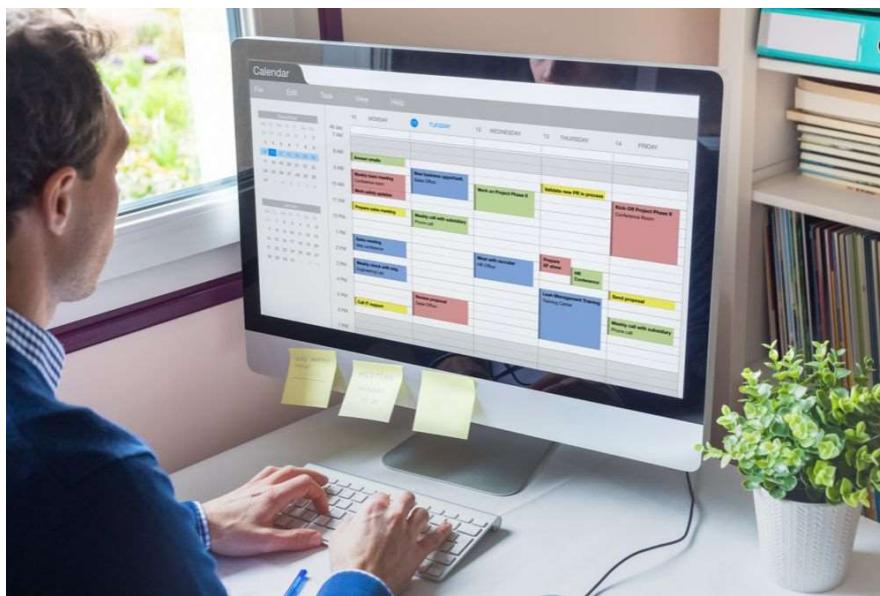


NÃO PODE FALTAR

## MANIFESTO E MÉTODOS ÁGEIS PARA GERENCIAMENTO DE SOFTWARES

Izabelly Soares de Morais



Fonte: Shutterstock.

### Deseja ouvir este material?

Áudio disponível no material digital.

#### CONVITE AO ESTUDO

Olá, prezado aluno! Seja bem-vindo a esta unidade que abrangerá pontos interessantes sobre o software e os processos que norteiam o seu desenvolvimento, tais como os modelos de projetos de software, tendo como foco os que adotam a filosofia ágil, como o *Scrum*, bem como técnicas de design, que também são aplicadas em projetos na área de tecnologia. Esses recursos ágeis, no decorrer dos anos, passaram a ser uma ferramenta primordial nas atividades cotidianas. Após a popularização dos equipamentos eletrônicos, essas ferramentas se tornaram primordiais para a automatização das atividades, independentemente de sua complexidade.

Por meio do conhecimento sobre o ciclo de desenvolvimento de um sistema, tendo como base o uso das metodologias ágeis, você será capaz de aplicá-lo em seu cotidiano profissional, utilizando as práticas de análise para propor soluções de problemas. Além disso, você será capaz de modelar e aplicar técnicas ágeis e de design em projetos de software.

Neste material, serão abordados todos os conceitos que contextualizam a existência atual da metodologia ágil, tais como: seus fundamentos, valores e alguns exemplos das principais metodologias utilizadas. Levando-se em consideração a importância dos assuntos abordados, conteúdo será dividido em

três momentos, a fim de que cada um apresente a você os objetivos específicos e as competências que, juntos, promoverão a aquisição de conhecimentos ricos em detalhes.

Na primeira seção, você compreenderá conceitos que norteiam o surgimento do termo “ágil” voltado ao universo do desenvolvimento de software. Para isso, você vai conhecer o que foi o “Manifesto Ágil” e seus respectivos valores, e para identificar a influência que a mudança das percepções dos profissionais da área de tecnologia causou nos processos de projeto de software, serão apresentados conceitos sobre as principais metodologias ágeis que surgiram desde então. Além disso, você vai notar que todo o empiricismo trazido por essas novas ideias vieram com o objetivo de tornar menos complexo todo o ciclo de vida do software.

Em seguida, após ter adquirido conhecimento acerca dos temas relatados na primeira seção você compreenderá melhor a origem e a definição do *Scrum*; verá que é uma metodologia ágil que possui diversas particularidades, como definições de papéis dentro da equipe, geração de artefatos e rituais ou cerimônias que são aplicadas para trazer agilidade ao processo.

Já na terceira seção, você aprenderá sobre a origem do uso de técnicas de design, tais como o *design thinking*, mas com a perspectiva voltada aos projetos de tecnologia, bem como sobre o MVP, sigla utilizada para o que chamamos de mínimo produto viável, que pode ser considerada uma versão inicial daquela sua proposta de software. Com a flexibilidade levantada pela filosofia do Manifesto Ágil, falaremos a respeito da pivotagem, ou seja, da capacidade de uma equipe envolvida em um projeto de software de realizar mudanças de forma prática, sem trazer prejuízos a todo o projeto.

Agora que já sabe os assuntos que serão estudados, prepare-se para embarcar, ainda mais a fundo, no universo do projeto de software!

Uma dica: ao conhecer os conceitos, imagine sempre como aplicá-los em futuras ou presentes situações de trabalho.

Bons estudos e boa sorte!

#### PRATICAR PARA APRENDER

Prezado aluno, ao estudarmos tecnologia, percebemos a quantidade de assuntos que são abordados em diversos contextos e que acabam se tornando um arcabouço de definições e possibilidades de práticas. No entanto, saiba que o mais importante dessa viagem no mundo tecnológico é a ação, pois, na prática, o seu diferencial, enquanto profissional da área, está na forma como você lida com as situações adversas do dia a dia.

Empresas da área de TI atuam no mais variado tipo de segmento, devido à grande relevância da automatização de processos trazidos pela tecnologia. Portanto, nesta seção, você conhecerá como surgiu o uso da metodologia ágil dentro do processo de desenvolvimento de software. Saiba que, de início, vários estudiosos da área decidiram se reunir e criar o “Manifesto Ágil”, que foi responsável por todas as

ações voltadas à aplicação e ao desenvolvimento da metodologia ágil no mundo. Aqui, você também verá alguns tipos de metodologias ágeis, e até tradicionais, com o objetivo de reconhecer a diferença entre ambas.

Uma determinada empresa de tecnologia possui 30 anos de experiência no mercado tecnológico e seu principal ramo de negócio são sistemas para redes de supermercados. Esse tipo de sistema começou a operar na empresa a menos de 10 anos, e devido à complexidade exposta por essa demanda, os principais funcionários do setor de tecnologia que atuam na empresa notaram que seus processos de desenvolvimento estavam bastante onerosos, pois, com a alta concorrência, as demais empresas estavam produzindo software em um tempo reduzido e com uma qualidade superior. Além disso, o desafio se estende à velocidade em que a tecnologia muda.

Dessa forma, você, como contratado de uma empresa de tecnologia para implantar a metodologia ágil em uma equipe acostumada com a linearidade da filosofia de desenvolvimento tradicional, terá como desafio analisar todo o contexto envolvido e, posteriormente, aplicar a metodologia ágil XP (Programação extrema). Como a XP poderá ser implantada na empresa, tendo como objetivo deixar os software mais competitivos no mercado?

Você, com a sua expertise, está ciente de que a metodologia ágil pode ser desenvolvida utilizando-se diversas ferramentas, como é o caso da XP e do *Scrum*. Para esse momento, você resolveu implantar a metodologia XP, porque observou que, além de ajudar no processo, também deixará o software mais atrativo para ao mercado.

Dadas as circunstâncias, você, então, deverá desenvolver um relatório para a gerência, apontando os principais atributos e os processos para o seu desenvolvimento. Lembre-se de que a primeira etapa é a de planejamento, logo, analise como a equipe está atuando em seus projetos, como documenta as informações, se aplica alguma boa prática de desenvolvimento ou, até mesmo, se segue alguma metodologia de desenvolvimento de forma fiel aos seus princípios. Para complementar seu relatório inicial de planejamento, você pode incluir relatos de como as atividades básicas do projeto de software estão sendo executadas na empresa.

Então, vamos começar a pensar nisso imediatamente! Saiba, também, que seus conhecimentos de vida e sua experiência também contam nessa aventura, pois, ao longo da vida, somos capazes de nos adaptar a diversas situações, e com os conhecimentos aqui adquiridos, você vai longe!

## CONCEITO-CHAVE

### METODOLOGIAS PARA O DESENVOLVIMENTO DE SOFTWARE

Em toda área de estudo, existem conceitos que foram sendo consolidados ao longo dos anos, por diversos estudiosos e especialistas, e com a engenharia de software não foi diferente. Em meados da década de 1970, começou a surgir a necessidade de se definir metodologias para o processo de desenvolvimento de

software com o objetivo de estabelecer etapas que norteassem todo o ciclo, o qual possui alguns processos considerados como fundamentais, entre eles, Sommerville (2011, p.18) destaca:

- Especificação de software: funcionalidade do software e as restrições a seu funcionamento devem ser definidas.
- Projeto e implementação de software: o software deve ser produzido para atender às especificações.
- Validação de software: o software deve ser validado para garantir que atenda às demandas do cliente.
- Evolução do software: o software deve evoluir para atender às necessidades de mudança dos clientes.

### METODOLOGIA OU MODELO CASCATA

Apesar de serem considerados fundamentais, tais processos possuem subatividades, para que juntas consigam contribuir para a finalização do software. Essa primeira ideia serviu de base para criação das demais metodologias. Considerada uma das primeiras, conforme Pressman (2011), a metodologia ou modelo Cascata foi criada em meados da década de 1970, e sua ideia principal, como seu próprio nome já sugere, é que uma etapa seja executada apenas quando a anterior for finalizada, além dos processos fundamentais citados acima, o modelo apresenta algumas outras etapas, com isso, seu ciclo fica com a seguinte ordem de execução:

- **Comunicação:** que contempla a inicialização de um projeto e, consequentemente, a etapa de levantamento de requisitos.
- **Planejamento:** em que o cronograma e as estimativas do projeto são definidos.
- **Modelagem:** sendo contemplada pela análise e pelo projeto do sistema em si.
- **Construção:** refere-se à implementação, em que a codificação é complementada com os testes.
- **Entrega:** que abrange também a etapa de suporte e implantação do sistema no ambiente do cliente.

#### VOCABULÁRIO

O modelo cascata também pode ser conhecido como "modelo sequencial linear" ou "modelo tradicional". Ele recebe esses nomes por ter sido um dos primeiros modelos de desenvolvimento de software.

### DESENVOLVIMENTO INCREMENTAL

Após a criação do modelo cascata, outras necessidades durante o processo foram sendo percebidas, contribuindo para que novas metodologias fossem criadas, tendo como base sempre o que já foi testado até o aquele momento. Com isso, surgiram outros modelos, tais como: desenvolvimento incremental, que foi desenvolvido no intuito de suprir a necessidade de dar mais agilidade ao processo,

tendo em vista que, com a perspectiva da aplicação cascata, a execução de uma nova etapa ficava sujeita ao término da etapa anterior e, em algumas situações, estava ficando inviável, por exemplo, realizar testes apenas nas etapas finais ou, até mesmo, validar alguma ideia do sistema apenas quando ele fosse implementado e/ou testado.

o  
Ver anotações**ASSIMILE**

A escolha da metodologia a ser utilizada é sempre um desafio. Dessa forma, o modelo cascata pode ser viável em um projeto considerado curto; as suas etapas conseguem ser executadas de forma mais simples e, consequentemente, os testes poderão retornar com menos frequência. Esse ponto de vista faz sentido, porque esse tipo de metodologia exige que todas as definições do projeto sejam muito bem definidas em seu início. Porém, é importante falarmos que essa escolha pode variar bastante, pois diversos outros pontos devem ser avaliados.

Hoje, essa ideia de testar apenas no final do ciclo já passa a ser vista como uma ação onerosa para o projeto, pois, após muitos relatos de experiência no ramo, deduziu-se que é mais barato resolver algum problema no início do projeto do que em suas etapas finais. Então, continuando com esse processo evolutivo das metodologias, chegamos à prática denominada iterativa e incremental.

Essa linha do tempo e de evolução das metodologias aconteceu de forma paralela ao progresso da sociedade, então, a necessidade de se lidar com mudanças, tanto do projeto em si quanto das tecnologias, trouxe, ao profissional da área, certa necessidade de lidar com esse tipo de situação. Pense que, desde a criação dos primeiros softwares, tudo mudou, e após a década de 1990, esse processo se intensificou devido à globalização e, consequentemente, ao compartilhamento de informações, que trouxe novas e desafiadoras situações para as equipes de desenvolvimento.

**EXEMPLIFICANDO**

Algumas metodologias ditas como “tradicionalis” utilizam etapas mais flexíveis de processo de desenvolvimento. Entre elas, é possível mencionar o modelo de prototipagem, modelo V, modelos de processo evolucionário, modelo espiral, entre outros. Por exemplo, nesses modelos, a ideia de iteração e interação foi sendo implementada aos poucos, então, o modelo V trouxe a possibilidade de executar etapas em paralelo, diferentemente do modelo cascata. Já o modelo de prototipação, apesar de poder ser utilizado em complemento a outro modelo, defende a ideia de que, por meio do protótipo, tem-se uma ideia mais clara do sistema. Então, cada modelo apresenta uma evolução do que foi criado anteriormente.

As empresas e os profissionais começaram a listar diversos pontos interessantes. Conforme Ian Sommerville (2011) relata em sua obra, já fazia um tempo que essa percepção estava sendo fortemente notada no mundo do software. Na década de 1980, a IBM começou a utilizar a metodologia incremental em seus processos;

além disso, as próprias linguagens de programação também estavam evoluindo, e na década de 1990, houve o amadurecimento das noções sobre abordagens ágeis. O autor ainda menciona a criação da metodologia de Desenvolvimento de Sistemas Dinâmicos (DSDM, que vem do termo em inglês *dynamic systems development method*), Scrum e Extreme Programming (XP), que são as mais conhecidas e, devido ao seu tempo de criação, as mais utilizadas para se melhorar o fluxo dos projetos.

Como resultado, todos desejavam ter um software com qualidade e que tivesse sido produzido com processos maduros de desenvolvimento. Frente a isso, a metodologia considerada ágil se diferencia da metodologia tradicional por permitir a flexibilidade de suas ações durante o ciclo do software e as mudanças constantes que podem ocorrer no produto que está sendo desenvolvido; além disso, conforme Fowler (2002, apud PRESSMAN, 2016, p. 70), as etapas de projeto e construção do sistema podem acontecer de forma intercalada, tendo em vista que ocorrem interações a todo instante e que uma parte do software acaba sendo resultado dessas ações.

### OS 4 VALORES DO MANIFESTO ÁGIL

Em 2001, Kent Beck e outros 16 especialistas em desenvolvimento de software desenvolveram o Manifesto para o desenvolvimento ágil de software e declararam os seguintes valores:

Figura 2.1 | Valores do *Scrum*

A) Indivíduos e interações mais que processos e ferramentas. 	B) Colaboração com o cliente mais que negociação de contratos. 
C) Software em funcionamento mais que documentação abrangente. 	D) Responder a mudanças mais que seguir um plano. 

Fonte: Shutterstock; adaptada de Beck *et al.* (2001).

Por meio desses valores, nota-se que o mundo do desenvolvimento estava realmente necessitado de uma repaginada, em que não só os processos adotados para o desenvolvimento de software precisavam de mudanças, mas também a questão cultural das empresas, pois, só assim, essas mudanças poderiam chegar até o modo como as pessoas enxergavam e atuavam dentro de seus projetos. Por

esse motivo, os quatro valores deixam claro que as pessoas são, sim, importantes, independentemente de estarem assumindo um papel dentro da equipe de desenvolvimento ou enquanto clientes.

Nas entrelinhas desses ideais, também fica visível que, ao utilizar a metodologia ágil, um software é ou deve ser produzido com agilidade, no sentido de que será desenvolvido ao longo dos processos apenas o que for necessário e que terá de atender a requisitos de qualidade, ou seja, não vai adiantar querer pular etapas ou trazer a rapidez para dentro do projeto e gerar, como resultado final, um software que se tornará obsoleto.

De acordo com Pressman *et al.* (2016), a agilidade é uma ação aplicável a qualquer projeto de software, porém, para que seus objetivos sejam alcançados, o projeto deve ser elaborado de forma que a equipe de desenvolvimento possa fazer um alinhamento entre as tarefas, bem como eliminar o que não é essencial ao projeto, ou que atrapalha a sua fluidez. Ainda sob o ponto de vista do autor, a estratégia de entrega incremental consegue dar retorno ao cliente de forma mais rápida, além de oferecer um software operacional.

Como complemento aos quatro valores do Manifesto Ágil, criado por Beck *et al.* e citados anteriormente, foram desenvolvidos, também, doze princípios que ajudam a fomentar melhor essas ideias:

- Nossa maior prioridade é satisfazer o cliente por meio da entrega contínua e adiantada de software com valor agregado.
- Mudanças nos requisitos são bem-vindas, mesmo tardivamente, no desenvolvimento. Processos ágeis tiram vantagem das mudanças visando à vantagem competitiva para o cliente.
- Entregar, frequentemente, software funcionando, de poucas semanas a poucos meses, preferindo a menor escala de tempo.
- Pessoas de negócio e desenvolvedores devem trabalhar diariamente em conjunto durante todo o projeto.
- Construa projetos em torno de indivíduos motivados. Dê a eles o ambiente e o suporte necessários e confie neles a realização do trabalho.
- O método mais eficiente e eficaz de transmitir informações para e entre uma equipe de desenvolvimento é por meio de conversa face a face.
- Software funcionando é a medida primária de progresso.
- Os processos ágeis promovem desenvolvimento sustentável. Os patrocinadores, desenvolvedores e usuários devem ser capazes de manter um ritmo constante indefinidamente.
- Contínua atenção à excelência técnica e ao bom design aumenta a agilidade.
- Simplicidade - a arte de maximizar a quantidade de trabalho não realizado é essencial.

- As melhores arquiteturas, os melhores requisitos e designs emergem de equipes organizadas.
- Em intervalos regulares, a equipe reflete a respeito de como se tornar mais eficaz e, então, refina e ajusta seu comportamento.

Ao aplicar os princípios do manifesto, o objetivo é agregar valor ao negócio do cliente, então, por exemplo, a partir do momento em que você demonstra ao cliente que ele faz parte da equipe e está inserido no projeto, além de atender aos princípios, você o deixa mais satisfeito com os processos, tendo em vista que ele conseguirá visualizar o valor do seu investimento.

Com essa participação, é possível mensurar se as funcionalidades que estão sendo desenvolvidas realmente atendem às necessidades reais do cliente. Muitas vezes, no início do projeto, ele não consegue expor completamente o que deseja.

Ao mesmo tempo, é possível observar o progresso da equipe para atingir o objetivo final, que é o de finalizar o software, e então, por meio da iteração e interação, notar se o que foi definido no início realmente foi o correto. Além disso, outro ponto positivo é a adaptação às mudanças, que, apesar de causar, a princípio, um pouco de desconforto no planejamento, tem seu lado positivo ao trazer as certezas sobre o projeto.

Por exemplo, caso a implementação da leitura de código de barras no sistema, durante o processo, evoluia para a leitura de QRcode, por meio dos ciclos, será possível realizar um novo planejamento para implementação dessa nova funcionalidade ou mensurar se será mais ágil reutilizar o que já foi produzido e adaptá-lo à nova realidade.

Ao mesmo tempo, perceba que essa mudança, apesar de modificar os planos, traz valor ao negócio do cliente, pois seu sistema será finalizado de acordo com as demandas do usuário, que acompanha as atualizações tecnológicas.

Novamente, você pode perceber que os criadores do manifesto enfatizavam, de forma detalhada, tudo aquilo que defendiam em seus quatro valores iniciais, logo, seguindo a linha do tempo, da evolução das metodologias, após terem sido bastante utilizadas, as metodologias tradicionais, a partir desse momento, serviram como base para as novas que estavam surgindo.

Para Pressman et al. (2016, p. 66):

“

A engenharia de software ágil combina filosofia com um conjunto de princípios de desenvolvimento. A filosofia defende a satisfação do cliente e a entrega incremental antecipada; equipes de projeto pequenas e altamente motivadas; métodos informais; artefatos de engenharia de software mínimos; e, acima de tudo, simplicidade no desenvolvimento geral. Os princípios de desenvolvimento priorizam a entrega mais do que a análise e o projeto (embora essas atividades não sejam desencorajadas); também priorizam a comunicação ativa e contínua entre desenvolvedores e clientes.

A princípio, um dos pontos a se pensar quanto à decisão de utilizar novas metodologias é como a mudança ou adaptação da cultura organizacional acontecerá. De acordo com Cohn (2011, p. 29), “após identificar essas pequenas fraquezas, no entanto, ainda ficamos com o problema de como eliminá-las. É difícil (e com frequência impossível) prever exatamente como as pessoas responderão às várias pequenas mudanças que serão necessárias no percurso de se tornar ágil.”

o  
Ver anotações

## | PRINCIPAIS METODOLOGIAS ÁGEIS

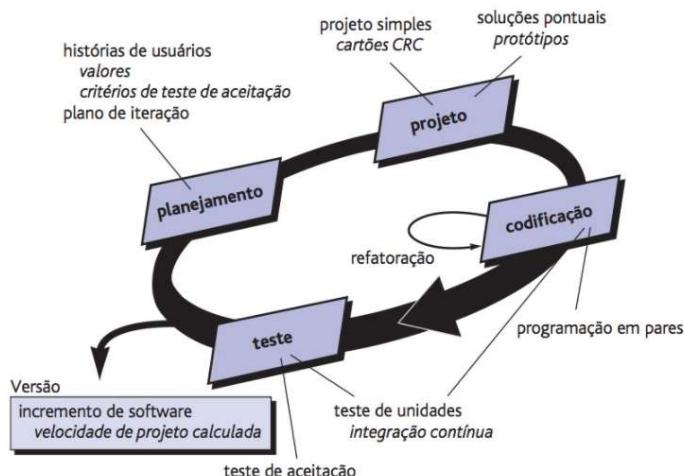
Existem diversas metodologias consideradas ágeis, até mesmo porque, como mencionamos mais acima, a tendência é que haja sempre uma evolução do que já foi testado anteriormente, logo, com as metodologias, não seria diferente.

Entre as principais e mais utilizadas, é possível destacar:

## | PROGRAMAÇÃO EXTREMA (XP)

A Programação Extrema ou, como é popularmente chamada, a “XP”, que é uma abreviação de seu nome em inglês, *Extreme programming*. Pressman (2011) ratifica a importância da metodologia XP devido à possibilidade de avaliações imediatas perante o contexto do projeto, que envolve o planejamento das atividades e a organização da equipe. Ela possui quatro atividades principais (planejamento, projeto, codificação e teste) além de algumas outras subatividades.

Figura 2.2 | O processo da *Extreme Programming (XP)*



Fonte: Pressman *et al.* (2016, p. 72).

Como é possível observar, as subatividades da metodologia são responsáveis por complementar todo o ciclo. O **planejamento** é o primeiro passo para qualquer atividade de sucesso, pois a definição dos objetivos e das etapas que serão executadas facilita bastante na execução das demais atividades. Dessa forma, esse é o momento em que toda a equipe listará os requisitos que aquele sistema deverá ter para atender às necessidades do cliente, por isso, é de extrema relevância. Em seguida, o desafio será definir qual o tempo estimado para solucionar todos os desejos do cliente.

Para facilitar, a equipe divide essas requisições em cartões, sendo que, cada um contará uma “história do usuário”. A divisão das tarefas ocorre devido à necessidade de se atender aos valores trazidos pela filosofia do Manifesto Ágil, que é a de dar prioridade aos indivíduos e às interações, dessa forma, o cliente terá acesso às funcionalidades do sistema no decorrer de todo o processo.

Outro aspecto bastante marcante da XP é a simplicidade no **projeto**, pensando-se apenas no presente e não estimando ações futuras. Além disso, a metodologia defende o desenvolvimento "*test-first*", em que testes automatizados são realizados com o objetivo de avaliar as novas funcionalidades. Seguindo uma perspectiva macro de todo o processo XP, é possível mencionar a etapa de **codificação**, como mostra a Figura 2.1, sendo composta pela refatoração e a programação em pares, que são ações bastante importantes, já que, por meio delas, a etapa de implementação pode ganhar mais qualidade, uma vez que a refatoração contínua traz melhorias ao código e a programação em pares traz a contribuição dos membros da equipe para o trabalho naquele momento. Tendo em vista que a propriedade do que está sendo desenvolvido é de todos da equipe, então, caso seja necessário mudar algo, aquele que for mais apto, conforme suas habilidades, poderá realizar a alteração, e claro, tudo dentro do escopo do projeto.

Citando novamente as premissas da metodologia ágil, a entrega contínua exige que a metodologia que quer atender ao que a filosofia defende deve realizar ou se adaptar à integração contínua do sistema. E como os integrantes da equipe, ou seja, os *stakeholders* são de extrema relevância, tudo isso deve ocorrer levando-se em consideração um ritmo de trabalho sustentável, em que se consiga produzir mais, com qualidade, mas sem ultrapassar as horas predefinidas de trabalho para a equipe.

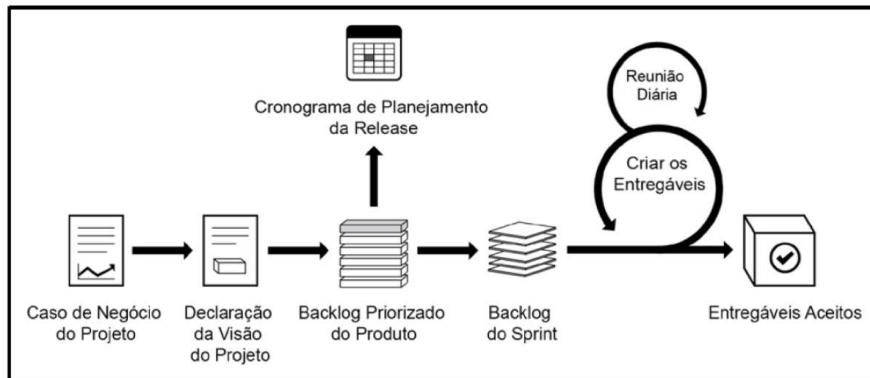
Após a realização dos **testes**, as funcionalidades que foram definidas da etapa de planejamento serão entregues ao cliente (geralmente, elas são chamadas de incrementos de software) e todo o ciclo volta a girar novamente, um novo planejamento ocorre e assim por diante.

#### ■ METODOLOGIA SCRUM

A metodologia *Scrum* recebeu esse nome tendo-se como base as regras estabelecidas nas partidas de Rugby. Ela foi desenvolvida por Jeff Sutherland, na década de 1990.

Apesar de receber muitas vezes um nome em destaque, o Scrum é desenvolvido e atualizado ao longo dos anos por diversos estudiosos da área. O Scrum apresenta em suas práticas diversos padrões relacionados aos processos de software, que, ao longo do tempo, trouxeram resultados positivos e projetos mais eficazes (PRESSMAN, 2016).

Figura 2.3 | Uti - Fluxo do processo *Scrum*



Fonte: SCRUMstudy (2016, p. 2).

Conforme Sommerville (2011), existem três fases que caracterizam a metodologia, e a primeira delas é a de planejamento, que define os objetivos do projeto. E como pode ser visto na Figura 2.2, o ciclo se inicia com o entendimento do negócio do cliente, representado na figura pelo **"caso de negócio do projeto"**.

Posteriormente, na **"declaração da visão do projeto"**, é possível obter os objetivos, que são mencionados por Sommerville como sendo uma das etapas do *Scrum*.

A segunda fase é composta por várias *sprints*, e a *sprint* caracteriza o ciclo do *Scrum*; dentro dela, acontecem várias outras subatividades, como a reunião entre os *stakeholders*, ou seja, com todos os envolvidos no projeto, desde a equipe de desenvolvimento até o cliente ou seu representante. Posteriormente, listas são criadas com o objetivo de se documentar as necessidades reais do cliente e até mesmo para mensurar o tempo que o processo todo levará para ser concluído. O nome dessas listas é *backlog*, que pode ser do produto em geral ou da *sprint*, ou seja, do ciclo.

Em cada ciclo, é possível ter também a definição de itens entregáveis e outras cerimônias, como a reunião diária, em que cada membro da equipe relata como foi o seu dia de trabalho, o que está fazendo, se houve alguma dificuldade e qual será sua próxima atividade.

Por meio dessa perspectiva interativa e incremental, o *Scrum* passa a demonstrar que atende aos princípios norteados pelo Manifesto Ágil cujo foco é o software em funcionamento ao final do processo.

Na prática, o cliente vai tendo acesso ao seu sistema antes mesmo de ele ser concluído; o que ocorre é que suas funcionalidades, ou seja, partes menores vão ficando prontas aos poucos, logo, em um sistema de compras e-commerce, por exemplo, a cada iteração, algo pode ser entregue. Vamos supor que na primeira *sprint* houve a finalização da funcionalidade de cadastro, na próxima Sprint, será entregue a funcionalidade de realizar vendas e assim por diante. Ao final do processo, o cliente terá seu sistema pronto por completo, porém terá participado de todo o ciclo e tido a oportunidade de ir aperfeiçoando as decisões tomadas no início, trazendo mais qualidade e valor ao seu sistema.

## | MÉTODO DE DESENVOLVIMENTO DE SISTEMAS DINÂMICOS (DSDM)

A tendência é que novas metodologias sejam criadas para suprir as demandas que vão surgindo, com isso, a DSDM surge com o intuito de atender a restrições relacionadas ao prazo. A sua ideia principal é dar agilidade ao processo por meio do uso de protótipos que vão sendo incrementados à medida que o projeto avança. Conforme Pressman *et al.*(2016), a metodologia apresenta o uso de iterações em que o trabalho dedicado é referente apenas às funcionalidades do ciclo. Posteriormente, é possível inserir mais especificações após o entendimento dos requisitos de negócio.

Por compartilhar de uma perspectiva semelhante às demais metodologias citadas até então, o DSDM define três diferentes ciclos iterativos (Pressman, 2016):

- Iteração de modelos funcionais, trazendo o objetivo de demonstrar protótipos aos clientes para obter requisitos adicionais ao produto por meio do feedback, ou seja, os clientes poderão ter uma ideia de como será o produto final por meio do uso dos protótipos. É importante lembrar que a metodologia tem como ideia principal o uso dos protótipos em uma perspectiva evolutiva, ou seja, em que ele possa ir se aperfeiçoando ao longo do processo.
- Iteração de projeto e desenvolvimento, em que os protótipos possam participar de dois processos ao mesmo tempo, sendo um de iteração de modelos funcionais e outro de iteração de projeto e desenvolvimento, isso ocorre porque os modelos funcionais visam à capacitação de todos diante do negócio do cliente.
- Por fim, existe o ciclo de implementação, sendo possível utilizar a última versão do incremento do software gerado pelas iterações durante o processo. Diante disso, fique atento, pois um protótipo, a depender de seu desenvolvimento e sua evolução, pode não conter todos os detalhes que a aplicação deve ter.
- Outro exemplo que pode ser citado é o da Modelagem ágil, que abrange um ponto de vista em que o entendimento do escopo do projeto como um todo deve ser compreendido da melhor maneira possível pela equipe de desenvolvimento.

## | MODELAGEM ÁGIL

Conforme Ambler (2002 *apud* Pressman, 2016), a modelagem ágil atende aos valores do Manifesto Ágil, para isso, são listados alguns princípios considerados básicos e suplementares, os quais podem ser vistos abaixo:

- Modelar com um objetivo, pois quando se tem um objetivo, fica mais simples a decisão acerca das notações, dos softwares e dos detalhes que precisarão ser utilizados.
- Usar modelos diversos. (Entende-se que cada modelo pode contribuir de alguma maneira para o projeto.) Além disso, sob esse ponto de vista, a filosofia da modelagem ágil defende que os pontos fortes e fracos das ferramentas que serão utilizadas devem ser elencados.

o

Ver anotações

- Construir modelos que agregam valor em vez de mudanças constantes de ação dentro do projeto.
- Trazer, por meio da modelagem, informações relevantes.

**REFLITA**

Algumas empresas de tecnologia têm anos de existência, mas ainda que não tenham um tempo tão considerável, como 20, 30 anos, as empresas definem alguns processos, entre eles, o de desenvolvimento de software. Com o passar dos anos, surgem novas tecnologias, metodologias, ferramentas. Diante disso, reflita sobre como deve se dar a transição de uma novidade para outra em todos os aspectos da empresa, desde a questão cultural dos colaboradores ao uso e à implementação das novas tecnologias, as quais evoluem em um pequeno espaço de tempo.

Caso você seja contratado para implantar a metodologia ágil dentro da empresa, quais ações você colocaria em prática para realizar essa mudança organizacional e como avaliaria os pontos positivos que a metodologia ágil pode trazer aos processos já existentes?

o  
Ver anotações**| FERRAMENTAS AUTOMATIZADAS**

Tendo como base os conceitos adquiridos sobre a agilidade, é possível compreender que o objetivo é adaptar-se para conseguir acompanhar a evolução das coisas.

Conforme Pressman (2016), o empirismo está relacionado ao desenvolvimento de regulamentações acerca do contexto tecnológico, além disso, existe a criação de ferramentas automatizadas cujo uso contínuo pode trazer um melhor uso dos recursos, tendo em vista o aperfeiçoamento das ações.

Para uma melhor compreensão, é possível fazer uma analogia com o *Scrum*, que possui três pilares principais: transparência, inspeção e adaptação. As práticas do *Scrum* são baseadas nesses elementos, e esse termo se encaixa na história da metodologia, porque, assim como as demais, foi desenvolvida como base nas práticas de diversos estudiosos ao longo dos anos.

**| EVOLUÇÃO E AMADURECIMENTO DOS PROCESSOS**

Como você pode observar onde se aplicam a evolução e o amadurecimento dos processos? Perceba que, por meio das reuniões diárias (*daily scrum*), por exemplo, é possível mensurar como anda o progresso do projeto; então, quando o time *scrum* relata o que fez no dia, se surgiu ou não alguma dificuldade durante a atividade e o que está previsto para ser feito no dia seguinte, é possível caracterizar certo processo evolutivo, bem como compreender o projeto por meio da transparência dos fatos, dos relatos, e adaptá-lo, uma vez que, ao final de cada *sprint*, uma nova *backlog* terá que ser definida e assim por diante.

O conceito de empirismo é visto dentro desse contexto apresentado quando, por meio dos ciclos, ou seja, das *sprints*, ocorre a automatização das práticas dentro do ciclo do projeto, trazendo, com isso, qualidade ao projeto e ao software, que será o

resultado final dessas ações.

Diante da diversidade de processos de software que podem ser utilizados.

Note que todos os artefatos, as cerimônias e os conceitos aplicados pelas metodologias podem ser complementados com a necessidade de auto-organização, ações colaborativas entre os integrantes da equipe, priorização com base em valor e a iteração incremental. Além disso, é importante salientarmos que as metodologias podem ser adaptadas e utilizadas conjuntamente; o importante é que suas diretrizes atendam às demandas do cliente e, consequentemente, de todos os demais envolvidos no projeto.

Ver anotações

## FAÇA VALER A PENA

### Questão 1

A engenharia de software apresenta um arcabouço robusto de metodologias e conceitos voltados ao ciclo de vida de um software. Diante desse contexto, algumas metodologias de desenvolvimento surgiram ao longo dos anos, sendo, as mais atuais, as chamadas metodologias ágeis.

Sobre o conceito de metodologia ágil, assinale a alternativa correta:

- a. O conceito está relacionado à velocidade com que um software fica pronto. Os projetos levam em torno de uma semana para serem finalizados, conforme o padrão.
- b. A metodologia ágil surgiu antes da criação das metodologias tradicionais, que foram baseadas em seus princípios.
- c. O uso da metodologia ágil visa à redução da equipe de desenvolvimento, limitando a equipe ao desenvolvedor e ao analista de requisitos.
- d. A filosofia ágil fomenta que o software em pleno funcionamento é mais importante que o desenvolvimento de documentação.
- e. A filosofia ágil defende que uma etapa do processo só deve ser executada quando sua etapa anterior terminou...

### Questão 2

De acordo com Pressman *et al.* (2016, p.31), “quando se elabora um produto ou sistema, é importante seguir uma série de passos previsíveis – um roteiro que ajude a criar um resultado de alta qualidade e dentro do prazo estabelecido. O roteiro é denominado processo de software”.

Considerando as informações apresentadas, analise as afirmativas a seguir:

- I. Um modelo de processo de software é definido com o objetivo de organizar os processos executados por uma equipe durante o ciclo de vida de um software.
- II. A definição do modelo de processo de software deve ser padronizada a todos os projetos executados pela empresa, sendo inserida apenas a marca do cliente.
- III. Os processos de software podem ser considerados tradicionais, quando seguem padrões predefinidos no início do projeto, e ágeis, quando estão abertos a mudanças.
- IV. O modelo ágil cascata tem como premissa a execução de todas as suas etapas em paralelo, trazendo rapidez e eficiência a todo o processo de desenvolvimento.

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

a. I e III, apenas.

b. I e II, apenas.

c. II e III, apenas.

d. I, II e IV, apenas.

e. II, III e IV, apenas.

Ver anotações

### Questão 3

Um software é um artefato complexo o suficiente para demandar, de seus desenvolvedores, habilidades diversas. Além disso, a multidisciplinaridade que o caracteriza demanda que processos e metodologias de desenvolvimento sejam desenvolvidos; como exemplo, podemos mencionar a existência dos processos tidos como tradicionais e os ágeis.

Tomando como referência as diferentes filosofias que norteiam os processos de software, julgue cada uma das afirmativas a seguir como (V) verdadeira ou (F) falsa.

- ( ) A comunicação caracteriza a primeira etapa básica de um projeto de software, fazendo com que seja dado início às etapas que antecedem o planejamento.
- ( ) A metodologia ágil tem como principal característica o desenvolvimento linear de suas etapas, trazendo conexão entre todas.
- ( ) A modelagem é realizada antes da comunicação e do planejamento, em que a equipe de desenvolvimento mostra suas ideias ao cliente.
- ( ) A especificação do software deve ser capaz de listar suas principais funcionalidades e restrições.

Assinale a alternativa que apresenta a sequência CORRETA.

a. V – F – F – F.

b. V – F – F – V.

c. F – F – V – F.

d. F – V – F – V.

e. V – V – F – V.

### REFERÊNCIAS

BECK, K. et al. **Manifesto para desenvolvimento ágil de software**. 2001.

Disponível em: <https://bit.ly/3qviusu>. Acesso em: 30 dez. 2020.

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

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.

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

o  
Ver anotações