito às universidades e aos setores governamentais, já que, na época, os custos para se adquirir o próprio hardware era bem elevado.

A engenharia de software é uma área que aborda o projeto de software, já que expõe exemplos e métricas voltados ao seu ciclo de vida. Outro ponto interessante e até reflexivo dentro desse mundo do desenvolvimento de software é que existe a gestão de projeto com uma perspectiva mais geral e existe uma gestão ou projeto de software em si, ou seja, em que todo o ciclo é visto como foco. Por fim, as duas perspectivas de gestão se complementam. Trata-se de um conjunto de ações multidisciplinares que contribui para trazer a qualidade ao software e ao seu projeto.

Ainda sob esse ponto de vista, é importante mencionarmos que existem algumas atividades e subatividades consideradas básicas para todo projeto de software; entre elas, podemos destacar:

- **Especificação de software:** momento em que ocorre o processo inicial de comunicação entre os *stakeholders* e o início do planejamento de todo o projeto.
- **Projeto e a implementação de software:** nessa fase, o planejamento realizado anteriormente é avaliado e organizado para poder ser executado e para que a fase de implementação seja inicializada.
- **Validação de software:** é uma etapa em que podem ocorrer testes ou outras formas de se confirmar se tudo o que foi planejado foi implementado de forma correta, bem como se atende às reais demandas do cliente.
- **Evolução do software:** é a capacidade que o software tem de ser atualizado e apto a atender às possíveis mudanças que podem ocorrer ou com as regras de negócio do cliente ou mesmo com novas funcionalidades.

## | METODOLOGIA ÁGIL

Portanto, seguindo o fluxo natural, em que as atividades cotidianas passam por atualizações, as metodologias tradicionais, trazendo essas atividades citadas anteriormente, também evoluem, além disso, trata-se de um processo contínuo.

Mas é possível visualizar outra categoria de processo de software, que é a metodologia ágil; no entanto, essa metodologia, na realidade, traz uma vivência da filosofia ágil que foi sendo implantada no universo do desenvolvimento de software desde a década de 1990, após a criação do “Manifesto Ágil”, em que vários estudiosos e especialistas se reuniram para trazer atualizações e novas formas de desenvolver sistemas.

Conforme afirmação de Ferreira (2020, p. 28), o Manifesto Ágil “é uma declaração de valores e princípios que fundamentam como essa atividade pode trazer benefícios em ganhos de produtividade e, em muitos casos, diminuição de custos”. Com isso, foram apresentados alguns princípios, os quais foram listados no Quadro 2.3.

Quadro 2.3 | Princípios da filosofia ágil

Princípios	Descrição
Envolvimento do cliente	Os clientes devem estar intimamente envolvidos no processo de desenvolvimento. Seu papel é fornecer e priorizar novos requisitos do sistema e avaliar suas interações.
Entrega incremental	O software é desenvolvido em incrementos com o cliente, especificando os requisitos para serem incluídos em cada um.
Pessoas, não processos	As habilidades da equipe de desenvolvimento devem ser reconhecidas e exploradas. Membros da equipe devem desenvolver suas próprias maneiras de trabalhar, sem processos prescritivos.
Aceitar as mudanças	Deve-se ter em mente que os requisitos do sistema vão mudar. Por isso, projete o sistema de maneira a acomodar essas mudanças.
Manter a simplicidade	Focalize a simplicidade, tanto do software a ser desenvolvido quanto do processo de desenvolvimento. Sempre que possível, trabalhe ativamente para eliminar a complexidade do sistema.

Fonte: Sommerville (2011, p. 40).

Por meio de seus princípios, ficou claro que o uso do termo “ágil” não se limitava à rapidez no desenvolvimento, mas também à entrega de um produto com qualidade, em um espaço de tempo menor do que antes e com a aproximação do cliente na solução dos seus problemas. Fique atento ao uso do termo problema, que se refere à situação que o sistema terá que solucionar ou aos processos que serão automatizados após a finalização do desenvolvimento do sistema.

#### REFLITA

Tendo como base o cenário social da atualidade, em que, devido à globalização, é possível ter acesso a tecnologias utilizadas em todo o mundo, assim como realizar conexões intercontinentais por meio da internet, a demanda pelo desenvolvimento de software de forma eficiente e eficaz faz com que novas metodologias surjam a cada dia. Mas é interessante pensarmos que metodologias vistas como tradicionais também podem ser adaptadas com recursos ágeis para serem aplicadas. Você já imaginou como seria a experiência de desenvolvimento por meio das práticas defendidas pelo Manifesto Ágil?

Uma das primeiras metodologias ágeis criadas foi a **XP (Programação Extrema - Extreme programming)**, em meados da década de 1990; posteriormente, vieram muitas outras, tendo em vista que as empresas começaram a utilizá-las e a relatar

os benefícios trazidos a todo o processo, com economia de recursos e mais qualidade ao sistema (PRESSMAN, 2016).

Dentro desse espaço de tempo, surgiu a metodologia ágil **Scrum**, que recebe esse nome devido ao esporte Rugby, fazendo uma analogia entre o esporte e a metodologia. A união das pessoas para atingirem um objetivo acaba sendo a maior razão do uso desse termo.

Ken Schwaber e Jeff Sutherland foram os principais percursoros do conceito *Scrum* e sua aplicabilidade para o desenvolvimento de software; eles apresentaram o termo durante a conferência *Object-Oriented Programming, Systems, Languages & Applications (OOPSLA)*, em 1995, em Austin, Texas. Após essa apresentação, vários especialistas se dedicaram a refinar o conceito e a metodologia do *Scrum* (SCRUMstudy, 2016).

As vantagens de uso dessa metodologia é, a princípio, bem interessante para serem observadas quando aplicadas em projetos de software; entre elas, podemos destacar a capacidade de adaptação, transparéncia, feedback contínuo e, consequentemente, a melhoria contínua, a eficácia nas entregas etc. Mas como tudo isso é possível? Você acha que por ter essas características, seu uso se torna mais fácil para a empresa que deseja adotá-la como principal processo de desenvolvimento?

A transição para o Scrum não pode ser um processo que “articule e defina todo o processo de mudança necessário para eliminar a lacuna entre ‘como é’ e ‘como ficará’ e crie planos táticos” (CARR; HARD; TRAHANT, 1996 *apud* COHN, 2011, p. 29).

#### ■ REGRAS E INDICAÇÕES DOS PRINCÍPIOS E PRÁTICAS SCRUM

A literatura define algumas regras ou indicações que podem ou não ser seguidas em relação aos princípios e às práticas do *Scrum*, porém lembre-se de que as empresas, na prática, adotam tal metodologia, mas possuem a liberdade de adaptá-la da melhor maneira possível ao seu contexto. Podemos destacar algumas indicações:

- As equipes podem ser chamadas de "Times Scrum". Conforme Cruz (2020, p. 41-42), “o Time Scrum é composto por três papéis: Scrum Master, Product Owner e Time. [...] O Time deve ser interdisciplinar, e geralmente quando é utilizado só esse termo, estamos nos referindo aos desenvolvedores”.
- O indicado é que tenham entre seis a dez membros (isso não limita o uso da metodologia apenas a projetos simples).

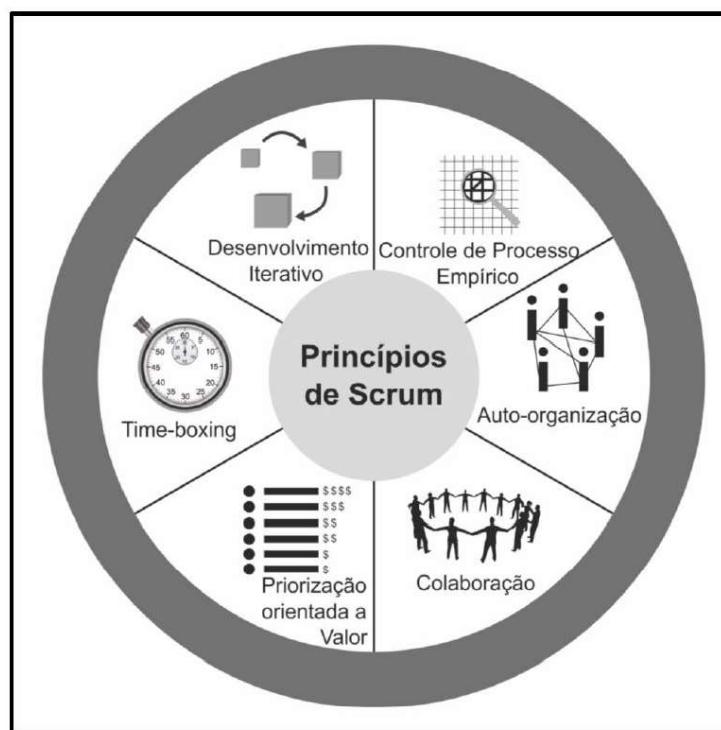
Assim como a filosofia ágil possui seus valores, o *Scrum* possui alguns princípios básicos, como:

- Desenvolvimento iterativo: sendo representado pela *sprint*, em que cada ciclo será composto por diversas particularidades.

o

Ver anotações

- Controle de processo empírico: possibilidade da realização de melhorias por meio da observação e das lições aprendidas em cada iteração.
- Auto-organização: devido à definição de papéis e cerimônias, os passos são muito bem definidos e cada um sabe seu papel dentro do time.
- Colaboração: essa característica é bem explícita e traz a responsabilidade dos resultados a todos.
- Priorização orientada a valor: o valor que o sistema pode trazer ao negócio do cliente é uma das prioridades.
- Time-boxing: refere-se ao tempo determinado para algumas cerimônias, como indicação de sprints entre 2 e 6 semanas, duração de 15 minutos nas reuniões diárias, entre outros.

o  
Ver anotaçõesFigura 2.4 | Princípios do *Scrum*

Fonte: SCRUMstudy (2016, p. 8).

#### I PAPÉIS DOS MEMBROS DA EQUIPE NO SCRUM

Para executar seus princípios, a metodologia estabelece papéis para serem atribuídos aos membros da equipe. Entre eles, podemos destacar os papéis centrais (**dono do produto, scrum master e time scrum**) e os papéis considerados não essenciais (**stakeholders, scrum guidance body, fornecedores, dono do produto chefe e scrum master chefe**).

Os papéis centrais são considerados de extrema relevância, então, conforme as diretrizes do *Scrum*, ao receberem essas nomenclaturas, os componentes da equipe devem assumir suas respectivas funcionalidades. Para isso, é interessante conhecer um pouco mais sobre cada um deles.

Quadro 2.3 | Papéis centrais do *Scrum*

Papéis centrais	Descrição
<b>Dono do produto (ou <i>product owner</i>):</b>	Tem como função trazer uma perspectiva sob o ponto de vista do cliente em relação aos limites e às restrições que o cliente quer para aquele produto que será desenvolvido. Ele deve possuir argumentos suficientes para justificar a importância do projeto, sendo o representante do cliente dentro da equipe.
<b>Scrum Master:</b>	Responsável por propiciar à equipe, aqui chamada de Time <i>Scrum</i> , um ambiente organizado e alinhado aos valores que o projeto deve atingir. Também é de sua responsabilidade verificar se os conceitos <i>Scrum</i> estão sendo aplicados de maneira conceitualmente correta.
<b>Time Scrum:</b>	É a equipe responsável pela execução do projeto.

Fonte: elaborado pela autora (2020).

Por mais que alguns papéis se assemelhem aos de um projeto com metodologia tradicional, o *Scrum Master*, por exemplo, não deve ser associado ao gerente de projetos, tendo em vista que, conforme os princípios do Manifesto Ágil, o time deve estar motivado a trabalhar em determinado projeto. Frente a isso, por mais que se tenha esse papel, em que uma das principais ações é a de organizar o projeto como um todo, no geral, vendo pela perspectiva do Manifesto Ágil, ele é apenas mais um componente da equipe. Em resumo, Cruz (2013, p. 41) diz que a função do *Scrum Master* "deve garantir que o Time *Scrum* siga o Fluxo *Scrum* e deve também remover todos e quaisquer impedimentos que possam interferir no objetivo do Time *Scrum*".

Entre os papéis considerados não essenciais, é importante conceituar que o termo **stakeholder** é um termo utilizado para referenciar os usuários e clientes do projeto. Em algumas situações, e principalmente em outras metodologias, o time de desenvolvimento também pode se encaixar nesse termo. O **Scrum Guidance Body (SGB)** é opcional e se refere a profissionais especialistas em orientar os demais profissionais do time quanto às regulamentações e a outros detalhes.

Já os **fornecedores** são aqueles indivíduos ou empresas que fornecem algum produto ou serviço para ser utilizado no projeto. Como exemplo, é possível citar uma situação em que a empresa precisa de serviços na área de segurança da informação, porém não contempla suas expertises, dessa forma, a empresa contrata outra empresa especializada no assunto. Os demais papéis não essenciais (dono do produto, chefe e *scrum master* chefe) contemplam as mesmas funcionalidades mencionadas no quadro, porém podem ser mais utilizados em projetos maiores.

## ■ FASES SCRUM

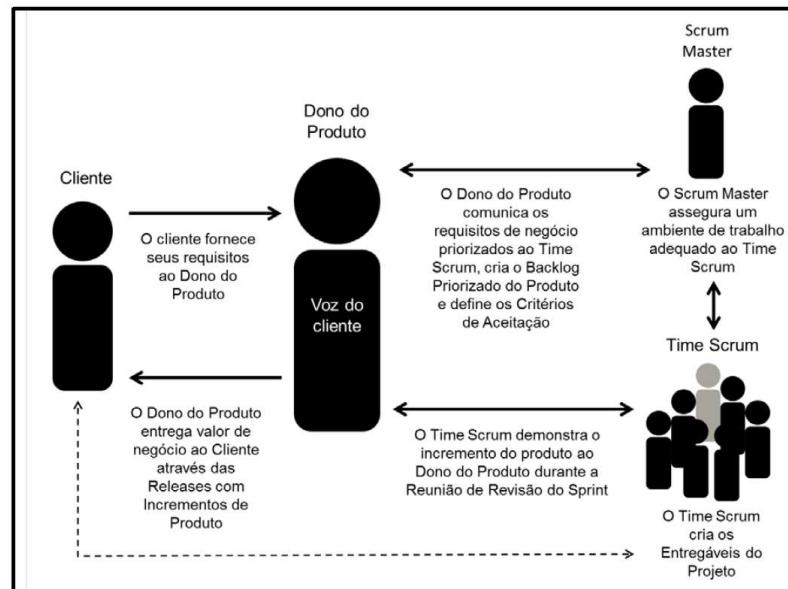
Além disso, o Scrum possui 5 fases: **início** (onde haverá a criação da visão do projeto; a identificação do *Scrum Master*, dos *stakeholders* e do time *Scrum*; o desenvolvimento de épicos; a criação de *backlog* priorizado do produto; e o planejamento da release.), **planejamento e estimativa** (em que haverá a criação e a análise de histórias do usuário; a criação e a estimativa para as tarefas e a criação do *backlog* do *sprint*), **implementação** (em que os entregáveis do projeto são criados, as reuniões diárias são conduzidas e acontece o refinamento do *backlog* priorizado do produto), **revisão e retrospectiva** (em que ocorrem demonstrações e validações do *sprint*, bem como sua retrospectiva) e, por fim, o **release** (em que os entregáveis são disponibilizados aos clientes e ocorre a retrospectiva de todo o projeto).

o Ver anotações

### ASSIMILE

A fase de planejamento é importante para todo tipo de projeto. No contexto das metodologias ágeis, é possível notar algumas práticas para facilitar essa etapa tão importante. Entre elas, é possível destacar a da XP, em que os requisitos dos usuários são listados em cartões e chamados de "histórias do usuário". Isso ocorre para ajudar a equipe a mensurar a prioridade e o tempo estimado para execução das atividades. No *Scrum*, essa lista de requisitos se chama *backlog*, porém as funcionalidades são separadas por um ciclo, que, conforme a metodologia, chama-se *sprint*.

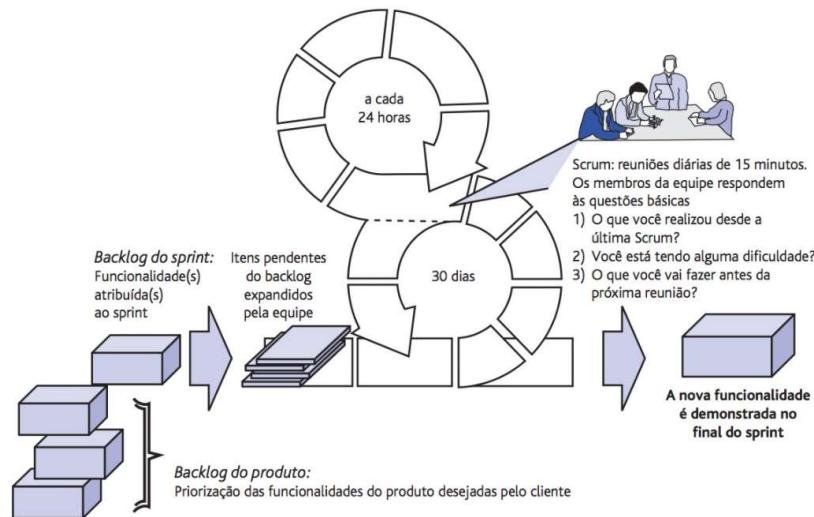
Figura 2.5 | Organização do *Scrum*



Fonte: SCRUMstudy (2016, p. 13).

O fluxo mostrado na figura 2.5, sobre a organização do Scrum, será fundamentado por meio do fluxo de processos da metodologia, mostrado na figura 2.6.

Figura 2.6 | Fluxo do processo *Scrum*



Ver anotações

Fonte: Pressman et al. (2016, p. 78).

Por meio da conexão entre as Figuras 2.5 e 2.6 é possível observar que tudo tem início quando o cliente consegue expor tudo aquilo que deseja em seu software; mas o desafio está justamente em conseguir fazer isso apenas no começo do projeto. É nesse cenário que conseguimos visualizar a vantagem de uma metodologia ser iterativa, ou seja, dividida em ciclos (que são chamados, no *Scrum*, de *sprints*), e incremental, em que, a cada final de ciclo, algum incremento ou funcionalidade do software é entregue ao cliente. Logo, a capacidade de receber decisões de mudanças por parte do cliente acaba sendo facilitada por essas ações.

A transparência permite que todos os ângulos de qualquer processo *Scrum* sejam observados por qualquer pessoa. Isso promove um fluxo de informação fácil e transparente em toda a organização e cria uma cultura de trabalho aberta. Em *Scrum*, a transparência é representada por meio das etapas mencionadas sobre os processos da metodologia (SCRUMstudy, 2016, p. 21).

Na figura 2.6 é possível observar quais atividades compõem o ciclo, ou seja, a *sprint* do *Scrum*. Para uma melhor compreensão, apresentaremos o significado de cada termo:

- Realizações de reuniões iniciais que definem as prioridades do cliente. Essas reuniões são chamadas de **"Reunião do Stakeholder"**.
- **Backlog:** refere-se à lista de requisitos que todo o processo deve atender. Ainda é possível encontrar uma subatividade do *backlog*, que pode ser chamado de "**backlog do produto**" (ou *backlog* priorizado do produto), que é composto pelos requisitos mais importantes daquele produto. Conforme o SCRUMstudy (2016), esses recursos são documentados por meio das histórias do usuário.

#### ATENÇÃO

As histórias do usuário possuem critérios de aceitação baseados nos objetivos que foram estabelecidos para as funcionalidades do projeto. Note que, entre os papéis que podem ser desempenhados dentro do *Scrum*, o

dono do produto é o componente com mais capacidade de estabelecer esses critérios, tendo em vista seu conhecimento nos negócios do cliente. Então, ao final da *sprint*, ele aceita ou rejeita as entregas conforme os critérios de aceitação, e após essa cerimônia, a história do usuário pode ou não ser considerada pronta. Caso as entregas sejam rejeitadas, voltam ao *Backlog* do produto priorizado e atualizado, e isso ocorrerá durante o processo de **refinamento do backlog priorizado**.

• Ver anotações

- Existe também o "**backlog da sprint**". Esse *backlog* é uma lista com as prioridades para aquela *sprint*.
- **Revisão da sprint.** "A reunião de Revisão da Sprint, que vem do inglês Sprint Review , é também conhecida como Apresentação da Sprint , cujo objetivo maior é a revisão do Product Owner, ou do cliente, em todos os itens concluídos pelo Time" (CRUZ, 2013, p.44). Ainda sob o ponto de vista do autor, é indicado que essa cerimônia, por ser um evento *time-boxed*, tenha a durabilidade de quatro horas, para que seja possível conferir e avaliar o que está sendo considerado pronto, levando-se em conta o que está sendo entregue versus o que deveria ser entregue.
- **Retrospectiva da Sprint.** como o próprio nome já sugere, haverá uma análise do que foi realizado durante a última *sprint*. Nessa cerimônia, tudo será levado em consideração, desde o time aos processos e até as ferramentas que foram utilizadas. Conforme Cruz (2013, p. 44), trata-se de uma cerimônia que deve durar, em média, umas três horas. O autor elenca alguns itens que devem ser priorizados, como os itens que deram certo e os que precisam de melhorias. Além disso, o objetivo é refinar os processos para a próxima *sprint*.

#### EXEMPLIFICANDO

Na prática, uma das cerimônias mais adotadas do *Scrum* é a reunião diária. Muitas vezes, a durabilidade da reunião acaba sendo adaptada à necessidade da equipe, mas é uma excelente oportunidade para todos saberem em que status o projeto se encontra no geral, já que todos dirão o que fizeram, o que vão fazer e se tiveram alguma dificuldade nas atividades do dia. Caso as equipes ultrapassem dez membros, pode ser mais adequado dividi-la em grupos (ou times *scrum*) menores.

#### FERRAMENTAS DE FLUXO

- **Azure DevOps**

Figura 2.7 | Azure DevOps

Ver anotações

Fonte: captura de tela do Azure DevOps.

- Trata-se de uma ferramenta desenvolvida pela Microsoft e pode ser acessada gratuitamente; além disso, é possível estabelecer os processos adequados para o projeto, criar histórias do usuário (*stories*), funcionalidades (*features*), épicos (*epics*) e definir bugs, *tasks*. É possível, também, adicionar o time e criar *backlog* e sprint para facilitar o acompanhamento do desenvolvimento do sistema. Porém, para projetos mais complexos, a empresa pode cobrar, tendo em vista que é uma ferramenta que disponibiliza outros serviços, como integração com repositórios, serviços em nuvem, entre outros.

#### • Asana

Figura 2.8 | Asana

Fonte: captura de tela do site asana.

- É um recurso que pode ser utilizado de forma gratuita e dá ao usuário a liberdade de escolher como ele deseja organizar seu projeto, seja por meio de planilhas, seja de documentos ou plataformas de slides ou criando cartões (*cards*) e quadros que podem assumir uma interface Kanban ou Agile. Outras opções são os gráficos de Gantt ou cronogramas visuais, além de anotações feitas em papel ou listas de tarefas.

O **Kanban** também é uma ferramenta utilizada, independentemente do Asana, e foi desenvolvido pela Toyota, na década de 1960. O significado do termo é “cartão”, em japonês, e a ideia é trazer um recurso visual para deixar transparentes os procedimentos do projeto, também chamado de “quadro kanban” (MORGAN *et al.*, 2016).

Figura 2.9 | Kanban



Fonte: Shutterstock.

Ver anotações

O quadro pode assumir a configuração que for mais adequada ao projeto em desenvolvimento e, normalmente, apresenta três campos: o primeiro é o que deve ser feito, em seguida, o que está sendo feito, por fim, o que já foi finalizado. Como dito anteriormente, o objetivo é trazer a demonstração visual para que todos da equipe consigam acompanhar o progresso das atividades.

A escolha da metodologia sempre vai depender do escopo do projeto que será desenvolvido, tendo em vista que é uma decisão que envolve não apenas as questões tecnológicas, em si, mas também a cultura organizacional das empresas. Além disso, ela influenciará em vários aspectos no resultado final.

Após esses conceitos aprendidos, é possível compreender que as cerimônias e os artefatos gerados pelo *Scrum* são o resultado de muitos anos de experiências na área do desenvolvimento de software e podem contribuir de forma positiva para todo o processo.

## FAÇA VALER A PENA

### Questão 1

A metodologia ágil *Scrum* recebeu esse nome devido ao esporte Rugby. O objetivo é que todos os envolvidos trabalhem juntos para atingir um mesmo objetivo, o que demonstra, claramente, o foco de uma metodologia ágil ao ser aplicada no ciclo de vida de um software. Nessa metodologia, existem papéis que são assumidos dentro da equipe.

Assinale a alternativa que descreve o papel do *Scrum* responsável por criar a justificativa de negócio para o desenvolvimento do sistema:

a. *Scrum master.*

b. *Product owner.*

c. *Stakeholder.*

d. *Scrum guidance body.*

e. *Time scrum.*

**Questão 2**

Dentro de um projeto de software existem muitos profissionais envolvidos, os quais, na maioria das vezes, são generalizados por meio do termo "stakeholder", mas na metodologia ágil *Scrum*, alguns perfis recebem termos específicos.

Então, o \_\_\_\_\_ representa, dentro de um projeto *Scrum*, as necessidades do cliente. Já o \_\_\_\_\_ tem a função de verificar se os conceitos Scrum estão sendo aplicados de maneira conceitualmente correta.

Assinale a alternativa que completa as lacunas corretamente.

a. dono do produto (product owner) e scrum master.

b. time scrum e dono do produto.

c. cliente e dono do produto.

d. dono do produto e time scrum.

e. scrum master e time scrum.

**Questão 3**

As mudanças sociais estão acontecendo cada vez mais rápido devido à globalização. Com isso, a facilidade na disseminação das informações acelera, também, a criação de novos produtos e serviços, tornando obsoletos os métodos considerados tradicionais para o desenvolvimento desses artefatos. O *Scrum* fornece oportunidades para o desenvolvimento de artefatos e produtos por meio da prática da metodologia ágil e das suas cerimônias e definições. Entre elas, podemos apontar 5 fases: início, planejamento e estímulo, implementação, revisão e retrospectiva e, por fim, release.

Tomando como base as 5 fases no *Scrum*, avalie as informações a seguir.

I. Início é a fase reservada para a criação e análise de histórias do usuário e estimativa para as tarefas do *backlog* da *Sprint*.

II. Planejamento e estímulo é a fase reservada para desenvolvimento de épicos, criação de *backlog* priorizado do produto e planejamento da *release*.

III. Implementação é a fase reservada para criação da visão do projeto, identificação do *scrum master*, dos *stakeholders* e do time *Scrum*.

IV. Revisão e retrospectiva são atividades que fazem parte da fase reservada, de realização de demonstrações, validações da *sprint* e sua retrospectiva.

V. Release é a fase reservada para disponibilização dos entregáveis aos clientes. Além disso, ocorre a retrospectiva de todo o projeto.

Considerando o contexto apresentado, é correto o que se afirma em:

a. I e III, apenas.

b. II e III, apenas.

c. III, IV e V, apenas.

d. I, II, III e IV, apenas.

e\_ I, II, III, IV e V.

## REFERÊNCIAS

COHN, M. **Desenvolvimento de software com Scrum**: aplicando métodos ágeis com Sucesso. Porto Alegre: Bookman, 2011.

CRUZ, F. **Scrum e PMBOK**: unidos no gerenciamento de projetos. Rio de Janeiro: Brasport.2013.

SCRUMstudy. **Um guia para o conhecimento em scrum (Guia SBOK)**. Avondale: SCRUMstudy, 2016.

SOMMERVILLE, I. **Engenharia de Software**. 9. ed. São Paulo: Pearson Prentice Hall, 2011.

Ver anotações