

NÃO PODE FALTAR

DIFERENÇAS E SEMELHANÇAS NA CONCEPÇÃO E NO DESENVOLVIMENTO DO PROJETO

Izabelly Soares de Moraes



Fonte: Shutterstock.

Deseja ouvir este material?

Áudio disponível no material digital.

CONVITE AO ESTUDO

Olá, prezado aluno! Seja bem-vindo à quarta unidade da disciplina de Projeto de Software, que apresentará informações relevantes referentes a algumas áreas sobre o ciclo de vida e ações que ocorrem durante o desenvolvimento de um sistema. Para abrangermos esses assuntos, trataremos das diferenças e semelhanças na concepção e no desenvolvimento do projeto, e como complemento, veremos as ferramentas para gestão de projetos. Além disso, vamos compreender o ciclo de vida sob perspectivas diferentes, já que lidamos com certa diversidade de produtos e serviços que fazem uso do software para ser promovido.

Na **primeira seção**, você aprenderá as diferenças e as semelhanças na concepção e no desenvolvimento do projeto. Trataremos das atividades que alguns membros da equipe devem tomar diante da organização de um projeto de software, já que é por meio da colaboração de todos que os processos vão sendo desenvolvidos com sucesso. Então, ficará clara a função que cada um terá, como os *stakeholders*, o gerente de projetos, o *product owner* e, olhando sob uma perspectiva macro, como os times que adotam a filosofia ágil se organizam.

Saiba que, além de terem papéis e funções bem definidas dentro de um projeto ágil, existem ferramentas que auxiliam esses profissionais em suas atividades, e trataremos desse assunto na **segunda seção**. Inclusive, gerir a comunicação é um desafio. Por exemplo, imagine uma equipe com mais de 10 pessoas e vários

artefatos sendo gerados a cada momento. Sem organização, a equipe poderá não atender aos objetivos traçados durante o planejamento. Portanto, existem ferramentas que auxiliam nesse momento, além disso, elas podem ser utilizadas tanto em metodologias ágeis quanto em metodologias tradicionais (sequenciais).

Por fim, na **terceira seção**, complementando conceitualmente os assuntos que serão vistos nas seções anteriores, você verá as diferentes abordagens do ciclo de vida do software e tratará de assuntos sobre manutenção do software, responsabilidades assumidas no projeto, planejamento de releases e como ocorrem entregas e pós-entregas.

O importante é você conhecer esses conceitos e ter consciência de que eles podem ser aplicados em cenários diversos, em que você será o protagonista capaz de aplicar não só as metodologias, mas também de fazer uso da ferramenta que melhor servir.

Bons estudos e boa sorte!

PRATICAR PARA APRENDER

Prezado aluno, você vai aprender, nesta seção, o papel dos *stakeholders*, que podem ser representados pelos clientes ou usuários do sistema. Em outros tipos de projetos, eles podem envolver também os membros da equipe de desenvolvimento, designers, entre outros profissionais, mas quando a metodologia é ágil e é a *Scrum*, quem participa do desenvolvimento em si é chamado de *Time Scrum*.

Outro ponto interessante é a presença do gerente de projetos, responsável por diversas atividades voltadas ao projeto em geral, exercendo uma função que percorre todas as demais, sempre verificando se o que foi planejado está sendo executado. Note que o objetivo não é apenas realizar a atividade, mas realizá-la da maneira correta. No *Scrum*, existe um papel denominado *Scrum Master*, que deve garantir que as práticas do *Scrum* sejam aplicadas. Já o *Product Owner* assume o papel denominado “dono do produto”, em que, na realidade, ele se torna o representante do negócio do cliente dentro do projeto, ou seja, conhece tudo que pode agregar valor ao negócio do cliente.

O objetivo dessa divisão de papéis é garantir que cada um faça a sua parte no projeto e que, ao ser finalizado, as habilidades utilizadas tenham sido fundamentais para a qualidade de um produto que, por meio dos processos executados durante o seu desenvolvimento, atenda às necessidades do cliente e agregue valor ao seu negócio. Essa auto-organização dos times ágeis trazem um diferencial e vantagens em utilizar esse tipo de metodologia.

O uso do software agrega ao negócio a eficiência e eficácia dos processos, já que automatiza várias atividades que antes eram desempenhadas apenas pelo ser humano, e muitas, passíveis de erros. Apesar de não serem totalmente perfeitos, os software são desenvolvidos e podem ir sendo adaptados de acordo as demandas dos clientes. Então, se hoje o sistema precisa realizar leitura de código de barras, amanhã ele pode ser adaptado para realizar leitura de QRCode, e assim por diante, por isso, é um recurso muito requisitado em vários nichos de negócios.

Uma fábrica de mantimentos alimentícios criada em 1980 decidiu modernizar seus sistemas, já que alguns de seus processos ainda aconteciam de forma tradicional, e quando se tratava de atividades burocráticas, como elaboração de relatórios e documentos, tinha como base, dados armazenados em papéis e registros da empresa.

Com o objetivo de agilizar e dar continuidade a essa ideia, a empresa criou seu próprio setor de tecnologia, onde alguns de seus sistemas serão desenvolvidos e outros serão obtidos por meio de parcerias com outras empresas.

Por isso, a empresa de tecnologia que você trabalha foi contratada, e você assumirá o papel de gerente de projeto em um projeto com o objetivo de criar um sistema voltado à contabilidade da empresa. Por fim, esse sistema deverá ser capaz de gerar relatórios gerenciais com todos os dados obtidos por meio dos demais sistemas, ou seja, haverá uma integração para a geração dos dados.

Devido aos seus conhecimentos, você participará de todas as etapas e deverá apresentar o planejamento inicial de todos os passos que serão executados durante o ciclo de vida do software.

Para isso, utilize a estrutura analítica do projeto, apresentando as etapas que ele assumirá e que deverá ser desenvolvida pelo time envolvido.

Saber cada atividade e o cargo envolvido nos processos de software com o uso da metodologia ágil é muito importante. Agora, você será capaz de compreender melhor não só as cerimônias, mas também como cada atividade ocorre!

CONCEITO-CHAVE

PAPEL DOS *STAKEHOLDERS* EM CADA UMA DAS ABORDAGENS

Hoje, o software é algo primordial, podendo exercer funções diversas. Então, passou a assumir outros campos de atuação, seja como um produto, seja como serviço que pode apresentar soluções para algum problema idealizado por alguém ou por alguma situação. Trazendo essa ideia para a prática, esse alguém é o cliente, ou seja, aquele que sentiu, em algum momento, que precisava automatizar alguma ação de sua rotina, que pode ou não fazer parte diretamente de seu cotidiano ou do seu usuário, que é o cliente do cliente.

Por exemplo, caso esteja desenvolvendo um sistema para um supermercado, você poderá atender tanto às demandas do cliente, que pode estar envolvido diretamente com o negócio, como ser proprietário do estabelecimento e financiar o projeto para o usuário, que fará uso do sistema de outra perspectiva. Ainda dentro desse cenário, podemos ter os funcionários do supermercado, ou seja, temos três perfis diferentes que terão demandas diversificadas em relação às funcionalidades do sistema.

Dentro dessa iteração, ainda se encontram profissionais aptos a desenvolver essa solução para o cliente, cada um com sua expertise, então, teremos desenvolvedores, equipe de especificação de requisitos, designers, analistas, entre outros. Até este ponto, você pode pensar que isso é suficiente, porém é

importante mencionarmos a necessidade da definição de um processo de

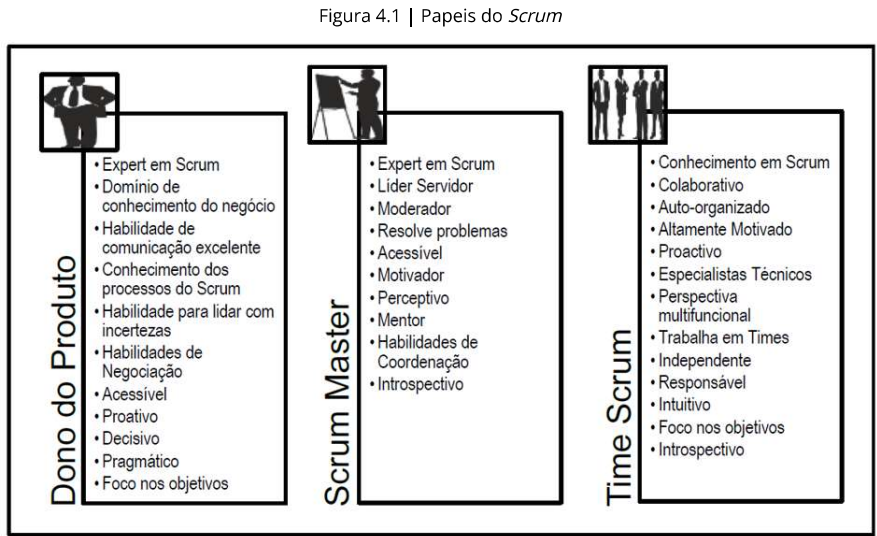
desenvolvimento, da aplicação de metodologias e de inovação constante, pois, um dos maiores desafios é lidar com a diversidade e a quantidade de atividades que estão envolvidas nessa missão.

A organização de um time pode variar bastante, a filosofia ágil defende alguns princípios que acabam interferindo também nesse momento. Entre eles, estão: indivíduos e interações são vistos como mais importantes do que processos e ferramentas, colaboração com o cliente mais que negociação de contratos, software em funcionamento mais que documentação abrangente, responder a mudanças mais que seguir um plano.

Partindo desses princípios, os indivíduos são muito relevantes em um projeto; na realidade, sempre foram, porém, as metodologias com processos mais tradicionais ou sequenciais, como também são conhecidos, trazem outros pontos relevantes; a partir daí, podemos começar a notar diferenciais entre as metodologias. Para Pressman (2016, p. 93), a filosofia ágil “ênfatiza a satisfação do cliente e a entrega prévia incremental de software, pequenas equipes de projeto altamente motivadas, métodos informais, mínimos artefatos de engenharia de software e total simplicidade de desenvolvimento.”

■ PAPÉIS DA METODOLOGIA SCRUM

O *Scrum* possui algumas semelhanças com a XP, porém traz nomenclaturas aos membros da equipe que utilizará os conceitos da metodologia. A Figura 4.1 traz as principais características de cada papel desempenhado no *Scrum*.



Fonte: SCRUMstudy 2016, p. 48).

Ao observar a Figura 4.1, é possível notar as diferentes atividades que podem ser desempenhadas no processo de desenvolvimento. Mas não se resumem a apenas essas, uma vez que o modo de atuação pode variar à medida que a etapa do ciclo está sendo executada em determinado momento.

O Time *Scrum* é composto pelos desenvolvedores, designers, especificadores e demais pessoas que atuarão diretamente no desenvolvimento do sistema, ou seja, são eles que vão lidar com os *stories*, com as necessidades dos clientes. O Quadro 4.1 traz os processos e as responsabilidades do **Time Scrum**.

Quadro 4.1 | Responsabilidades do Time *Scrum* em processos do *Scrum*

| Processos | Responsabilidades do Time Scrum |
|---|--|
| Formar o Time <i>Scrum</i> . | <ul style="list-style-type: none">• Fornecer <i>inputs</i> para a criação do plano de colaboração e plano de <i>team building</i>. |
| Desenvolver os épico(s). | <ul style="list-style-type: none">• Garantir uma compreensão clara sobre os épico(s) e as personas. |
| <i>Backlog</i> priorizado do produto. | <ul style="list-style-type: none">• Compreender as histórias de usuário no <i>backlog</i> priorizado do produto.• Concordar com outros membros do Time cental do <i>Scrum</i> sobre a duração da <i>sprint</i>.• Buscar esclarecer novos produtos ou mudanças nos produtos já existentes, se houver, no <i>backlog</i> priorizado do produto refinado. |
| Conduzir o planejamento da <i>release</i> . | <ul style="list-style-type: none">• Concordar com outros membros do Time cental do <i>Scrum</i> sobre a duração da <i>sprint</i>.• Buscar esclarecer novos produtos ou mudanças nos produtos já existentes, se houver, no <i>backlog</i> priorizado do produto refinado. |
| Criar as histórias de usuário. | <ul style="list-style-type: none">• Fornecer inputs para o dono do produto na criação das histórias de usuário. |
| Aprovar, estimar e comprometer as histórias de usuário. | <ul style="list-style-type: none">• Estimar as histórias de usuário aprovadas pelo dono do produto.• Comprometer as histórias de usuário a serem concluídas na <i>sprint</i>. |
| Criar as tarefas. | <ul style="list-style-type: none">• Desenvolver a lista de tarefas com base em histórias de usuário e dependências acordadas. |
| Estimar as tarefas. | <ul style="list-style-type: none">• Estimar os esforços para as tarefas identificadas e, se necessário, atualizar a Lista de Tarefas. |
| Criar o <i>backlog</i> da <i>Sprint</i> . | <ul style="list-style-type: none">• Desenvolver o <i>backlog</i> da <i>sprint</i> e o gráfico <i>burndown</i> da <i>sprint</i>. |
| Criar os entregáveis. | <ul style="list-style-type: none">• Criar os entregáveis.• Identificar riscos e implementar ações de mitigação de risco, se houver.• Atualizar o Registro de impedimento e dependências. |

| | |
|--|---|
| Conduzir a reunião diária. | <ul style="list-style-type: none"> • Atualizar o gráfico <i>burndown</i>, <i>scrumboard</i> e o registro de impedimentos. • Discutir problemas enfrentados por membros individuais e buscar soluções para motivar o time. • Identificar riscos, se houver. • Submeter solicitações de mudança, se necessário. |
| Refinamento do <i>backlog</i> priorizado do produto. | <ul style="list-style-type: none"> • Participar de reuniões de revisão do <i>backlog</i> priorizado do produto. |
| Convocar o <i>Scrum</i> de <i>Scrums</i> . | <ul style="list-style-type: none"> • Fornecer inputs ao <i>Scrum Master</i> para Reuniões do <i>Scrum</i> de <i>Scrums</i> (SoS). |
| Demonstrar e validar a <i>Sprint</i> . | <ul style="list-style-type: none"> • Demonstrar ao dono do produto as entregas concluídas, que requerem aprovação. |
| Retrospectiva da <i>Sprint</i> . | <ul style="list-style-type: none"> • Identificar oportunidades de melhorias na <i>Sprint</i> atual e concordar com todas as melhorias viáveis para a próxima <i>Sprint</i>. |
| Retrospectiva do projeto. | <ul style="list-style-type: none"> • Participar da reunião de retrospectiva do projeto. |

Fonte: SCRUMstudy (2016, p. 47).

ASSIMILE

Devido à multidisciplinaridade do Time *Scrum*, um plano de *Team Building* é formado pelo *Scrum Master* com o objetivo de sanar algum problema que possa surgir envolvendo o time. O objetivo é sempre prezar pela colaboração e participação efetiva de todos durante o projeto. O *Team Building* tem uma visão de “trabalho em equipe”, ressaltando a ideia de colaboração.

No Quadro 4.1, é possível notar que, para cada processo, o Time *Scrum* deve executar ações que coerentes com as atividades a serem realizadas. Além disso, conforme as perspectivas do *Scrum*, tudo que é produzido é de responsabilidade de todos da equipe, ou seja, devem prezar pela qualidade, já que cada um contribui, de alguma forma, para o projeto.

PRODUCT OWNER (DONO DO PRODUTO)

Outro papel bem importante dentro do time ágil *Scrum* é o do **Product Owner** (P.O), ou, em português, “dono do produto”. Ele representa o cliente, logo, deve saber de todos os detalhes que envolvem o negócio do cliente, a fim de que possa contribuir, junto dos demais membros da equipe, para agregar valor as propostas trazidas por ele durante todo o processo.

Segundo Ferreira (2020), é de responsabilidade do P.O pensar na melhor estratégia para alinhar com a equipe quais requisitos devem ser inseridos no *backlog* da *Sprint*, pois ele sabe qual funcionalidade trará mais valor ao negócio do cliente. Então, por exemplo, se o sistema que está em desenvolvimento é para uma farmácia e o objetivo inicial do cliente é realizar o fluxo de estoque, uma

funcionalidade que talvez seja de maior prioridade é a de realizar o cadastro dos produtos. Claro que as demais não serão negligenciadas, porém a visão do que realmente é prioridade é muito importante. Então, após a primeira *sprint*, essa funcionalidade estaria pronta e o P.O verificaria, conforme os critérios de aceitação, se realmente o artefato dessa *sprint* atende às necessidades do cliente. Caso não atenda, o item volta ao *backlog* do produto, onde será analisado e voltará para a *sprint* para ser corrigido ou, após a análise da viabilidade das mudanças, ser totalmente renovado. Portanto, deixar claro que o projeto tem fundamentos e realizar a comunicação entre os *stakeholders* para garantir a entrega de valor ao projeto são as principais funções do P.O.

SCRUM MASTER E GERENTE DE PROJETOS

Outro papel que também possui atividades voltadas a sua função é a de **Scrum Master**, e ao contrário do que todos pensam, não é a mesma função do **gerente de projetos**, já que sua função principal é a de garantir que todas as cerimônias e estratégias do *Scrum* estejam sendo executadas no projeto e de forma correta. Fazendo um comparativo entre as duas funções, o gerente de projetos desempenha atividades voltadas a uma perspectiva geral, que acompanha todo o planejamento do projeto em si.

Conforme o Guia PMBOK (2017, p. 552), o gerente de projetos “é a pessoa designada pela organização executora para liderar a equipe responsável por alcançar os objetivos do projeto. Os relacionamentos de subordinação do gerente do projeto baseiam-se na estrutura da organização e na governança do projeto.”

Logo, note pontos que o diferenciam do Scrum Master, pois, em projetos gerais, podem existir níveis hierárquicos, porém na metodologia ágil, essa não é a ideia. Ainda conforme a perspectiva trazida pelo Guia PMBOK (2017), é de responsabilidade do gerente de projetos adaptar abordagens, ciclo de vida e processos voltados ao gerenciamento do seu projeto para satisfazer as necessidades dos envolvidos, o que deixa claro que nem sempre é obrigatório escolher uma metodologia ágil para seu projeto.

Por exemplo: existem alguns cenários em que uma metodologia tradicional ou sequencial, como é comumente chamada, pode ser aplicada, pois trará bons resultados, talvez até mais eficientes que outros tipos de abordagens. Geralmente, projetos pequenos têm sucesso diante do seu uso, o que não elimina também a aplicação em projetos maiores, mas, atualmente, a metodologia ágil está ganhando mais espaço, até mesmo porque muda um pouco a cultura organizacional, trazendo ideia de transparência, colaboração e de que o resultado é oriundo do trabalho de todos.

O *Scrum Master* é bem focado nas práticas do *Scrum*, então, claramente, não pode ser comparado com o trabalho do gerente de projetos, já que esse papel existe apenas quando a metodologia ágil *Scrum* é utilizada. No Quadro 4.2 é possível conhecer um pouco mais as responsabilidades do *Scrum Master*.

Quadro 4.2 | Responsabilidades do *Scrum Master* em processos do *Scrum*

| Processos | Responsabilidades do <i>Scrum Master</i> |
|--|---|
| Identificar o <i>Scrum Master</i> e o(s) <i>Stakeholder(s)</i> . | <ul style="list-style-type: none"> Ajudar a identificar o(s) <i>Stakeholder(s)</i> para o projeto. |
| Formar o Time <i>Scrum</i> . | <ul style="list-style-type: none"> Facilitar a seleção do Time <i>Scrum</i>. Facilitar a criação do plano de colaboração e do plano de <i>team building</i>. Garantir a disponibilidade de backup de recursos para o bom funcionamento do projeto. |
| Desenvolver o(s) épico(s). | <ul style="list-style-type: none"> Facilitar a criação de épico(s) e de personas. |
| Criar o <i>backlog</i> priorizado do produto. | <ul style="list-style-type: none"> Ajudar o dono do produto na criação do <i>backlog</i> priorizado do produto e na definição dos critérios de pronto. |
| Conduzir o planejamento da <i>release</i> . | <ul style="list-style-type: none"> Coordenar a criação do cronograma de planejamento da <i>release</i>. Determinar a duração da <i>Sprint</i>. |
| Criar as histórias de usuário. | <ul style="list-style-type: none"> Auxiliar o Time <i>Scrum</i> na criação das histórias de usuário e em seus critérios de aceitação. |
| Aprovar, estimar e comprometer as histórias de usuário. | <ul style="list-style-type: none"> Facilitar as reuniões do Time <i>Scrum</i> para estimar e criar as histórias de usuário. |
| Criar as tarefas. | <ul style="list-style-type: none"> Facilitar ao Time <i>Scrum</i> a criação da lista de tarefas para a próxima <i>Sprint</i>. |
| Estimar as tarefas. | <ul style="list-style-type: none"> Auxiliar o Time <i>Scrum</i> a estimar os esforços. |
| Criar o <i>backlog</i> da <i>Sprint</i> . | <ul style="list-style-type: none"> Auxiliar o Time <i>Scrum</i> no desenvolvimento do <i>backlog</i> da <i>Sprint</i> e do gráfico <i>burndown</i> da <i>Sprint</i>. |
| Criar os entregáveis. | <ul style="list-style-type: none"> Suportar o Time <i>Scrum</i> na criação das entregas acordadas para a <i>Sprint</i>. Ajudar a atualizar o <i>scrumboard</i> e o registro de impedimentos. |
| Conduzir a reunião diária. | <ul style="list-style-type: none"> Garantir que o <i>scrumboard</i> e o registro de impedimentos continuem sendo atualizados. |
| Refinamento do <i>backlog</i> priorizado do produto. | <ul style="list-style-type: none"> Facilitar as reuniões de revisão do <i>backlog</i> priorizado do produto. |
| Convocar o <i>Scrum</i> de <i>Scrums</i> . | <ul style="list-style-type: none"> Garantir que os problemas que afetam o Time <i>Scrum</i> sejam discutidos e resolvidos. |

| | |
|--|--|
| Demonstrar e validar a <i>Sprint</i> . | <ul style="list-style-type: none"> • Facilitar a apresentação de entregas concluídas pelo Time <i>Scrum</i> para a aprovação do dono do produto. |
| Retrospectiva da <i>Sprint</i> . | <ul style="list-style-type: none"> • Garantir a existência de um ambiente ideal para o projeto, para o Time <i>Scrum</i> durante as <i>Sprints</i> seguintes. |
| Retrospectiva do projeto. | <ul style="list-style-type: none"> • Representar o Time Central do <i>Scrum</i>, fornecendo lições do projeto atual, se necessário. |

Fonte: SCRUMstudy (2016, p. 45).

É possível notar a associação entre cerimônias do *Scrum* e o papel desempenhado pelo *Scrum Master*. Por isso, suas funções são tão importantes, já que lida diretamente com o *backlog* e, conseqüentemente, pode nortear a continuidade e o progresso do projeto.

EXEMPLIFICANDO

Em algumas situações, existe a necessidade da organização dos vários processos, de cerimônias e times utilizando a metodologia ágil *Scrum*. A "*Scrum de Scrums*" foi desenvolvida com o intuito de trazer uma escala ágil para os projetos; ela é responsável por proporcionar conexão entre equipes diferentes em busca de soluções complexas. Então, em um cenário fictício, imagine a existência de um projeto com mais de 100 membros divididos em 10 equipes, cada uma com seu *Scrum Master*, *product pwner* e seu Time *Scrum*. Então, as cerimônias serão realizadas dentro das equipes e, posteriormente, com seus *Scrum Masters* e *product owners* reunidos.

Mas não se esqueça de que, na metodologia ágil, não existe hierarquia de cargos, então, o *Scrum Master* segue no mesmo nível que os demais. Ainda podem ser encontrados outros papéis, como o *Scrum Master* do Programa (para um programa) ou um *Scrum Master* do Portfólio (para um portfólio) (SCRUMSTUDY, 2016).

REFLITA

Apesar da metodologia ágil trazer uma ideia de simplificar os processos ou as etapas de desenvolvimento, percebe-se que existem, na verdade, várias definições a serem seguidas, entre elas, é possível notar que a forma como são utilizadas podem variar de acordo as necessidades das equipes. Mas pense que todas essas cerimônias são oriundas de anos de experiências, que trouxeram pontos positivos e negativos que podem estar presentes em um projeto de software. Você, estando inserido em uma equipe com esse cenário, adotando o *Scrum*, com qual papel você mais se identificaria, sabendo que poderia assumir o de *Scrum Master*, *Product Owner* ou algum membro do Time *Scrum*?

Para melhor compreensão, um **time ágil** possui a divisão clara de cada função dentro da equipe; o profissional consegue direcionar melhor seus conhecimentos e suas habilidades, porém isso não deve ser visto como trabalho individual, pelo contrário, lembre-se dos princípios das metodologias ágeis e da colaboração com os demais envolvidos no projeto. Por exemplo, apesar da importância das funcionalidades, existe também os especificadores, designers que trabalham juntos, organizando as informações da melhor maneira possível para que os desenvolvedores possam fazer a implementação.

Conforme a metodologia, a organização da equipe ágil pode variar um pouco. Por exemplo, na Programação Extrema (XP), alguns valores se destacam, tais como a importância da comunicação, a simplificação nos processos e a iteração com *feedbacks*, o que contribui para a busca pela qualidade tanto nas atividades como no produto final. Inclusive, a comunicação e a transparência nos processos é essencial, pois, por meio dessa ação, todo projeto flui. O ponto de vista simplista traz a essas equipes, durante sua fase de planejamento inicial, a projeção apenas do que deve ser realizado naquele momento, já que a cada iteração, mudanças podem ocorrer.

A ideia da colaboração traz a todos os envolvidos a responsabilidade de seguir o que vai sendo definido ao longo do processo, já que, na metodologia ágil, mudanças ocorrem a cada ciclo. Além disso, a colaboração entre a equipe é primordial para o sucesso do projeto. Cabe a cada membro desempenhar suas atividades da melhor maneira possível, até mesmo porque a qualidade ainda é um ponto alto que trouxe destaque às vantagens de se utilizar uma metodologia ágil.

FAÇA A VALER A PENA

Questão 1

A globalização acaba trazendo um cenário de operação às empresas de tecnologia que demanda a produção de software cada vez mais rápido, porém que garante a qualidade, independentemente da complexidade do que está sendo desenvolvido. Para se adequar à nova realidade, muitas adotam metodologias que trazem agilidade a todo o processo. Uma das filosofias dessas medidas é definir uma função para cada membro da equipe. A função de verificar se o que foi entregue realmente atende à necessidade e aos critérios de aceitação relacionados ao negócio do cliente. Para isso, o papel de um dos componentes da equipe é muito importante.

Assinale a alternativa correta:

a. Esse é o papel do Time *Scrum*.

b. Esse é o papel do *Scrum Master*.

c. Esse é o papel do *Product Owner*.

d. Esse é o papel do cliente.

e. Esse é o papel do usuário.

Questão 2

As mudanças trazem grandes desafios a qualquer pessoa, e no desenvolvimento de software não é diferente. Na atualidade, as demandas passam por esse fenômeno frequentemente, fazendo com que a equipe envolvida no projeto tenha que se adaptar para conseguir cumprir com suas atribuições.

Tomando como referência os diferentes papéis que podem ser assumidos em um projeto de software, julgue as afirmativas a seguir como (V) Verdadeiras ou (F) Falsas.

() O *Scrum Master* é um especialista técnico com habilidades específicas em uma tecnologia.

() O *Time Scrum* deve ser colaborativo e auto-organizado para conseguir desempenhar suas funções.

() O Dono do produto tem o foco nos objetivos e deve saber lidar com incertezas durante o projeto.

() O *Scrum Master* entende de todas as cerimônias do Scrum e garante a aplicabilidade correta de todas elas pela equipe.

Assinale a alternativa que apresenta a sequência CORRETA.

a. F - V - V - V - V.

b. V - V - F - F - F.

c. F - V - V - F - F.

d. V - F - F - F - F.

e. F - V - F - F - F.

Questão 3

O gerente de projetos é uma função que foi criada para gerir os processos de desenvolvimento mais tradicionais, a diferença do *Scrum Master* é bem significativa e as duas funções não devem, em momento algum, ser comparadas. Além disso, os demais envolvidos no projeto também recebem atividades direcionadas a seus papéis dentro da equipe.

Tomando como base as atividades desses dois papéis dentro de um projeto de software, avalie as informações a seguir.

I. O *Scrum Master* expõe, para todos os envolvidos no projeto, as práticas *Scrum*.

II. O gerente de projetos e o *Scrum Master* desempenham as mesmas funções, são responsáveis pelas cerimônias *Scrum* de um projeto.

III. O *Scrum Master* possui o mesmo nível hierárquico que todos os demais envolvidos no projeto.

IV. O gerente de projeto é visto como um líder e é responsável pelo planejamento e pelo acompanhamento de todo o projeto.

V. O gerente de projetos e o *Scrum Master* são responsáveis pelo negócio do cliente e pelos critérios de aceitação das entregas de software.

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

a. I e III, apenas.

b. II, III e IV, apenas.

c. III, IV e V, apenas.

d. I, III e IV, apenas.

e. V e IV, apenas.

REFERÊNCIAS

FERREIRA, M. B. **Métodos ágeis e melhoria de processos**. Curitiba: Contentus, 2020.

PMI — Project Management Institut. **Guia do conhecimento em gerenciamento de projetos: guia PMBOK**. 6. ed. [S.l.]: PMI, 2017.

PRESSMAN, R. S.; MAXIM, B. R. **Engenharia de software: uma abordagem profissional**. 8. ed. Porto Alegre: AMGH, 2016.

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