

# Projeto de Software

Daniela Teresa Rossignoli Uebele

## **Unidade-1 Seção-1**

### **CONHECENDO A DISCIPLINA**

Caro aluno, na história recente dos projetos de software, assistimos aos momentos de sucesso e aos fracassos notáveis; aprendemos com eles e fomos estabelecendo processos capazes de maximizar nossas chances de acertar.

Para as unidades que se seguem, estudaremos os fundamentos da gestão de projetos em TI, conhecendo as técnicas de gestão de projetos normatizadas para o planejamento, a execução e o monitoramento de projetos de Software. Os projetos de software não são uma ciência exata, afinal, não existe uma maneira única e garantida de se chegar a um resultado. Por isso, nosso desafio é ainda maior.

Uma das maneiras de se conseguir um resultado é garantindo a evolução do projeto durante suas diversas fases, respeitando prazos e recursos humanos e financeiros disponíveis.

Na Unidade 1 do livro, estudaremos uma introdução à análise de projeto de sistemas, o gerenciamento de projetos em cascata e as fases de processos para o gerenciamento de projeto.

Diante disso, o gerente de projeto é o profissional indispensável no trabalho de transformar, com sucesso, ideias em realidade; para tanto, ele conta com ferramentas poderosas e com apoio profissional para exercer sua atividade. Mediante esse fato, tem-se o motivo da existência do PMI. Tudo que é necessário para

realizar a gerência de um projeto, buscando resultados positivos, está no principal documento de referência do PMI, o PMBOK (Project Management Body of Knowledge).

Em seguida, na Unidade 2, continuaremos falando dos gerenciamentos ágeis, das definições e dos valores ágeis, dos tipos de metodologia e empiricismo, bem como estudaremos o Scrum, abordando a técnica de Design Thinking em projetos de TI, o Minimum Viable Product (MVP) e os conceitos de pivotagem. Já na Unidade 3, prosseguiremos discutindo a respeito das gestões de risco e de qualidade, bem como das questões referentes à documentação. Finalizaremos, então, com a Unidade 4, tratando do modelo sequencial e ágil, verificando as diferenças e semelhanças na concepção e no desenvolvimento de projetos, conhecendo ferramentas viáveis tanto para a gestão do modelo ágil como sequencial e os seus respectivos ciclos de vida. E se tratando de ciclo de vida, você conhecerá importantes atividades da manutenção, do planejamento de releases e do pós-lançamento.

Os desafios de adquirir e dominar tanto conhecimento são realmente vastos, mas, com certeza, a prática lhe oferecerá todo o conhecimento que precisa para evoluir e ter sucesso profissional. PERSISTA!

## Seção-2

# NÃO PODE FALTAR

## INTRODUÇÃO À ANÁLISE E AO PROJETO DE SISTEMAS



### **CONVITE AO ESTUDO**

No atual universo corporativo, é fundamental que os profissionais da área de Tecnologia da Informação (TI) tenham conhecimento das inúmeras técnicas de gestão de projetos (preferencialmente as normatizadas) e se aprofundem nas que foram criadas ou adaptadas para o setor de tecnologia da informação e comunicação. Com as grandes transformações digitais, é um dos setores que têm apresentado maior crescimento e um desenvolvimento constante, exigindo o aprendizado de novas habilidades e aptidões e de profissionais com perfil inovador para cargos que sequer existiam até então.

As organizações no Brasil vivem uma era de reconstrução. O mercado passa por momentos de instabilidade, e é devido a essa variação que a área de maior destaque é a de TI. Investir em tecnologia não é voltar-se apenas para a inovação, mas é uma forma de sobrevivência e de se tornar ou permanecer competitivo (ZIMMERMANN, 2001).

Para contribuir com o planejamento e a execução de forma mais assertiva, independentemente do prazo, apresentando uma visão ampla para a condução do negócio, importante para a gestão de um projeto com conceitos bem fundamentados, é preciso conhecer e compreender modelos de projetos de software, podendo ser este o diferencial para obtenção ou não de sucesso no projeto.

Na primeira seção desta unidade, você estudará a introdução à análise e ao projeto de sistemas, em que serão abordados os conceitos de projetos, a importância da gestão de projetos, a formação da equipe (participantes, matriz de responsabilidades e plano de comunicação), os problemas na gestão, os conflitos, as leis de Murphy e os mitos relacionados ao gerenciamento de projetos de software.

Na segunda seção, partiremos para o gerenciamento de projetos em cascata, em que os assuntos estarão voltados para os Fundamentos de Processos para gestão de projetos de TI, o Ciclo de Vida do projeto, o Gerenciamento de Projetos e os Fundamentos de ITIL, COBIT, PMI e PMBOK.

Na terceira e última seção, você vai estudar as fases dos processos para gerenciamento de projetos, sendo eles: o processo de iniciação, o processo de planejamento, o processo de execução, os processos de monitoramento, o controle e o encerramento.

Pois bem! Acredito que você terá a oportunidade de aprender e aplicar os conhecimentos aqui apresentados. Sucesso e bons estudos!

## **PRATICAR PARA APRENDER**

Todo início de uma nova fase vem recheado de desafios, de incertezas, de expectativas e, com certeza, de muito aprendizado a ser adquirido. Diante dessa situação, vem a pergunta-chave: seguir ou recuar? Afinal, você adentrará em um mundo desconhecido, e isso, com certeza, é um tanto assustador. Você já se pegou pensando em desistir? Ficou analisando as inúmeras possibilidades do sim e do não? Pensou em todas as pessoas que poderiam lhe ajudar?

Com isso, podemos perceber que é natural de nós, seres humanos, fazer uma análise de mundo e do nosso mundo em especial. Caso você conheça pessoas que primeiro agem para depois verificar as possibilidades, provavelmente, já deve ter notado que a quantidade de insucessos é maior.

O desejo de acertar, sem dúvida, é o que move a humanidade, mas esses acertos estão baseados em sonhos, em desejos de se adquirir novas habilidades e conhecimentos. Mas para que as realizações ocorram, é necessário um planejamento de tarefas que constituem um projeto. Seguindo esse raciocínio, os projetos podem ser classificados em pessoal, profissional e social.

Mediante todo esse contexto filosófico, fica claro que, se temos pequenos projetos e buscamos meios de torná-los realidade, as empresas também são movidas por esse mesmo desejo, como conseguir lançar novos produtos e/ou serviços que visem a aumentar o número de consumidores e fidelizar os que já fazem uso dos produtos. Isso ocorre por meio dos projetos. Já havia pensado em tudo isso?

Quando você inicia uma nova etapa em sua vida, ingressando como funcionário de uma organização, o sucesso dessa organização passa a ser também um pedacinho do seu sucesso pessoal, mas para que isso ocorra, são necessárias inúmeras definições prévias.

Todos criamos expectativas e elas estão baseadas no prazo de conclusão do projeto, nas funcionalidades que se espera ter e na

qualidade de cada uma delas, e é importante lembrarmos que a expectativa não é exclusividade de um único integrante, mas de todas as pessoas envolvidas, principalmente daquelas que estarão em contato, diretamente, com o software, já que são elas as mais afetadas pelo sucesso ou insucesso de um projeto.

Mas e se não cumprir o prazo? E se as funções do sistema apresentarem falhas no seu funcionamento? Caso um dos itens anteriores ocorra ou mesmo os dois, quanto custaria ajustar o prazo ou consertar as falhas? Como podemos evitar essas situações desagradáveis? Conflitos, mitos e problemas de gestão andam lado a lado nesses desafios!

A empresa de desenvolvimento de projetos de sistemas pela qual você foi contratado alocou você para ingressar como membro de uma equipe de desenvolvimento de sistemas, em que terá acesso a inúmeras informações referentes ao departamento de gestão da própria empresa de TI, e muitas dessas informações estarão dentro de um processo de sigilo.

Você trabalhará com um módulo novo, que será implantando em seis meses. Esse módulo visa a gerar um controle detalhado de cada um dos desenvolvedores contratados e alocados nos inúmeros projetos que são desenvolvidos em loco na contratante.

Entre as inúmeras informações pessoais dos profissionais, serão inseridas informações como o projeto em que está envolvido, quanto tempo ficará alocado em cada um dos projetos, se cumpriu antes, depois ou no tempo previsto para a atividade, a sua área de atuação, os cursos, os treinamentos, as certificações concluídas e as datas de validade dessas certificações.

Todo esse módulo a ser desenvolvido terá como principal objetivo avaliar o andamento de cada um dos projetos, bem como o profissional inserido nas equipes. Na sua equipe, você terá contato com inúmeras áreas de conhecimento a serem trabalhadas como: segurança da informação, estruturação de dados, análise de requisitos e, como sempre, um dos maiores desafios: o cronograma.

A empresa em que você foi contratado trabalha com gestão orientada para o mercado.

Portanto, para esse projeto, um dos desafios é definir quais as habilidades e características requeridas para os profissionais a serem alocados ou contratados para o desenvolvimento desse módulo do sistema, bem como pensar no gerente de projeto.

É importante definir qual a comunicação adequada entre todos os *stakeholders*; qual a proposta para o processo de comunicação e planejar as ações mediante a possibilidade de ocorrer conflitos durante o desenvolvimento do sistema; quais habilidades são importantes para se conduzir essas situações e evitar prejuízos; e quais as evidências que podem ser observadas para indicar o início dos conflitos, considerando as incertezas e todo o conhecimento estudado nesta unidade. Seria muito interessante pensar em como estruturar esse projeto, assim, terá ideia do desafio que enfrentará pela frente, no seu dia a dia de trabalho.

Pois é, justamente para evitar problemas como esses e tantos outros que o estudo da gestão de projetos se faz importante.

#### **CONCEITO-CHAVE**

#### **CONCEITOS E IMPORTÂNCIA DA GESTÃO DE PROJETOS**

Sempre que você for estudar algum assunto, considere relevante uma análise das origens e principalmente sua trajetória. Para começar, vamos entender o significado da palavra projeto, que vem do latim, *projectus*, e significa “Plano, planejamento que se faz com a intenção de realizar ou desenvolver alguma coisa.” (PROJETO, 2020). Ampliando para uma aplicação, projeto significa elaborar um plano detalhado de um empreendimento a ser realizado, não ficando apenas na ideia. Deve ser considerado, também, o período em que as tarefas estão sendo executadas, ou seja, inclui a utilização de recursos físicos, humanos e financeiros.

Ao procurarmos, na história, projetos que fizeram a diferença, deparamo-nos com inúmeras obras de relevância que mostram que “projetar” faz parte da natureza humana. Exemplos notáveis como as Pirâmides do Egito, em 3.500 a.C., a Muralha da China, 1.046 a.C., a construção dos formidáveis castelos do século XII, a Revolução Industrial no século XIX, ou mesmo a Usina de Itaipu. Em todo lugar que olhamos, podemos observar a realização de grandes projetos. Mas o que todos eles têm em comum? Como sabemos que são projetos? (CARVALHO, 2018).

Todo e qualquer projeto é definido por três aspectos fundamentais, a saber: tempo, planejamento e, talvez o mais importante, ser único, uma vez que terá características exclusivas.

Um projeto é temporário. Isso quer dizer que possui um tempo bem definido para sua execução, com começo, meio e fim, e é isso que difere projeto de rotina/produção. Por ser temporário, ele precisa ser planejado, necessitando de uma preparação e de um planejamento prévio, de forma a definir quais, quantas e quando as atividades devem ser controladas e coordenadas.

#### **ASSIMILE**

Uma atividade aparentemente idêntica pode ser considerada como projeto ou não, a depender do contexto. Por exemplo: uma viagem internacional. Se estivermos falando de um cidadão planejando suas férias, temos, claramente, um projeto; no entanto, se estivermos nos referindo a um empresário viajando a negócios, essa viagem passa a ser considerada um planejamento de rotina.

Vamos explorar melhor os conceitos de projetos procurando as definições segundo autores e organizações renomadas e em períodos distintos.

Segundo PMI, um dos mais renomados e respeitados institutos de gerenciamento de projeto sem fins lucrativos, que gerou o guia PMBOK: “projeto é um esforço temporário empreendido para criar um produto, serviço ou resultado único” (2017, p. 4) e define boas práticas como “a aplicação do conhecimento, habilidades,

ferramentas e técnicas podem aumentar as chances de sucesso de muitos projetos em entregar valor de negócio e resultados esperado” (2017, p. 2).

Kerzner (2002, p. 15), define: “Projeto é um empreendimento com o objetivo bem definido, que consome recursos e opera sob pressão de prazos, custo e qualidade.” Já a norma ISO 10006, (1997, p. 2), define projeto como: “um processo único, consistindo em um grupo de atividades coordenadas e controladas com datas para início e término, empreendido para o alcance de um objetivo conforme requisitos específicos, incluindo limitações de tempo, custo e recursos”. Por fim, Tuman afirma que “um projeto é uma organização de pessoas dedicadas atingir um propósito e objetivo específico” (1983, p. 26).

#### REFLITA

Estamos, sempre, realizando inúmeros projetos, dos mais variados portes e com as mais diversas relevâncias. Isso significa que você já se envolveu em projetos sem nem ao menos perceber, quer ver? Você já realizou alguma viagem? Já fez algum curso? Organizou alguma festa? Certamente, todos temos algum desejo a alcançar, não é mesmo? Já se pegou refletindo sobre qual será a próxima etapa para alcançar um objetivo? Sobre qual a ferramenta mais adequada para isso ou o conhecimento que precisa adquirir? Qual o melhor caminho? Quem poderá ajudar você?

#### GERENCIAMENTO DE PROJETO

Gerenciar um projeto ainda é um enorme desafio, pois existem poucos profissionais preparados para efetuar um bom planejamento, de forma a antecipar o futuro para o agora. Um projeto excede um conjunto ordenado de tarefas com um objetivo específico; ele não deve ser elaborado de forma exclusivamente técnica, ou seja, dentro das normas definidas para cada área, mas também é necessário ter elementos que venham a prever resultados, minimizar os riscos de fracasso e maximizar as chances de sucesso.

Quando estamos envolvidos em qualquer tipo de projeto, buscamos alcançar eficiência e eficácia.

A propósito, você conhece a diferença entre eficiência e eficácia? Eficiência se refere às qualidades dos recursos que estão sendo usados para se alcançar o resultado, ao conhecimento técnico, às melhores ferramentas, tudo aquilo que contribui para o alcance do objetivo, ainda que não garanta o resultado. Já a eficácia visa ao resultado, então, podemos entender que, para se ter um projeto concluído com sucesso, é necessário contar com ambas: eficiência e eficácia (CARVALHO JUNIOR, 2012).

E por que, então, deveríamos nos preocupar em aprender a gerenciar projetos? Afinal de contas, a história mostra que projetar e realizar projetos é algo nato do ser humano. Todos os projetos citados tiveram sua conclusão, no entanto, nem todos foram concluídos dentro do período planejado ou entregues com a qualidade esperada; além disso, provavelmente, muitos ultrapassaram as previsões de gastos.

Com o passar das décadas, com a evolução industrial e, hoje, com a evolução digital, muitos softwares e aplicativos foram desenvolvidos com inúmeros propósitos. Alguns foram bem sucedidos, outros nem tanto.

O que queremos é melhorar nossas chances de sucesso na produção de softwares, e estamos considerando toda a expertise em projetos vitoriosos, passando a aplicá-los na análise de um sistema, adaptando ou criando formas de acompanhar e gerenciar aspectos fundamentais, como:

- Requisitos baseados nos clientes.
- Prazos (data de início e encerramento), custos e recursos específicos.
- Atividades e operações coordenadas e controladas.
- Pressões de macro e microambientes.

Quando examinamos essa lista, reparamos que todos os tópicos parecem bastante óbvios, mas são surpreendentes as motivações que podemos obter com um bom gerenciamento de projeto:

- Requisitos baseados nos clientes.
- Prazos (data de início e encerramento), custos e recursos específicos.
- Atividades e operações coordenadas e controladas.
- Pressões de macro e microambientes.

Quando examinamos essa lista, reparamos que todos os tópicos parecem bastante óbvios, mas são surpreendentes as motivações que podemos obter com um bom gerenciamento de projeto:

- Antecipar riscos, evitando surpresas com mudanças de requisitos ou perdas de recursos.
- Facilitar o processo de revisão do projeto a partir do detalhamento das atividades.
- Agilizar as tomadas de decisões com atividades que podem acontecer ao mesmo tempo.
- Agilizar as tomadas de decisões com atividades que podem acontecer ao mesmo tempo.
- Otimizar a locação de recursos humanos, compreendendo a competência e a aptidão necessárias para cada fase do projeto.

Com todos os aspectos que podem ser geridos de forma clara e objetiva, é possível obter inúmeros benefícios quantitativos e qualitativos a partir do momento que se inicia a utilização de métodos, técnicas e ferramentas de gestão. Esses resultados são tangíveis como:

- Otimização do tempo de realização das tarefas.
- Tomada de decisões mais rápidas, estruturadas e com qualidade.
- Maximização do lucro.
- Minimização de custos.
- Diminuição da burocracia.
- Redução do número de pessoas para a realização das tarefas.
- Maior qualidade e confiança, devido à redução de retrabalho e de falhas.

- Diminuição na troca de recursos humanos e agilidade no processo de implantação das melhores práticas.

Ainda levando em conta alguns aspectos considerados mais abstratos (menos tangíveis), podemos destacar:

- Melhora na coordenação e no acompanhamento da evolução das atividades e operações.
- Rápido aperfeiçoamento dos gerentes.
- Uma relação mais saudável com o cliente.
- Melhor controle dos processos.
- Equipe mais estimulada.
- Engajamento e maior apoio dos integrantes da equipe.
- Diminuição dos conflitos que necessitam do envolvimento da alta administração (CARVALHO, 2018).

Pensando em um cenário altamente competitivo na área de TI, em especial nos projetos de desenvolvimento de sistemas, são notórias as transformações nas estruturas organizacionais. Os empresários observam que suas organizações necessitam de dinamismo para a uma reestruturação rápida, em resposta às necessidades do mercado (KERZNER, 2002), o que leva o processo de formação das equipes de projetos sofrer um impacto direto, conforme a estrutura organizacional.

Identifica-se, atualmente, 3 estruturas:

- **Estrutura funcional:** consiste na divisão de departamento, sendo a mais tradicional.
- **Estrutura projetizada:** é orientada para o mercado.
- **Estrutura matricial:** sendo uma combinação das outras duas, classificada como matricial forte, equilibrada e fraca (COSTA; PEREIRA, 2019).

## **ESTRUTURA**

Vamos falar da “mais comum”, das organizações funcionais com estrutura funcional. A hierarquia rígida é a grande questão desse modelo; existe um superior que coordena de forma direta e bem

definida a sua equipe, e cada departamento é definido por sua especialização, conforme as necessidades técnicas.

### ***ESTRUTURA PROJETIZADA***

A estrutura projetizada vem ganhando espaço rapidamente no mercado, e isso ocorre por quatro motivos: a **velocidade de resposta**, conforme orientada pelo mercado; a **criação** de produtos, processos ou serviços que precisam de informações constantes das diversas áreas de conhecimento; o **crescimento tecnológico** em quase todas as organizações que tendem a desestruturar a gestão tradicional; e a **alta administração das organizações**, que já não se sentem tão confiantes para a coordenação e compreensão de todas as atividades que movem a organização. Nessa estrutura projetizada, a independência e a autoridade são competências atribuídas ao gerente de projeto, e os membros da equipe trabalham juntos (COSTA; PEREIRA, 2019).

### ***ESTRUTURA MATRICIAL***

Já na estrutura matricial, em que temos a mescla das duas outras formas estruturais de organização, vemos a criação de equipes de projetos coordenadas por um gerente de projeto, como ocorre na estrutura projetizada e junto a eles a equipe será subordinada por um gerente funcional, como ocorre nas organizações gerenciadas pela estrutura funcional. Dessa forma, os integrantes dos times de projetos são organizados conforme os colaboradores de cada setor funcional, que se dividem entre as atividades rotineiras e as responsabilidades do projeto (CARVALHO; RABECHINI JUNIOR, 2011)

### ***DIFERENÇAS ENTRE AS ESTRUTURAS ORGANIZACIONAIS***

A fim de demonstrarmos as estruturas organizacionais dos projetos, separamos as diferenças entre as 3 estruturas, inclusive as subdivisões da estrutura matricial, pois o que determina essas divisões é o nível relativo de influência entre os gerentes de projetos e os gerentes funcionais (Quadro 1.1)

Características do projeto						
Tipos de estrutura organizacional	Grupos de trabalho organizados por	Autoridade do gerente de projeto	Papel do gerente de projeto	Disponibilidade de recursos	Quem gerencia o orçamento do projeto?	Pessoal administrativo de gerenciamento de projetos
Orgânico ou simples	Flexível: pessoas trabalhando lado a lado	Pouca ou nenhuma	Em tempo parcial: pode ou não ser um papel designado como coordenador	Pouca ou nenhuma	Proprietário ou operador	Pouco ou nenhum
Funcional (centralizado)	Trabalho realizado (ex.: engenharia, fabricação, etc.)	Pouca ou nenhuma	Em tempo parcial: pode ou não ser um papel designado como coordenador	Pouca ou nenhuma	Gerente funcional	Em tempo parcial
Multidivisional (pode replicar funções para cada divisão com pouca centralização)	Um de: produto; processo de produção; portfólio; programa; região geográfica; tipo de cliente	Pouca ou nenhuma	Em tempo parcial: pode ou não ser um papel designado como coordenador	Pouca ou nenhuma	Gerente funcional	Em tempo parcial
Matriz forte	Por função, com gerente do projeto como função	Moderada a alta	Função designada em tempo integral	Moderada a alta	Gerente do projeto	Full-time
Matriz fraca	Função	Baixa	Em tempo parcial: feito como parte de outro trabalho e não uma função designada, como coordenador	Baixa	Gerente funcional	Em tempo parcial
Matriz equilibrada	Função	Baixa a moderada	Em tempo parcial: incorporado nas funções como uma habilidade e pode não ser um papel designado, como coordenador	Baixa a moderada	Misto	Em tempo parcial
Orientado a projetos (composto, híbrido)	Projeto	Alta a quase total	Função designada em tempo integral	Alta a quase total	Gerente do projeto	Em tempo integral
Virtual	Estrutura de rede com nós nos pontos de contato com outras pessoas	Baixa a moderada	Em tempo integral ou parcial	Baixa a moderada	Misto	Poderia ser em tempo integral ou parcial

Características do projeto						
Híbrido	Mix de outros tipos	Mista	Misto	Mista	Misto	Misto
EGP*	Mix de outros tipos	Alta a quase total	Função designada em tempo integral	Alta a quase total	Gerente do projeto	Em tempo integral

Quadro 1.1 | Quadro da influência das estruturas organizacionais nos projetos

\*EGP refere-se a um portfólio, programa ou escritório/organização de gerenciamento de projetos.

Fonte: PMI (2017, p. 47).

O tamanho da equipe, os objetivos do projeto, as condições específicas do projeto; a divisão de autoridade e de responsabilidade; as condições específicas de cada projeto e a política da empresa são fatores determinantes para a escolha da estrutura mais adequada.

## DEFINIÇÃO DA EQUIPE DE UM PROJETO

Para se definir a equipe de um projeto, é importante analisar as características que variam conforme a complexidade dele. Dentro de uma equipe de gestão de projeto, sem nenhuma dúvida, a peça-chave entre os integrantes é o gerente de projeto, e as características desse profissional são as mais variadas:

- Ele deve ser multitarefa, focado na tarefa.
- Muito bom no trabalho em equipe, uma vez que passará a maior parte do tempo comunicando-se.
- Ter habilidades de negociação muito fortes.
- Estar sempre pronto para resolver problemas assim que estes surgirem.
- Não postergar.
- Ser o primeiro a chegar e o último a sair, pois deverá liderar servindo de modelo para a equipe (GERVÁSIO, 2007).

Um gerente de projeto, ao ser designado para a função de liderança, tem como principal atribuição o total conhecimento do ciclo de vida de um projeto. Trata-se de um profissional multidisciplinar, apto a identificar os problemas e buscar soluções eficientes e eficazes, não importando a área do

conhecimento. Quanto mais cedo identificar as falhas na fase inicial do projeto, o custo será muito inferior do que se o erro for detectado do meio para o final do projeto.

Considerando os apontamentos anteriores, busca-se um profissional generalista, que possua uma visão sistêmica. Esse profissional deve possuir bom domínio em inúmeros métodos de gestão de projeto, e para o desenvolvimento de projetos para a área de TI, é impreterível o conhecimento em boas práticas para documentação e do processo de desenvolvimento de software. Estar em constante aprendizado e atualização dos novos métodos ágeis de desenvolvimento é um grande diferencial (PMI, 2017).

A equipe de projeto precisa ser capaz de ouvir, bem como ser unida e sentir-se motivada em prol do sucesso do projeto. As reuniões devem ser curtas e objetivas, e todos devem ser comunicativos e participativos. A leitura e a busca por informações são cruciais para o bom andamento de cada atividade (HENRICHSEN, 2002).

Nos projetos classificados como grandes, devido à quantidade de atividades a serem desenvolvidas, são criados subprojetos com equipes menores, e é comum a designação de um líder, este profissional não atua como chefe, mas que também não é tido como um membro comum. Ele terá como atribuição boa parte das características do gerente de projeto sem ter o poder de sê-lo. Sendo assim, podemos dizer que ser um líder é um degrau para a aquisição de conhecimento e habilidades para se tornar um gerente de projeto (MEYER, 2006).

Entre os 16 fatores principais associados ao insucesso de um software, 9 estão relacionados com problemas de gestão no projeto. A atividade de ser um gestor de projeto de software é a de maior grau de responsabilidade, e é importante que haja um alto grau de organização, bem como a ambientação de recursos humanos para manter todos os envolvidos motivados. Essas habilidades são necessárias para honrar os cronogramas, principalmente quando envolve a utilização de novas tecnologias.

## COMUNICAÇÃO DENTRO DA ÁREA DE TI

Tendo em vista todos os requisitos dos profissionais da área de TI, entramos em um universo pouco confortável para a maioria atuante na área: o da comunicação. Com o objetivo de se ter uma comunicação clara, objetiva e, principalmente, informativa, é fundamental entender o que é necessário ser comunicado e a quem comunicar.

Levando-se em conta o PMBOK, a resposta de algumas questões pode auxiliar na tomada de decisão sobre o que, para quem e quando escrever. Vamos ver algumas delas:

- Qual é a informação necessária?
- Quem precisa dessa informação?
- Qual o nível de detalhamento necessário?
- Quem é o responsável por essa informação?
- Em que momento a informação deve estar disponível?
- Por quais meios de comunicação ela deverá ser enviada?

A equipe de gestão do projeto deverá designar um método, a periodicidade, os formatos e as TICs (tecnologia de informação e comunicação), que serão utilizadas entre as partes interessadas durante o desenvolvimento do projeto. A partir do Quadro 1.2 a seguir, podemos ter um panorama dos métodos de comunicação.

Quadro 1.2 | Categoria dos métodos e TICs

Métodos de comunicação	Descrição
Comunicação interativa	Troca de informações entre dois ou mais <i>stakeholders</i> . Essa é a maneira que se entende como a eficiente para garantir um entendimento comum a todos os envolvidos.
Comunicação ativa ( <i>push</i> )	Devem ser direcionadas para os interessados específicos, que foram designados a obter as informações ou que necessitam delas. De forma geral, não se tem garantias de que as informações foram recebidas ou mesmo compreendidas, só havendo a garantia da distribuição.
Comunicação passiva ( <i>pull</i> )	Útil para informar muitas pessoas, com grandes volumes de informação. De forma geral, as pessoas acessam o conteúdo remotamente, escolhendo quando querem obter as informações, especificando um tempo para isso acontecer.

Fonte: adaptado de Carvalho (2018).

Criar e usar os canais definidos no planejamento do projeto são ações fundamentais para que as trocas de informações referentes ao projeto aconteçam com transparência, assim, será possível esclarecer qualquer dúvida que venha a surgir.

Em gestão de projeto não pode haver suposições ou a crença de que algo tem que ser de uma determina maneira. É fundamental ter evidências sobre as informações, de preferência, por escrito, para que haja um amparado em dados (CARVALHO, 2018).

Figura 1.1 | Processo do gerenciamento de comunicação



Fonte: adaptada de Candido *et al.* (2012).

## CONFLITOS E PLANEJAMENTO PREVENTIVO

Quando se trabalha com pessoas, é inerente a existência de conflitos. Normalmente, eles acontecem devido à falta de comunicação. Exatamente por isso, é um item muito importante a ser tratado no processo de planejamento do projeto, como vimos anteriormente.

A chave para manter os conflitos de forma administrável é elaborar o planejamento preventivo.

Quais as atitudes mais adequadas para se resolver conflitos em projetos? Identifique a causa do conflito: é pessoal, profissional

ou de poder? O que causou a discórdia? É possível reverter? O importante é agir rápido; não permita que esse problema se prolongue; quanto maior o tempo para resolvê-lo, provavelmente, mais complicado ele ficará.

Ouvir cada um dos envolvidos é fundamental, já que cada pessoa contará a sua versão, o seu ponto de vista, o que não podemos dizer que está errado. Após ouvir todos, é necessário ponderar o que realmente é o problema, quais são as causas e, principalmente, solucioná-lo. O importante é agir com imparcialidade (CARVALHO, 2018).

Manter o foco na solução é um caminho bem certeiro, procurar culpados não levará a nada, o importante é focar a resolução do conflito, assim, todos poderão voltar às suas atividades o quanto antes. Um momento muito delicado é quando o conflito ocorre com você; nesse caso, é importante manter a calma e não discutir na hora de raiva.

Creio que, ainda no sentido de identificar problema e resolvê-los, o primeiro passo é assumir os erros. Se você é líder, o importante é ser o exemplo; caso a questão o envolva, peça desculpas e procure soluções para minimizar as consequências. Assumir falhas não é sinal de fraqueza. Os problemas não surgem do nada, eles vêm mostrando pequenos sinais, principalmente quando envolve a equipe, então, esteja muito atento a sua equipe, no menor sinal de divergência, busque a conversa (COSTA; PEREIRA, 2019).

Considerando tantas possibilidades na gestão de um projeto, a quantidade de variáveis existentes para poder analisar, é notória a importância de uma boa gestão de projeto. E com todo esse contexto já apresentado, vale lembrarmos a Lei de Murphy, criada em 1949, quando um engenheiro e capitão da Força Aérea Americana, Edward Murphy, responsável pela criação de um equipamento de medição dos batimentos cardíacos e de respiração, no projeto de teste dos efeitos da desaceleração rápida em pilotos de aeronaves, foi chamado por um técnico para averiguar os problemas que o equipamento apresentava. Ao

perceber que ele havia sido instalado de forma errada, disse: “Se alguma coisa tem a mais remota chance de dar errado, certamente dará”.

#### **EXEMPLIFICANDO**

Um fator importante que nos faz ter a sensação de que as coisas dão erradas somente conosco é a nossa memória seletiva, que, geralmente, reforça-nos os momentos em que a Lei de Murphy funcionou, como o dia em que saímos de casa com um pé de cada meia ou com a blusa do avesso; quando o pão caiu com a manteiga para baixo e a fila em que estávamos demorou mais para seguir do que a outra.

#### **LEIS DE MURPHY**

As leis de Murphy sempre envolvem um senso de bom humor, mas, aparentemente, elas são uma simples e indiscutível verdade, sendo que já existem comprovações científicas de algumas delas, principalmente das ligadas à probabilidade. Considerando o desenvolvimento de software, isto se aplica muito bem: “A informação mais necessária é sempre a menos disponível” (PERCÍLIA, [s.d., s.p.]). Na prática, notamos o fato ocorrendo após a conclusão do sistema ou de uma parte do sistema no momento dos testes ou mesmo na apresentação ao cliente, quando algum dado ou uma informação é esquecida, o que pode implicar graves consequências para o sucesso do Gerenciamento de Projeto.

Pensando nessa última lei de Murphy adaptada para a área de TI, podemos relatar alguns mitos que estão erroneamente relacionados aos profissionais e ao desenvolvimento. Mesmo depois de tantos “milhões” de anos de estudos para se conseguir impedir falhas no planejamento, no desenvolvimento e na finalização de um projeto, na área de sistemas ainda existem inúmeros mitos nos quais um profissional despreparado é capaz de acreditar, como achar que um engenheiro não é capaz de se comunicar ou de ser um líder, o que é um enorme equívoco, pois essas habilidades de comunicação e de liderança podem ser trabalhadas e, ao longo do tempo, desenvolvidas. Muitos

profissionais escolhem a área de TI por serem introvertidos, imaginando que passarão a maior parte do tempo atrás de um monitor e que não terão a necessidade de se comunicar com as pessoas.

Nesse universo grandioso do desenvolvimento de software, encontramos grandes líderes que são introvertidos até hoje, mas que aprenderam a se superar, como Bill Gates (Microsoft), Steve Wozniak (Apple) e Larry Page (Google). Acreditar que o profissional de TI não é criativo, mas apenas lógico, com certeza, está longe de ser uma verdade, afinal, muitas vezes, para entender o que o cliente está solicitando, tem que acionar uma imaginação criativa.

### **MITOS REFERENTES AO DESENVOLVIMENTO DE SOFTWARE**

Saindo dos mitos referentes aos profissionais da área de TI, vamos falar dos mitos referentes ao desenvolvimento de software. Para que perder tempo preocupando-se com os detalhes do projeto? É possível construí-lo pensando apenas em seu propósito geral? Você já se viu em uma situação em que o usuário lhe pediu apenas um botãozinho para fazer isso e depois aquilo e pronto? Quando se pensa no desenvolvimento de um software, deve-se pensar no maior número possível de funcionalidades que devem ser implantadas. É claro que, no início, será bem complicado abranger todo o produto, o que significa que alguma coisa entrará para correção e a manutenção.

Acreditar que se houver atraso no cronograma basta contratar mais desenvolvedores para que os problemas sejam resolvidos é um equívoco; é bem provável que o efeito seja exatamente o oposto, pois, para incluir novos programadores é necessário um processo de familiarização com o código, com a equipe e com o projeto. Desenvolvimento de Software não é uma linha de produção. O profissional precisa entender o processo da empresa para a qual está trabalhando.

Se o programa funciona, então, o trabalho está concluído? E se o programa não está pronto, não tem como iniciar os testes de

qualidade? O fato é que um programa que funciona dentro do ambiente de desenvolvimento não significa que está realmente pronto, pois é necessário saber inúmeras premissas, como: a infraestrutura da empresa está adequada ao sistema? O banco de dados está adequado para o volume de dados e acessos? A tecnologia usada é ideal para a organização? O código está comentado? Um programa é mais do que um executável.

Outra falha muito grave está em acreditar que se os requisitos sofrerem alterações, as alterações serão realizadas de forma rápida e simples. Cuidado! Afinal de contas, valorizar a equipe é, sim, fundamental, mas toda alteração envolve muitos riscos; ela pode até parecer simples, mas quem é capaz de avaliar é o desenvolvedor. Toda alteração “simples” pode gerar outros problemas que nem imaginamos.

#### **REFLITA**

“Gerenciamento de projetos é a aplicação de conhecimentos, habilidades, ferramentas e técnicas às atividades do projeto a fim de **cumprir seus requisitos**” (PMI, 2017, p. 10).

Dentro de todo esses conhecimentos que você adquiriu nesta seção, podemos observar o quanto fascinante é o processo de gestão de um projeto; e não é à toa que, a cada ano, busca-se formas de obter uma “receita” para o sucesso. A busca pela perfeição é constante, e encontrar uma receita que garanta o sucesso realmente não é fácil. Seguindo essas inúmeras orientações, usando aquelas que se adequam melhor a cada situação, as chances de sucesso aumentam e, com certeza, é o sucesso que você vai alcançar.

#### **FAÇA VALER A PENA**

##### Questão 1

Por fazer parte da equipe de desenvolvimento de sistema para a criação do controle de compra e venda de peças de uma montadora de veículos, um colega lhe procurou para pedir um auxílio quanto a uma situação que vem ocorrendo com frequência em relação a outro membro da equipe. Porém, sem

citar nomes, apenas com o objetivo de compartilhar a decisão de busca, o gerente do projeto informa que seu par não tem fornecido todas as informações necessárias para o desenvolvimento do sistema, acrescentando, de forma enfática, que isso já vem ocorrendo a um certo tempo.

O que você acredita que um bom gerente de projeto deve fazer nessa situação? Assinale a alternativa correta.

- a. Substituir o membro que está causando o problema, independentemente da sua produtividade.**
- b. Passar as informações para o departamento de RH.**
- c. Falar com toda a equipe sobre a questão exposta, menos com o funcionário.**
- d. Chamar o membro da equipe para uma conversa e resolver o problema.**
- e. Convocar uma reunião com toda a equipe e expor a situação.**

Questão 2

A empresa de desenvolvimento de sistema, ao ser contratada para criação de um Software de prestação de serviço, é informada sobre todas as funcionalidades que o contratante acredita serem necessárias para o novo projeto do sistema de sua organização. Diante disso, foi informada a data que entende ser aceitável para continuar com a empresa no mesmo formato em que se encontra hoje; foi solicitado ao analista que informe as etapas de implantação, imaginando que o sistema possa ser desenvolvido em uma sequência e assim a fim de que possa controlar as entregas; e ficaram à disposição os funcionários responsáveis por cada um dos módulos, a fim de oferecerem os conhecimentos necessários para os profissionais que atuarão no desenvolvimento.

Para que esse sistema seja considerado um projeto, aponte quais das afirmações são verdadeiras e quais são falsas para ser parte integrante de um projeto.

- ( ) Possui um início e um fim bem definidos, denominados temporalidade.
- ( ) O conhecimento e a prática das áreas específicas estão envolvidos.
- ( ) A rotina de um setor ou de uma organização faz parte do processo de um projeto.
- ( ) Desenvolvimento de um serviço ou produto que nunca foi criado antes.
- ( ) O produto ou o serviço é criado a partir de uma sequência de atividades que podem ser controladas e monitoradas de alguma forma planejada.

Assinale a alternativa que apresenta a sequência CORRETA.

- a. F – V – V – V – F.
- b. V – V – F – F – V.
- c. V – F – F – V – V.
- d. F – F – V – V – V.
- e. V – F – V – V – F.

Questão 3

Ao se trabalhar com projetos de grande complexidade, ou seja, projetos que envolvem o uso de novas tecnologias ou mesmo que estejam relacionados a grandes transformações estruturais, é necessária a montagem de equipes multidisciplinares para se alcançar os resultados almejados, por envolver inúmeros conhecimentos nas mais diversas áreas. Devido a esta situação, os projetos de grande complexidade, devem ser desmembrados em subprojetos menores.

Sendo assim, analise as afirmativas a seguir:

- I. Assim, os profissionais para acompanhar estes projetos precisam ser profissionais generalistas.
- II. O gerente de projeto deve ser o maestro da equipe, guiando-a e sendo um bom negociador.

III. Os profissionais envolvidos devem ser especialistas e estar atentos aos detalhes. o suficiente para atuar em um projeto de grande complexidade

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

- a. I, II e III.**
- b. I, apenas.**
- c. II, apenas.**
- d. I e II, apenas.**
- e. I e III, apenas.**

#### ***REFERÊNCIAS***

ABNT — Associação Brasileira de Normas Técnicas. **NBR ISO 10006**. Sistema de gestão da qualidade. Diretrizes para a gestão da qualidade em empreendimentos. Rio de Janeiro: ABNT, 2003.

CANDIDO, R. *et al.* **Gerenciamento de projetos**. 1. ed. Curitiba: Aymará, 2012.

CARVALHO, F. C. A. de. Gestão de projetos. 2. ed. São Paulo: Pearson Education do Brasil, 2018.

CARVALHO JUNIOR, M. R. de. **Gestão de projetos: da academia à sociedade**. 1. ed. Curitiba: Intersaber, 2012.

CARVALHO, M. M. de; RABECHINI JUNIOR, R. **Fundamentos em gestão de projetos: construindo competências para gerenciar projetos**. Curitiba: Atlas, 2011.

COSTA, A. B. da; PEREIRA, F. da S. **Fundamentos de gestão de projetos da teoria à prática: como gerenciar projetos de sucesso**. 1. ed. Curitiba: Intersaber, 2019.

RISCO. In: **DICIO, Dicionário Online de Português**. Porto: 7Graus, 2020. Disponível em: <https://www.dicio.com.br/projeto/>. Acesso em: 27/07/2020.

**GERVÁSIO, S. Apostila gestão dos recursos humanos.** [S.l.]: IETEC, 2007.

**HELDMAN, K. Gerência de projetos:** guia para o exame oficial do PMI. Rio de Janeiro: Campus, 2003.

**HENRICHSEN, W. A. Discípulos são feitos, não nascem prontos.** 1. ed. Belo Horizonte: Atos, 2002.

**KERZNER, H. Gestão de projetos:** as melhores práticas. Porto Alegre: Bookman, 2002.. Acesso em: 29 out. 2020.

**MEYER, J.** A formação de um líder. 2. ed. Belo Horizonte: Sografe, 2006.

**PERCÍLIA, E. Lei de Murphy.** [s.d.]. Disponível em: <https://bit.ly/3jW6ajk>. Acesso em: 23 dez. 2020.

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

**SAATY, T. L.** The analytic hierarchy and analytic network processes for the measurement of intangible criteria and for decision-making. In: FIGUEIRA, J.; GRECO, S.; EHRGOTT, M. **Multiple criteria decision analysis: state of the art surveys.** New York: Springer, 2005.

**TUMAN, G. J.** Development and implementation of effective project management information and control systems. In: **CLELAND, D. I.; KING, W. R. Project management handbook.** New York: Van Nostrand Reinhold, 1983.

**ZIMMERMANN, R. D. C. Mudanças de cenários competitivos e seus impactos no setor de saneamento:** estudo de multicasos em companhias de saneamento do estado de Santa Catarina. 2001. Tese (Engenharia de Produção) — Universidade Federal de Santa Catariana, Santa Catarina, 2001.

## FOCO NO MERCADO DE TRABALHO



Fonte: Shutterstock.

**Deseja ouvir este material?**

### ***SEM MEDO DE ERRAR***

No desenvolvimento de um projeto que contém inúmeras informações pessoais e profissionais dos colegas de trabalho e inclusive suas, a equipe deve ser composta de profissionais com conhecimentos dos mais variados, capazes de analisar os dados a serem tratados e posteriormente analisados, propor resolução dos problemas a serem resolvidos nesse módulo e prever possíveis problemas futuros. Para isso, são importantes a criatividade e a inovação na forma de gerar os relatórios de desempenho, além do comprometimento e das características de caráter de discrição. O gerente de projeto deve acompanhar o desenvolvimento muito de perto, observando o cumprimento dos prazos, deixando sempre a equipe motivada. Para todos os profissionais envolvidos, é fundamental que exista um bom relacionamento interpessoal, já que é um sistema que trata da análise de indivíduos, que envolve não apenas questões profissionais, mas também questões de cunho pessoal.

Para que o projeto consiga ser bem desenvolvido, é sempre interessante pensar em um plano de gestão da comunicação. Pensando na boa comunicação entre os stakeholders, uma das opções é utilizar a forma de comunicação interativa, a fim de se garantir que a informação alcance todos os interessados e, principalmente, seja compreendida de

forma ampla e precisa. Para esse projeto, o gerente será responsável por comunicar os stakeholders devido às questões de informações sigilosas.

Por tanto, toda e qualquer comunicação se dará por meio de reuniões presenciais ou a distância, por meio de videoconferência, telefone ou internet para todos da equipe, mantendo-se a ética no trabalho, o planejamento e a organização propostos no início desse projeto, bem como flexibilidade e empatia entre todos os envolvidos no processo de desenvolvimento.

Pensando em tratar os conflitos prováveis para esse projeto, um dos riscos que a grande maioria dos projetos possui de forma eminentemente pode ser o mais comum ou o mais específico, como é de costume ocorrer em projetos de complexidade. Devemos nos preocupar com os conflitos de referência técnica, relacionados à forma de programação, além dos conflitos voltados para as prioridades das etapas, que serão os mais frequentes no âmbito técnico, mas nada impede que ocorram os demais, dessa forma, é de extrema relevância que o gerente do projeto observe o comportamento dos integrantes da equipe, buscando identificar o início de algum conflito, a fim de resolvê-lo o quanto antes, para que não afete o bom andamento do cronograma.

Além disso, é válido pensar em outras possibilidades, e sempre fazer um planejamento de análise de risco e de comunicação, bem como lembrar a importância da negociação e da comunicação por parte do gerente de projeto, a fim de saber como cada membro da equipe está visualizando o seu próprio desempenho, bem como buscar, juntamente ao programador, meios de resolver possíveis insatisfações pessoais. Como atender às necessidades individuais de cada integrante da equipe e, ao mesmo tempo, atender às metas definidas no início do projeto?

Para esse projeto e a tantos outros, é importante fazer com que todos participem de forma ativa das reuniões, expressando seus sucessos e suas frustrações, de forma que os seus pares possam auxiliar uns aos outros, em busca de soluções criativas, inovadoras e fazendo com que a equipe permaneça unida em prol do projeto, do prazo e das diretrizes propostas.

Como a empresa atua no modelo de gestão orientada pelo mercado e, neste caso, o projeto é para a gestão da própria empresa, o gerente de projeto também deve estar atendo aos *stakeholders* concorrentes, seguir as tendências de mercado, quando possível, e estar à frente do mercado, buscando a implementação de novas tecnologias, oferecendo aos seus desenvolvedores cursos de atualizações, bem como recursos para que se mantenham motivados, melhorando o desempenho individual e, consequentemente, da equipe.

Por fim, como pôde observar aqui, neste tópico, as possibilidades de soluções para o projeto serão diversas, porém temos a certeza de que a forma como você desenvolveu a sua solução estará de acordo com todas as colocações e os apontamentos feitos até aqui. Parabéns e sucesso.

# **FOCO NO MERCADO DE TRABALHO**

## **INTRODUÇÃO À ANÁLISE E AO PROJETO DE SISTEMAS**

*Daniela Teresa Rossignoli Uebele*



Fonte: Shutterstock.

**Deseja ouvir este material?**

### ***SEM MEDO DE ERRAR***

No desenvolvimento de um projeto que contém inúmeras informações pessoais e profissionais dos colegas de trabalho e inclusive suas, a equipe deve ser composta de profissionais com conhecimentos dos mais variados, capazes de analisar os dados a serem tratados e posteriormente analisados, propor resolução dos problemas a serem resolvidos nesse módulo e prever possíveis problemas futuros. Para isso, são importantes a criatividade e a inovação na forma de gerar os relatórios de desempenho, além do comprometimento e das características de caráter de discrição. O gerente de projeto deve acompanhar o desenvolvimento muito de perto, observando o cumprimento dos prazos, deixando sempre a equipe motivada. Para todos os profissionais envolvidos, é fundamental que exista um bom relacionamento interpessoal, já que é um sistema que trata da análise de indivíduos, que envolve não apenas questões profissionais, mas também questões de cunho pessoal.

Para que o projeto consiga ser bem desenvolvido, é sempre interessante pensar em um plano de gestão da comunicação. Pensando na boa comunicação entre os stakeholders, uma das opções é utilizar a forma de comunicação interativa, a fim de se garantir que a informação alcance todos os interessados e, principalmente, seja compreendida de

forma ampla e precisa. Para esse projeto, o gerente será responsável por comunicar os stakeholders devido às questões de informações sigilosas.

Por tanto, toda e qualquer comunicação se dará por meio de reuniões presenciais ou a distância, por meio de videoconferência, telefone ou internet para todos da equipe, mantendo-se a ética no trabalho, o planejamento e a organização propostos no início desse projeto, bem como flexibilidade e empatia entre todos os envolvidos no processo de desenvolvimento.

Pensando em tratar os conflitos prováveis para esse projeto, um dos riscos que a grande maioria dos projetos possui de forma eminentemente pode ser o mais comum ou o mais específico, como é de costume ocorrer em projetos de complexidade. Devemos nos preocupar com os conflitos de referência técnica, relacionados à forma de programação, além dos conflitos voltados para as prioridades das etapas, que serão os mais frequentes no âmbito técnico, mas nada impede que ocorram os demais, dessa forma, é de extrema relevância que o gerente do projeto observe o comportamento dos integrantes da equipe, buscando identificar o início de algum conflito, a fim de resolvê-lo o quanto antes, para que não afete o bom andamento do cronograma.

Além disso, é válido pensar em outras possibilidades, e sempre fazer um planejamento de análise de risco e de comunicação, bem como lembrar a importância da negociação e da comunicação por parte do gerente de projeto, a fim de saber como cada membro da equipe está visualizando o seu próprio desempenho, bem como buscar, juntamente ao programador, meios de resolver possíveis insatisfações pessoais. Como atender às necessidades individuais de cada integrante da equipe e, ao mesmo tempo, atender às metas definidas no início do projeto?

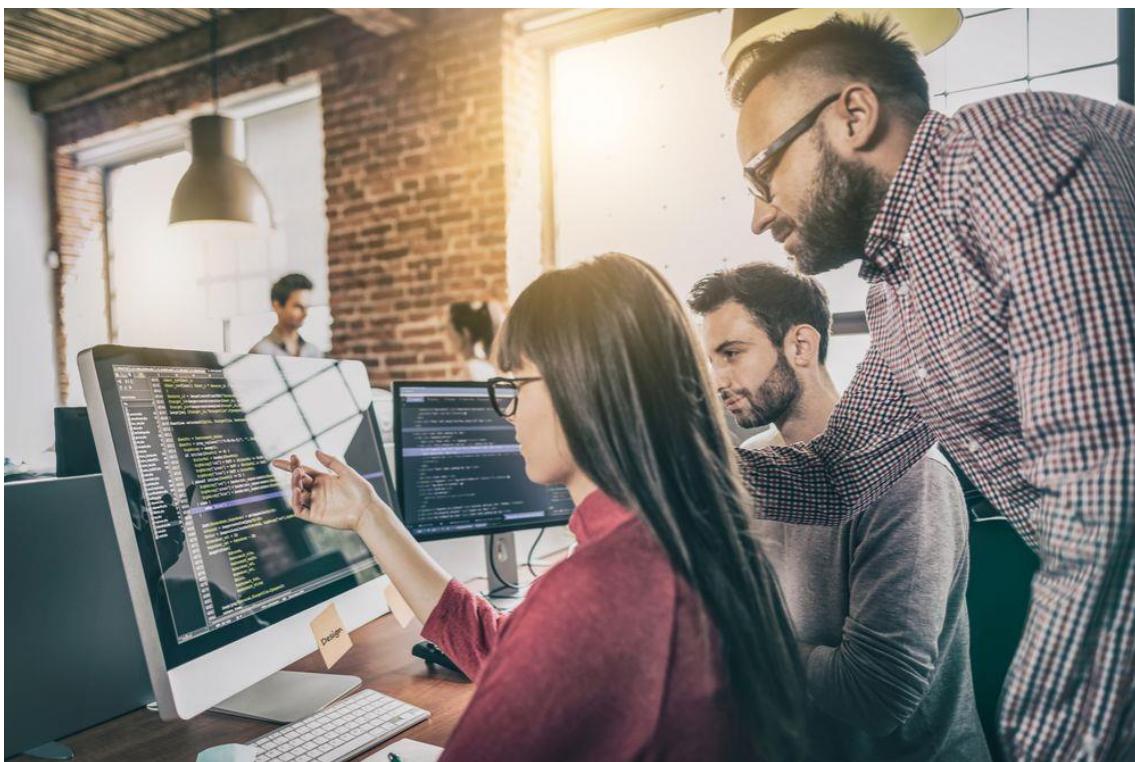
Para esse projeto e a tantos outros, é importante fazer com que todos participem de forma ativa das reuniões, expressando seus sucessos e suas frustrações, de forma que os seus pares possam auxiliar uns aos outros, em busca de soluções criativas, inovadoras e fazendo com que a equipe permaneça unida em prol do projeto, do prazo e das diretrizes propostas.

Como a empresa atua no modelo de gestão orientada pelo mercado e, neste caso, o projeto é para a gestão da própria empresa, o gerente de projeto também deve estar atendo aos *stakeholders* concorrentes, seguir as tendências de mercado, quando possível, e estar à frente do mercado, buscando a implementação de novas tecnologias, oferecendo aos seus desenvolvedores cursos de atualizações, bem como recursos para que se mantenham motivados, melhorando o desempenho individual e, consequentemente, da equipe.

Por fim, como pôde observar aqui, neste tópico, as possibilidades de soluções para o projeto serão diversas, porém temos a certeza de que a forma como você desenvolveu a sua solução estará de acordo com todas as colocações e os apontamentos feitos até aqui. Parabéns e sucesso.

## **NÃO PODE FALTAR**

### *GERENCIAMENTO DE PROJETOS EM CASCATA*



Fonte: Shutterstock.

**Deseja ouvir este material?**

#### ***PRATICAR PARA APRENDER***

Para você que está ingressando no universo de TI, não se assuste com tantas informações e com tantas siglas, apenas tenha consciência que ter ouvido falar ou ter lido algo sobre elas já é um grande passo para acompanhar e compreender as pessoas com alguns anos na área. Além disso, entender os fundamentos dos processos que envolvem a gestão de projetos, inclusive os projetos de TI, não será nenhum problema. Apesar de serem muito óbvios, é impressionante como foi necessário documentar e criar metodologias, também chamadas de tecnologias ou, ainda, de boas práticas, para que as organizações “obrigassem” seus funcionários a utilizá-los.

Com a união lógica de processos, criou-se o Ciclo de Vida do projeto. Por meio de fases bem delimitadas e definidas da sua importância e de seus objetivos, passaram a ficar mais certeiros os resultados de um Gerenciamento de Projeto, fazendo com que o tempo gasto na criação de um produto ou serviço realmente valha muito, agregando valores e tornando as empresas referências nos setores de negócio em que se propõem atuar.

A utilização de metodologias, *frameworks* e conceitos vieram para ofertar recursos tecnológicos, como softwares, para auxiliarem no processo de governança. Com maior valorização após 2002, com a Lei SOX aprovada no congresso norte americano, fazendo com que os fundamentos de ITIL, COBIT e PMI se tornassem mais relevantes dentro das organizações e os profissionais qualificados nestas metodologias pudessem comprovar sua eficiência e eficácia com as certificações e avaliações das equipes nas implantações.

Foi a partir da criação de métricas para se conseguir medir o grau de desenvolvimento de governança e de gestão dos projetos que se pôde definir o grau de maturidade para cada uma das empresas movidas pela gestão orientada pelo mercado, ainda assim, podendo ser aplicado nas empresas com gestão tradicional ou matricial.

Juntas, ITIL, COBIT e PMI formam uma base solidificada para aumentar o grau de governança e de gestão dos projetos e dos serviços em TI, independentemente de serem dos parceiros de negócio, das terceirizadas ou mesmo de uso interno — lembrando que há diretrizes em forma de guia, processos e bibliotecas que deverão ser adaptadas e utilizadas de acordo com a necessidade de cada empresa e/ou projeto, o que significa não ser necessário o total domínio de todo o conteúdo das três tecnologias, mas, sim, buscar nelas apoio de ferramentas e boas práticas para serem implantadas no momento adequado.

Retomando o projeto em que foi alocado pela empresa de desenvolvimento de software que visa à criação de um módulo do sistema de gestão dos desenvolvedores de inúmeros projetos geridos na própria empresa ou com seus desenvolvedores em loco no setor de TI da empresa contratante, o novo módulo terá como funcionalidade principal os registros das combinações de pares entre os desenvolvedores quando adotado o *Extreme Program*, que será adicionado aos demais módulos em produção.

Dentro da organização em que está trabalhando, a implantação da governança e da gestão de serviço dá-se por meio de COBIT e ITIL, respectivamente, e a gestão dos projetos ocorre com base no Guia PMBOK de boas práticas. Para o desenvolvimento desse módulo de monitoramento e gestão dos desenvolvedores de sua organização, a fim de mensurar a qualidade do serviço que está sendo prestado por cada um, dentro ou fora da organização, busca-se verificar o grau de satisfação do cliente e a eficiência dos profissionais. Sendo você um desenvolvedor criativo e inovador, pode sugerir algum processo de avaliação referente à organização e ao planejamento da equipe, bem como a práticas que avaliem o relacionamento interpessoal, a empatia e a ética de cada um dos profissionais, do ponto de vista e das pessoas dos setores em que cada um está atuando no momento do projeto, embasado no ITIL 4.

Pense em propor sugestões para criação de relatórios automatizados, sem a necessidade de expor quem está analisando, a fim de se evitar conflitos interpessoais. É importante que esse relatório traga informações com valores significativos.

Para mostrar o seu domínio na análise e resolução de problemas, vale formalizar uma sugestão de como aplicar as dez áreas do conhecimento da qual PMBOK trata, fazendo com que fique claro o processo de integração entre as áreas, dando enfoque aos grupos de processos de execução e de planejamento.

Lembre-se de que uma das atividades a serem tratadas é a garantia de um bom suporte de infraestrutura das organizações, em que o sistema desenvolvido atende não somente às necessidades externas da organização, mas também às necessidades internas, facilitando as estratégias e os objetivos da companhia no que diz respeito à gestão dos objetivos. Verifique um modelo baseado em COBIT para ser seguido periodicamente e defina essa periodicidade.

Os relatórios que serão gerados por meio do módulo que você está desenvolvendo devem estar alinhados aos objetivos de governança da organização, buscando-se métricas para definir os níveis de maturidade. Além disso, as avaliações com reconhecimento do mercado internacional fazem parte desse contexto de avaliação.

E quanto a esses relatórios, quais informações devem ser contempladas, a fim de auxiliarem na governança da empresa em que trabalha.

Considerando todo o conhecimento que adquirirá nesta seção e com o domínio das boas práticas, tornar-se-á um profissional bem informado, que se utilizará de modelos sólidos de normas para planejar, executar e monitorar os projetos em que trabalhará, independentemente de ser um membro da equipe de TI ou um gerente de projeto da empresa.

### **CONCEITO-CHAVE**

#### **PMBOK**

Para o desenvolvimento de softwares, faz-se importante utilizar técnicas de gestão de projeto, ferramentas e *frameworks* adequados para cada fase, conseguindo, assim, tratar e prevenir riscos eminentes ao desenvolvimento de softwares. Assim sendo, encontre fundamentos em guias de boas práticas tanto para a gestão do projeto como para a governança de TI (Tecnologia de Informação) nas organizações.

Para que você possa iniciar os estudos e compreender como utilizar esses recursos largamente empregados, falaremos do Guia de gestão de projetos PMBOK, criado e mantido pela PMI; veremos sobre governança da gestão interna em TI, com o COBIT; e para a gestão da qualidade dos serviços, vamos falar sobre ITIL (NEWTON,2011).

Ao conhecer esses três guias de boas práticas, você será capaz de selecionar os melhores *frameworks* e as ferramentas adequadas para cada tipo de projeto, pois é preciso levar em conta todos os aspectos dos processos e da gestão, considerando que todo projeto é único, temporário e possui sequência de atividades específicas.

Portanto, você, aluno, que será um gestor ou integrante de uma equipe de desenvolvimento de software, deverá analisar o projeto no qual estará envolvido, buscando escolher, dentro dos guias de boas práticas, o que melhor se adequa a sua realidade, pensando nas premissas e nas restrições do seu projeto de desenvolvimento de software, na equipe envolvida e no tipo de estrutura organizacional em que atuará (FERREIRA, 2020).

Neste momento, deve haver algumas perguntas sobre alguns termos já utilizados até aqui e que, talvez, não sejam familiares, mas vamos com calma! Vamos definir o que são ferramentas: são programas utilizados para facilitar a realização de algumas tarefas, tais como: gerar documentos, desenvolver a análise de software, gerar um cronograma, compartilhar as informações com a equipe, entre outras atividades do projeto.

Agora, o que são *frameworks*? São templates ou modelos com inúmeras funções, podendo ser utilizados pelo desenvolvedor e/ou gerente. O objetivo é economizar tempo para reproduzir as funções e/ou processos criados em diferentes projetos, facilitando o seu gerenciamento e a sua execução dos projetos (VERAS, 2019).

E o que são premissas? Trata-se de definir claramente o que se acredita ter como base para o funcionamento do sistema, o que não garante que isso é ou será uma realidade. Sendo assim, faz-se necessária a averiguação desses parâmetros.

Temos também as restrições, que são as condições restritivas, os limites impostos para o desenvolvimento do sistema, que poderá apresentar algum tipo de problema, como atrasos, aumento de custo, perda de qualidade, entre outros problemas a serem administrados durante o desenvolvimento do sistema.

Ao final do estudo, você estará capacitado a atender às necessidades do mercado, tendo como referência as boas práticas que foram documentadas e organizadas no *Project Management Body of Knowledge* (PMBOK) pelo *Project Management Institute* (PMI), fundado em 1969, na Pensilvânia, nos Estados Unidos. Tal guia oferece as melhores práticas para a área de TI, promovendo resultados de valores aos projetos desenvolvidos a partir de sugestões e práticas a serem adotadas de acordo com as necessidades de cada projeto.

Diante disso, veremos algumas das ferramentas sugeridas, como Termo de Abertura de Projeto (TAP), Estrutura Analítica de Projeto (EAP) e o seu dicionário (PMBOK, 2017).

#### **ASSIMILE**

Sempre que passamos por alguma experiência difícil, levamos os bons e os maus resultados como aprendizado, a fim de não cometermos os mesmos erros. Quando falamos de projetos, inclusive em desenvolvimento de software, isso também ocorre; os erros das decisões tomadas durante o período completo de um projeto, ou seja, do início da ideia até a sua conclusão ou mesmo desistência, nortearão um novo trabalho.

Além do PMBOK, existem inúmeras ferramentas ou frameworks para auxiliar os processos de desenvolvimento de projetos, como o *International Project Management Institute* (IPMA) e o *Projects in Controlled Environments* (PRINCE2), além daqueles mais específicos para a área de desenvolvimento de Software, como as metodologias ágeis, que auxiliam em métodos de programação, como *Extreme Programming* (XP), *Scrum* e *Dymamic Systems Development Method* (DSDM), que são métodos de programação que aceleram o desenvolvimento de software e buscam qualidade, fica a indicação para um aprofundamento (FERREIRA, 2020).

No entanto, quando falamos da análise de um projeto de forma ampla, que envolve uma estrutura empresarial mais tradicional/sequencial, e pelo fato de o PMBOK ter sido o

precursor da organização das práticas de sucesso aplicadas em inúmeros projetos no decorrer de anos, este se tornou a base da gestão de projetos (SOMMERVILLE, 2018).

Frente a isso, um gerente de projeto bem capacitado e com experiência em projetos, juntamente a sua equipe e demais envolvidos, conhecidos como stakeholders (toda e qualquer pessoa envolvida direta ou indiretamente no projeto), estabelecerão, logo no primeiro momento, o escopo do projeto e o escopo do produto, ou seja, os objetivos a serem alcançados, gerando um produto ou serviço quando da conclusão do projeto.

Dessa forma, falamos sobre o grupo de processo de Iniciação. A partir do escopo, serão definidos os recursos necessários o que inclui não somente os recursos materiais, e financeiros, como também os recursos humanos necessários, bem como o plano de gestão do tempo das pessoas, o controle de qualidade e inúmeras outras vertentes (COSTA; PEREIRA, 2019).

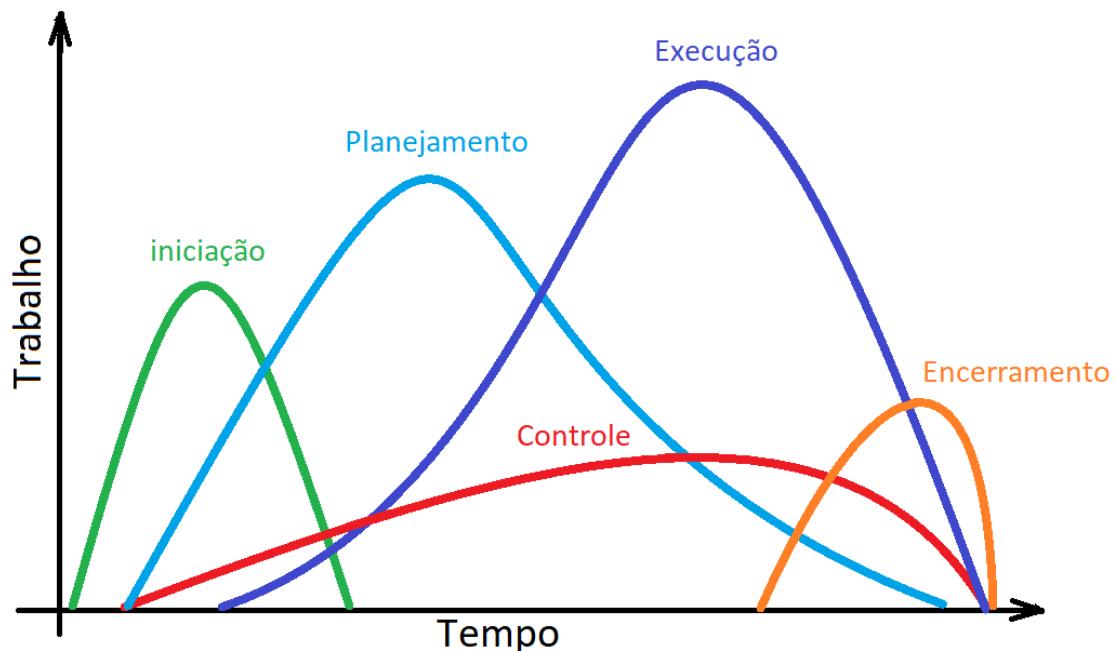
Ao ingressar em uma equipe de desenvolvimento que envolve a criação de uma nova aplicação de software ou mesmo um módulo adicional a ser implementado, a divisão em etapas ou processos torna-se necessária, afinal, definir uma sequência de atividades contínuas, a fim de que possam ser mensuradas, com a implantação de ferramentas que controlam os prazos, a qualidade e o custo, por meio de métricas que serão definidas no grupo de processo de planejamento, é o que tornará possível determinar o nível de maturidade da equipe e/ou empresa, a partir do seu grau de eficiência e eficácia (CARVALHO, 2018).

## ENTENDENDO OS PROCESSOS DO CICLO DE VIDA DE UM PROJETO

Visando a um bom gerenciamento de projetos com um controle adequado, as empresas se utilizam da divisão do projeto em cinco categorias, conhecidas como grupo de processos ou fases de um projeto, constituindo o seu ciclo de vida do projeto. Esses grupos de processo são: iniciação, planejamento, execução, monitoramento e controle e encerramento (Figura 1.2).

Com esses processos, percebeu-se que essas divisões se aplicam a todo e qualquer projeto a ser desenvolvido, inclusive aos de desenvolvimento de softwares. A aplicação das boas práticas do PMBOK somada à globalização e ao avanço da tecnologia tornaram-se um conjunto de práticas adequadas para se organizar o desenvolvimento de um sistema. A partir dessa divisão, pôde-se obter melhores resultados em projetos, devido à organização das atividades, permitindo um melhor controle do tempo, verificando-se a qualidade e, em casos de alterações, a agilidade na tomada de decisões (CARVALHO, 2018).

Figura 1.2 | Gráfico de representação genérica das cinco fases da gestão de um projeto



Fonte: adaptada de CARVALHO (2018).

### CICLO DE VIDA E SEUS PROCESSOS

Você consegue se imaginar como um gerente de projetos eficiente? Vamos ver o que é proposto pelo PMBOK quanto à importância de se conhecer e adotar o ciclo de vida e seus processos.

O primeiro processo, como o próprio nome sugere, é o grupo de processo de Iniciação, em que se define o escopo do produto e o escopo do projeto, obtendo autorização para iniciar. A importância da boa prática nessa fase é a definição clara dos objetivos a serem alcançados no desenvolvimento do software ou de um módulo adicional ao sistema em produção. Normalmente, os objetivos de um projeto são representados pela otimização de um procedimento ou pelo incremento de uma funcionalidade ao sistema, porém, em qualquer das hipóteses, deve agregar valor aos negócios da organização.

O segundo é o grupo de processo de **Planejamento**, em que se fazem inúmeras perguntas a fim de se obter as informações necessárias para um bom planejamento, definindo de forma clara o escopo do projeto, alinhando as ações com o objetivo dentro do tempo, custo e qualidade planejados. Exemplo de questionamentos que norteiam a construção de um plano: o que será feito? Quanto tempo temos para realizar? Qual a sequência das atividades? Quem será o responsável? Quais são os riscos? Qual o nível de qualidade que precisaremos atender? Qual será o custo?

É nesse processo que se documenta como serão efetuados os controles e o monitoramento do desenvolvimento do projeto, como será a administração das mudanças, como será o acompanhamento, a análise e o controle do progresso e do desempenho do projeto. Pensando na criação de um novo sistema, é muito claro observar essa etapa acontecendo (CANDIDO, 2012).

## EXEMPLIFICANDO

Ao desenvolvemos um sistema de agendamento para um especialista da área da saúde, devemos pensar em como esse sistema funcionará. Portanto, é importante assimilarmos não apenas as orientações do profissional especialista, mas também da secretária e dos pacientes, afinal, todos possuem interesse no bom funcionamento desse serviço.

Supondo que o sistema será web, permitindo aos pacientes o acesso à agenda do médico, os horários disponíveis deverão ser acessados, independentemente de se ter um paciente agendado ou ser o dia indisponível, pois o profissional não está atendendo somente no consultório.

Para esse sistema, organizar a sequência em que serão desenvolvidas as funcionalidades é uma maneira de dividir as entregas e permitir que os *stakeholders* deem o seu feedback antes mesmo da finalização do sistema, reduzindo, assim, o tempo de correções ou de falhas de entendimentos. Nessa descrição, falamos exclusivamente dos processos de iniciação e planejamento; avaliamos as funcionalidades básicas e começamos a pensar no planejamento das atividades, na forma de entrega e no planejamento da correção durante a execução, sempre buscando avaliar os riscos durante o processo de desenvolvimento do software.

O terceiro grupo é o de processo de **Execução**, momento em que se investe a maior parte do orçamento previsto e do tempo para execução e conclusão. É nesse processo que são desenvolvidas as atividades planejadas, realizando-se as pequenas conclusões dos produtos definidos no processo de planejamento. Além disso, o gerente de projeto deve atuar com maior ênfase juntamente à equipe, coletando e distribuindo informações, buscando garantir a qualidade planejada, interagir com fornecedores e usuários, mobilizar a equipe a buscar o melhor desempenho, e observar os riscos previstos e os possíveis imprevistos (PMBOK, 2017).

O quarto grupo é o de processos de **Monitoramento e Controle** definidos no planejamento, em que se aplicam os padrões métricos de desempenho que foram planejados. Durante o monitoramento e o controle, são detectadas as ameaças, devendo ser solucionadas imediatamente, bem como identificada a sua causa, para que seja anotada no item “lições aprendidas”. Além disso, é durante o monitoramento e controle que o gerente do projeto desempenha suas habilidades de negociação, quando necessário, para garantir que o projeto mantenha o foco em seus objetivos., inclusive, caso justifique deve proceder os ajustes no plano do projeto.

Por último, o quinto grupo é o de processo de **Encerramento**, em que se documentam as lições aprendidas para usá-las nos próximos projetos. Nessa etapa, também são feitos os encerramentos da documentação técnica. Apesar de tratar os processos como itens independentes, é fundamental entender que eles ocorrem de forma simultânea em alguns momentos e interagem entre si (COSTA; PEREIRA, 2019).

O terceiro grupo é o de processo de **Execução**, momento em que se investe a maior parte do orçamento previsto e do tempo para execução e conclusão. É nesse processo que são desenvolvidas as atividades planejadas, realizando-se as pequenas conclusões dos produtos definidos no processo de planejamento. Além disso, o gerente de projeto deve atuar com maior ênfase juntamente à equipe, coletando e distribuindo informações, buscando garantir a qualidade planejada, interagir com fornecedores e usuários,

mobilizar a equipe a buscar o melhor desempenho, e observar os riscos previstos e os possíveis imprevistos (PMBOK, 2017).

Como foi discutido, todo projeto terá seu ciclo de vida composto de algumas sequências de atividades. A gestão do projeto busca o maior sucesso para alcançar o escopo do projeto; e para que seja concluído com sucesso, é necessário o equilíbrio de quatro pilares: escopo, custo, qualidade e tempo.

#### REFLITA

O que faz com que uma empresa ou pessoa dê início a um projeto?

Se não há um propósito definido, é óbvio que não há motivo para se iniciar um projeto, não é?

Enquanto o projeto estiver em desenvolvimento, que mudanças poderão ocorrer?  
Quanto esse projeto pode ser modificado?

O que é possível controlar nos imprevistos?

Ao iniciar um projeto, é fundamental responder a essas e muitas outras perguntas, a fim de que se inicie um projeto com segurança e otimismo de conclusão com sucesso.

A divisão em cinco processos ocorre para que seja realmente possível analisar os resultados obtidos em cada um dos processos. Um bom exemplo, na área de desenvolvimento de sistema, quanto ao processo de execução, é a entrega de uma funcionalidade para análise dos dados e de informações que estão sendo geradas. Estando tudo dentro do esperado e com a aprovação dos *stakeholders*, é possível avançar para a próxima etapa, mas se for detectada alguma irregularidade, ou seja, a necessidade de mudanças, deverá voltar para o processo de planejamento, para os devidos ajustes estejam dentro de um custo aceitável e não afetando o tempo da entrega, será aprovado e o projeto terá autorização para continuar. Nesse exemplo, pudemos observar a existência dos processos de execução, monitoramento e controle, o retorno ao planejamento (uma vez que foi necessário) e o processo de encerramento de uma etapa.

## ÁREAS DE CONHECIMENTO PARA ORGANIZAR E OTIMIZAR A GESTÃO

O PMBOK define dez áreas de conhecimento para se organizar e otimizar a gestão, sendo essas atividades interrelacionadas e distribuídas nos cinco grupos de processos de um projeto. Muitas delas, aliás, encontram-se em todas as cinco fases.

Você irá conhecer as áreas do conhecimento para que possa aprimorar suas habilidades em gestão de projetos observando a figura 1.3.

Figura 1.3 | As 10 áreas do conhecimento segundo PMBOK



Fonte: adaptada de PMI (2017).

Considerando que, em breve, você estará envolvido em um processo de desenvolvimento de software ou, até mesmo, será um gerente de projeto, tendo como responsabilidade a liderança de uma equipe e o total domínio do ciclo de vida do projeto, em que, dentro de cada fase do desenvolvimento, existirá um grupo de atividades a serem cumpridas conforme o tempo, a qualidade e o custo, com o intuito de alcançar o escopo predefinido.

A partir da compreensão do ciclo de vida e de seus respectivos processos, o gerente do projeto consegue identificar possíveis desvios, estando apto a resolvê-los da forma mais adequada e, preferencialmente, no momento em que o problema surgir, para que os custos de reparação a falha sejam os menores possíveis. Dentro deste cenário de riscos que estão intrínsecos em um projeto, dispõe-se as áreas de conhecimento recursos, riscos e aquisição no topo da figura, mostrando o quanto é fundamental que as áreas do conhecimento comunicação, partes interessadas sejam integradas com todas as outras quatro áreas da base da Figura 1.3 (PMI, 2017).

Veja que a governança oferece *frameworks* de apoio para, juntamente à gestão de projetos de software, encontrarem as melhores soluções, ou seja, ela se associa ao gerenciamento de serviços de TI, permitindo um melhor controle dos processos, alinhamento das estratégias e gerenciamento dos recursos, priorizando, direcionando as atividades e buscando meios de atingir os objetivos da empresa, agregando valor e equilibrando os riscos com o retorno de investimento em TI (VARGA, 2020).

Pra a obtenção desses benefícios, existem os frameworks ITIL, uma biblioteca de infraestrutura de TI (*Information Technology Infrastructure Library*) que busca gerenciar os serviços de TI, definindo como as organizações podem implementar e utilizar, de forma adequada, os serviços do setor de, alinhando-os aos negócios da empresa (NEVES; DONÁ, 2020). Já por meio do *Control objectives for information and related technology* (COBIT) ou Objetivos de controle para tecnologias da informação e relacionadas, é possível escolher ferramentas de análise e modelos que possam auxiliar os gestores de TI a maximizarem a confiança da informação, dando

mais valor a esse ativo. O COBIT define boas práticas focadas no controle em vez da execução, sempre buscando otimizar os investimentos e reduzir os riscos do setor de TI (VARGA, 2020).

Apesar de ambos falarem de governança, observe que os focos de atuação são distintos. Enquanto ITIL trabalha a qualidade do serviço, ou seja, foca o atendimento ao cliente com rapidez, o preço justo e a qualidade, junto de ITIL pode-se implementar o COBIT, que gerencia os processos, fornece meios de se medir a performance e foca o que a empresa precisa, proporcionando confiança às informações e aos sistemas de informação (NEVES; DONÁ, 2020).

Você pode estudá-las individualmente, no entanto, é fundamental a compreensão de que as boas práticas, as ferramentas sugeridas e os *frameworks* sejam utilizados de forma conjunta, adequada a cada projeto. Em geral, não são utilizadas todas as ferramentas, por isso, utilizam-se métricas capazes de avaliar o grau de maturidade da empresa, que utiliza, constantemente, o desenvolvimento de projetos. É por meio da qualificação dos profissionais e, principalmente, das competências e das habilidades desenvolvidas pela equipe de projetos que são medidos os domínios no uso das práticas propostas pelos frameworks, metodologias de gestão e governança; neste contexto, as certificações são levadas em conta tanto para os profissionais como para as empresas (FREITAS, 2013).

## COBIT

O COBIT, criado pela ISACA, em 1996, com o objetivo de oferecer diretrizes para o desenvolvimento de negócio por meio do gerenciamento otimizado da informação, encontra-se, hoje, na versão COBIT 5, lançado em 2012, em um formato otimizado das versões anteriores e com uma atenção especial para as novas demandas de controle, sugerindo a utilização de outros *frameworks*, como ITIL, e contribuindo, ainda mais, com a segurança, a governança em TI e o gerenciamento de risco, trazendo confiabilidade e integridade (ROCHA; DINSMORE, 2015).

Ao pensarmos em uma forma de ilustrar a implantação do COBIT associada ao desenvolvimento de software, precisamos analisar 5 princípios. O **1º princípio** trata da necessidade de atender às necessidades dos stakeholders, ou seja, de se criar valor para toda e qualquer pessoa envolvida nos objetivos da organização. Esse processo inicia-se pelos proprietários da empresa ou pelos acionistas, que identificam as necessidades da empresa (ISACA, 2012).

Pense em uma instituição de ensino superior cujo valor que deseja entregar ao aluno é mais que um diploma e sim um conhecimento para sua vida profissional. Para tanto, o setor de TI tem um papel importante, considerando-se que a informação transita, hoje, por meio da tecnologia da informação. Pra isso, devemos pensar no **2º princípio**, o de cobrir a empresa fim a fim. Para tanto, faz-se importante implementar uma solução completa, desde o momento em que se busca a captação do aluno, oferecendo-lhe mais do que o diploma, mas profissionais capacitados e estrutura com tecnologia avançada, dentro e fora da instituição (VARGA, 2020).

Pronto, agora podemos falar do setor de TI. O aluno, para ingressar na instituição de ensino superior, precisa fazer vestibular, e hoje, já é possível que o aluno faça todo o processo de inscrição on-line. Após o vestibular, sua matrícula também poderá se dar

pelo mesmo método, sendo fundamental garantir a segurança da informação, além de uma estrutura física de servidores suficiente para atender à demanda.

Frente a isso, é necessário que a instituição pense em todo o processo, desde a captação do aluno até sua formação, gerando documentos claros a respeito de todo esse processo e verificando, de tempo em tempo, se há a necessidade de alteração em alguma etapa, sempre visando a permanecer à frente dos concorrentes e agregar facilidades ao seu cliente.

Para o **3º princípio**, é importante aplicar uma estrutura única e integrada; quanto mais simples for a estrutura, mais fácil será integrar as aplicações (ISACA, 2012).

Podemos verificar o quanto o setor de TI é importante para o processo, já que, a partir do ingresso do aluno à instituição, seus dados pessoais e acadêmicos passam a ser de responsabilidade da secretaria. O professor é responsável por agregar conhecimento e avaliar o aprendizado do aluno, no entanto, é interesse da coordenação e da direção verificar o bom andamento desse processo. Não se trata de saber a nota, especificamente, mas se aluno e professor estão satisfeitos com esse processo, buscando-se informações para as devidas melhorias, tanto para um como para outro. A secretaria se encarrega de informar os números quantitativos de aprovados, reprovados, desistentes e de mais situações, podendo, então, observar a necessidade de investimentos em determinadas áreas do ensino.

Diante disso, você pode notar o quão complexo é esse processo? Frente a isso, incluir processos de monitoramento é uma das sugestões que COBIT sugere para a busca de informações internas adequadas, a fim de se apontar onde devem ser inseridos os valores financeiros da empresa. Outro relatório relevante para ser analisado diz respeito às situações externas à instituição, à política, economia, atualização, concorrência, entre outros intempéries. Perceba quão complexa e volumosa é a quantidade de dados a serem analisados; de forma geral, todos podem vir do setor de TI.

No **4º princípio**, sobre habilitar uma abordagem holística, tratamos da comunicação e dos entendimentos integrados, portanto, para esse processo, analisamos os facilitadores que o COBIT sugere como fatores que, individualmente ou de forma conjunta, determinam se a nova proposta a ser implantada funcionará, fornecendo suporte à implantação de governança e ao sistema de gerenciamento de TI; sendo os facilitadores:

- Princípios, políticas e estruturas.
- Processos.
- Estrutura organizacional.
- Comportamento, cultura e ética.
- Informações.
- Serviços, infraestrutura e aplicações.
- Pessoas.
- Habilidades e competências (VARGAS, 2020).

Colocando na prática esse princípio, podemos refletir sobre a questão da implantação do sistema bibliotecário on-line, que é um sistema que visa garantir ao aluno o acesso a informações seguras por meio de literaturas reconhecidas e adquiridas, embasadas nas questões legais e éticas dos direitos autorais; no entanto, a aceitação por este processo não é de aderência coletiva, mas, sim, pela facilidade por meio de sites de pesquisa, os

quais permitem a cópia indiscriminadamente do conteúdo, mesmo sabendo das consequências pelo uso inadequado e indevido das informações e ainda sem a garantia da veracidade do conteúdo. Podemos observar, então, questões éticas, de princípios, comportamento e, se aprofundarmos, incluímos, praticamente, todos os facilitadores.

Quanto ao **5º princípio**, precisamos separar governança de gerenciamento, uma vez que são duas atividades distintas, ainda que se complementem e devem sempre estar alinhadas. O primeiro define os objetivos e o segundo define e controla a ação (ISACA, 2012). A governança com o COBIT fornece uma política transparente no controle da área de TI para o alcance dos objetivos da empresa, uma vez que possui metodologias que trazem métricas para avaliação de desempenho e resultados que auxiliam nas decisões e no gerenciamento de processos, reduzindo os riscos e agregando informações estratégicas.

O COBIT trabalha em um nível mais estratégico, garantindo a segurança para tomada de decisão, enquanto o gerenciamento da informação é o ponto central, descrevendo os princípios de suporte de uma empresa orientada ao mercado no que diz respeito aos seus recursos de TI e à utilização dos seus ativos. No decorrer dos anos, o COBIT foi ganhando formas e novas versões, sendo dividido em quadro domínios: planejamento e organização, aquisição e implementação, entrega e suporte e monitoramento e avaliação de desempenho (ROCHA; DINSMORE, 2015).

## ITIL

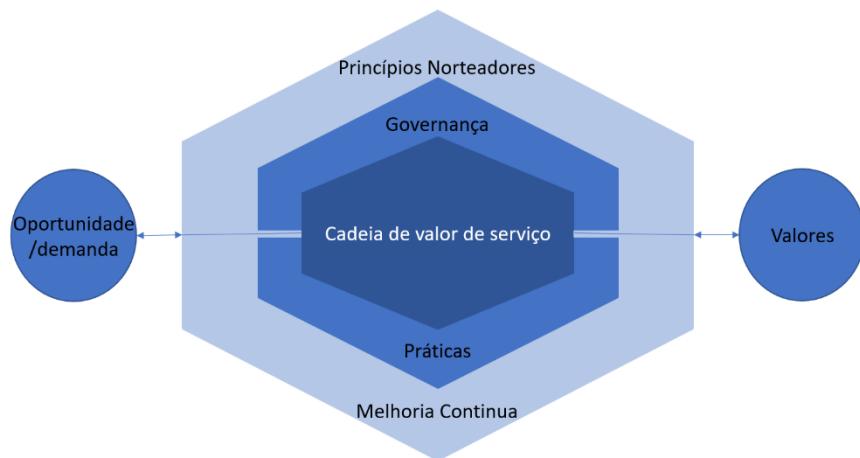
Desenvolvido no Reino Unido, em 1980, pela *Central Computing and Telecommunications Agency* (CCTA), o ITIL é um *framework* de recomendações para inúmeras empresas e profissionais a fim de gerenciarem os serviços de TI. Reconhecido e aplicado como modelo de gestão por organizações que prestam serviços de TI, também é utilizado nas áreas de TI interna das empresas, promovendo resultados significativos voltados à eficiência e ao controle dos serviços de TI. Permitindo um alinhamento entre os objetivos da TI aos da organização (FERREIRA, 2011).

Lançada a sua última versão, chamada ITIL 4, no início de 2019, ele traz uma abordagem evoluída, fornecendo ao leitor uma compreensão, de forma estruturada, do gerenciamento de serviços e de como evoluir para adotar tecnologias e formas de trabalhos modernas. Essa nova versão apresenta apenas um volume objetivo e simplificado. A proposta é ser um modelo flexível, que traz a combinação de seis atividades principais que visam a: planejar, melhorar, engajar, projetar e fazer a transição, obter/construir, fornecer e apoiar.

A ITIL 4 deixa claro que não anula os processos do ITIL V3. A nova versão tráz agilidade para que as organizações sejam mais eficazes e eficientes, conforme a necessidade de mudanças nas demandas dos *stakeholders*.

O Sistema de Valor de Serviço (SVS) vem com uma alteração significativa na estrutura ao transformar oportunidade/demandas em valor; para isso, tem como seus principais componentes no centro a cadeia de valor de serviço, a governança, a prática, em sua volta e os princípios orientadores e a melhoria contínua, englobando tudo, sugerindo uma estrutura de análise.

Figura 1.4 | Sistema de valor de serviço



Fonte: adaptado de AXELOS (2019).

ITIL 4 trabalha com o modelo de 4 dimensões: organização e pessoas; informação e tecnologia; parceiros e fornecedores e fluxos e processo de valor. No centro, sendo o foco do produto e serviço, estão os valores gerados pelas 4 dimensões. Em torno das 4 dimensões, estão os fatores que influenciam nas decisões no que tange a qualidade e mudança dos serviços, conforme ocorrem alterações externas como os fatores políticos, econômico, social, tecnológico, legal (leis) e ambiental (ISACA, 2019).

Um exemplo prático da aplicação dos conceitos e das práticas sugeridas pela ITIL na instituição de ensino que busca alinhar a qualidade ao atendimento do aluno no processo de solicitação de documentação juntamente à secretaria é a criação do serviço de protocolo eletrônico, que substitui a ida ao protocolo, otimizando o tempo do aluno e do funcionário do protocolo, ficando clara a importância do setor de TI no processo de otimização de serviço, melhorando a qualidade no sentido de agilizar um serviço que pode ser automatizado.

Nessa nova versão do ITIL 4, aparece um novo valor, que é o valor da cocriação, logo, a prestação de serviço de um profissional associado à sua organização agrupa valor à sua empresa (AXELOS, 2019).

É importante, também, que se observe as definições sobre governança e gestão. A governança provê as formas de controle com transparência, alinhando estratégia e gestão, enquanto que a gestão define, acompanha, mede e direciona o funcionamento da organização. O ITIL veio para gerenciar serviços, buscando proporcionar, principalmente: maior controle da gestão; aumento da satisfação dos usuários e dos clientes; redução do tempo das tarefas, das soluções dos problemas e dos custos operacionais (FREITAS, 2013).

## IMPLEMENTAÇÃO DO COBIT E DO ITIL NAS ORGANIZAÇÕES

A implementação do COBIT e do ITIL nas organizações proporciona benefícios para o setor de TI e ao negócio como um todo. Percebe-se que, a cada dia, as empresas estão mais dependentes da tecnologia, e o departamento de TI melhora a qualidade dos serviços, por consequência, todos os demais setores sofrem melhorias. Foi a partir da implantação de boas práticas e de *frameworks* de sucesso que a alta administração de certas organizações, tais como IBM, HP, Microsoft e Dell, notou mudanças na governança.

Uma vez que vivem de fornecer serviços e tecnologias, todas se utilizam desses recursos, ainda que não apliquem todas as ferramentas e/ou todos os processos (COSTA; PEREIRA, 2019).

Vale salientarmos que não existem apenas esses recursos que estamos citando neste material, porém são os mais difundidos nos mercados nacional e internacional, uma vez que estão em constante atualização, garantindo eficiência, qualidade nas atividades e visão holística ao ciclo de vida do processo de desenvolvimento de software.

Para complementar o estudo, é importante que busque conhecer metodologias como CMMI, *Scrum*, XP (*Extreme Program*) e PSDM; além disso, é fundamental que esteja sempre atualizado, de forma a atuar no mercado com desenvolvimento de softwares de qualidade, tendo em vista que fazer a gestão do desenvolvimento de um software é uma das tarefas mais complexas.

### EXEMPLIFICANDO

**Por que a gestão de desenvolvimento de software é considerada um tanto quanto complexa?** É comum a ideia de que o software não é tangível, dessa forma, o gerente do projeto não consegue visualizar o seu avanço, tendo de contar com uma documentação para um acompanhamento. Outro ponto que pode parecer complexo é que não há um processo padrão, pois cada software é único, mesmo aqueles que foram desenvolvidos e estão passando por um processo de evolução; dessa forma, o código sempre será criado do zero, o que impossibilita prever a ocorrência de falhas no sistema, podendo ser grave se se tratar de sistemas integrados ou de uma parte de um sistema maior.

Considerando esses fatos, não é difícil entender por que grande parte dos projetos de software acaba com atraso e/ou com um custo elevado (SOMMERVILLE, 2018).

Para você que está atuando no mercado de TI ou pensando em ingressar nessa área abrangente de conhecimento e informação, independentemente do setor de atuação e do tamanho do projeto em que está envolvido, bem como se está ou não utilizando as boas práticas do momento, serão exigidos um planejamento e uma organização bem apurada, sendo uma rotina comum de um gerente de projeto. Portanto, para evitar imprevistos, é preciso adotar uma estratégia adequada à sua realidade, pois as empresas modernas estão investindo em metodologias e técnicas para uma gestão inteligente, tentando garantir um bom andamento dos processos dentro do planejamento (SOMMERVILLE, 2018).

As três metodologias abordadas trazem uma humanização da tecnologia, e isso vem crescendo nos últimos anos. Profissionais que possuem certificações em ITIL, COBIT e

PMI, com a orientação do Guia PMBOK, estão sendo requisitados e valorizados pelo mercado. Dentro desse universo de evolução tecnológica que o planeta vem passando, a globalização, especialmente em serviços na área de TI, tem feito com que profissionais capacitados tenham destaque nas oportunidades, o que significa que você está alinhado com a evolução dos profissionais da área de Desenvolvimento de Sistemas por estudar este conteúdo e buscar se aprofundar nesses conhecimentos, bem como em certificações!

### FAÇA A VALER A PENA

#### Questão 1

Durante o desenvolvimento de um sistema, soube-se que um novo relatório não foi relatado pelo cliente no momento do processo de iniciação. Em casos como esse, o incidente, então, deve ser gerenciado de forma que o método e o procedimento padronizados no início do escopo do projeto sejam aplicados, gerando uma resposta eficiente e eficaz a partir de um relatório de incidentes, sendo, essa etapa, gerenciada continuamente.

Considerando o relato de uma situação de correção de falha e baseado em ITIL e PMBOK, classifique cada uma das afirmações como (V) verdadeira ou (F) falsa.

- ( ) Notamos que o procedimento de gerenciamento de incidentes do ITIL é incapaz de garantir que a normalidade na operação seja resgatada.
- ( ) O serviço deve ser reestabelecido o mais rápido possível, de forma que cause o menor impacto no negócio, no ITIL.
- ( ) O PMBOK sugere ferramentas visando a atender às mudanças que podem ocorrer no projeto, tornando, assim, o processo de alteração mais assertivo, com menor custo e tempo.

Assinale a alternativa que apresenta a sequência CORRETA.

- a. V – V – F.
- b. F – V – V.
- c. V – V – F.
- d. V – V – V.
- e. F – F – F.

#### Questão 2

A nova gestora de uma empresa optou por implantar a governança em TI utilizando o COBIT e o ITIL. Para tanto, ela precisou fazer um trabalho aprofundado junto aos acionistas e demais responsáveis pelo encaminhamento da organização. Para que a implementação fosse aceita, ela descreveu o cenário que a empresa se encontra e onde pretende chegar; mostrou os processos que seriam necessários por meio do *framework* do COBIT em um primeiro instante, já que é interessante dar um passo de cada vez, e está sempre averiguando o feedback, uma vez que, como nova responsável pela integração de toda a empresa, é necessário adquirir a confiança de todos, e isso inclui os funcionários mais tradicionais.

Em uma reunião de aprovação, ou seja, no processo inicial do projeto de implantação do conceito de governança, descreveu inúmeros cenários e solicitou aos participantes da reunião das inúmeras áreas que apontassem os problemas e propusessem soluções.

Assim, gerou um relatório, traçando os principais problemas, e a grande maioria apontava para a má utilização dos recursos que a empresa já possui e o péssimo serviço que era prestado por falta de uma documentação orientadora.

Considerando o contexto, avalie as afirmativas a seguir.

- I. O uso de métricas de desempenho não é previsto pelo COBIT, já que é um conjunto de práticas genéricas voltadas exclusivamente para a gestão do setor de TI.
- II. Pessoas e informações são os únicos recursos de TI identificados com importância no COBIT.
- III. Os níveis de maturidade são templates definidos nos processos de TI, sendo descritos e reconhecidos como possíveis situações atuais e futuras.
- IV. O COBIT visa a otimizar os investimentos e reduzir os riscos do setor de TI.

Considerando o contexto apresentado, assinale a alternativa correta.

- a. Apenas as afirmativas I, III e IV estão corretas.
- b. Apenas as afirmativas III e IV estão corretas.
- c. Apenas as afirmativas I e II estão corretas.
- d. Apenas as afirmativas II, III e IV estão corretas.
- e. Apenas as afirmativas I, II e III estão corretas.

#### Questão 3

Em uma reunião de gestores de uma companhia, houve uma discussão a respeito das necessidades da empresa, gerando, assim, um *brainstorm*. Mediante as inúmeras ideias e colocações de cada problema relatado, iniciou-se um processo de alinhamento desses problemas com o Negócio da empresa, classificando-os em grau de importância, ou seja, os que afetam diretamente o negócio e que precisam ser resolvidos o quanto antes, analisando-se todo o esforço necessário definindo o tempo, o custo e a qualidade exigida para cada um desses novos possíveis projetos. Dentro do custo, foram incluídos os recursos humanos, físicos e de informações já existentes na empresa por meio dos conceitos e métodos da gestão da governança já implantados na empresa, portanto, agora, ficou para a gestão a criação de todo os 5 processos do ciclo de vida do projeto.

A formação da equipe de projeto é um dos itens fundamentais em busca de se garantir o sucesso do projeto, por isso, faz-se necessário um estudo minucioso de quem será alocado em cada etapa do projeto.

Baseando-se no PMI, conforme o PMBOK, analise as afirmativas abaixo relacionadas ao grupo de processo de Iniciação.

- I. Não é necessariamente que todos os integrantes permaneçam do início ao final de um projeto.
- II. Os recursos humanos também podem ser alocados em momentos específicos, a fim de se maximizar a utilização do conhecimento do profissional dentro da empresa.

III. É realizada a definição dos membros da equipe levando-se em consideração, exclusivamente, a fase da Iniciação.

Considerando o contexto apresentado, assinale a alternativa correta.

- a. Apenas as afirmativas I e III estão corretas.
- b. Apenas as afirmativas I e II estão corretas.
- c. Apenas as afirmativas II e III estão corretas.
- d. Apenas a afirmativa I está correta.
- e. Apenas a afirmativa III está correta.

#### **REFERÊNCIAS**

AXELOS. **ITIL foundation**: ITIL 4 edition. London: AXELOS, 2019.

CANDIDO, R. et al. **Gerenciamento de projetos**. Curitiba: Aymará, 2012.

CARVALHO, F. C. A. de. **Gestão de projetos**. 2. ed. São Paulo: Pearson, 2018.

COSTA, A. B. da; PEREIRA, F. da S. **Fundamentos de gestão de projetos**: da teoria à prática – como gerenciar projetos de sucesso. Curitiba: Intersabers, 2019.

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

FERREIRA, P. A. dos S. **Implementação de processos da fase de operação de serviço do ITIL® em ambiente universitário**: o caso do ISCTE-IUL. 2011. Dissertação (Mestrado em Gestão de Sistemas de Informação) — Instituto Universitário de Lisboa. Lisboa. 2011. Disponível em: <http://hdl.handle.net/10071/2338>. Acesso em: 29 dez. 2020.

FREITAS, M. A. dos S. **Fundamento do gerenciamento de serviços de TI**. 2. ed. Rio de Janeiro: Brasport, 2013.

ISACA. **COBIT 5 Implementation**. Rolling Meadows: ISACA, 2012.

NEVES, P. S.; DONÁ, C. M. Avaliação da prestação de serviços de TI sob os modelos de governança de TI – um estudo de caso. **Brazilian Journal of Business**, Curitiba, v. 2, n. 1, 2020.

NEWTON, R. **Gestão de projetos**. 2. ed. São Paulo: Pearson Prentice Hall, 2011.

PINHEIRO, J. M.S. **Sarbanes-oxley e o impacto sobre a governança de TI**. 2007. Disponível em: <https://bit.ly/3audwas>. Acesso em: 29 dez. 2020.

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

REZENDE, L. H. S. Os impactos da Lei Sarbanes-Oxley Act e a governança corporativa no novo milênio. **EVS**, Goiás, v. 35, n. 4, 2008. Disponível em: <https://bit.ly/3u5U0sS>. Acesso em: 29 dez. 2020.

ROCHA, L.; DINSMORE, P. **EPG**: enterprise project governance. Governança corporativa de projetos. Rio de Janeiro: Brasport, 2015.

SOMMERVILLE, I. **Engenharia de software**. Tradução: Newton de Araújo Queiroz. 10. ed. São Paulo: Pearson Addison Wesley, 2018.

VARGA, S. **Contribuições para um sistema holístico de gerenciamento de infraestrutura de TI**. 2020. 168 f. Tese (Doutorado em Engenharia Elétrica) — Faculdade de Engenharia Elétrica e de Computação. Universidade Estadual de Campinas, Campinas. 2020.

VERAS, M. **Gestão da tecnologia da informação**: sustentação e inovação para a transformação digital. Rio de Janeiro: Brasport, 2019.

## ***FOCO NO MERCADO DE TRABALHO***

*MANIFESTO E MÉTODOS ÁGEIS PARA GERENCIAMENTO DE SOFTWARES*



Fonte: Shutterstock.

## Deseja ouvir este material?

### SEM MEDO DE ERRAR

O primeiro passo, independentemente da metodologia escolhida, é analisar todo o contexto e a trajetória de vida da empresa.

Dessa forma, você deverá entrevistar e observar o trabalho da equipe já existente na empresa. O objetivo é elencar as etapas de desenvolvimento que estão sendo utilizadas antes de você implantar a nova metodologia.

Posteriormente, você deverá observar como os processos são desempenhados e qual é o perfil dos *stakeholders* (não se esquecendo de conhecer, também, os clientes da empresa).

Em paralelo, você deverá realizar uma análise de mercado, que envolve os principais concorrentes da empresa em que trabalha, a fim de se certificar quanto à escolha da metodologia. No seu caso, a empresa indicou a XP, então, após essa análise, você já poderá planejar como contribuir para que a empresa atinja seus objetivos.

Outro desafio que você vai enfrentar é a mudança de cultura organizacional. Apesar de parecer simples trazer vantagens da utilização da metodologia ágil, esse, provavelmente, será seu maior desafio.

Mas tudo é possível, então, busque aproximar as pessoas da equipe para alinhar os objetivos delas aos que estão sendo solicitados a você nesse trabalho. Lembre-se de que as principais atividades que compõem o ciclo do XP são: planejamento, projeto, codificação e testes.

Durante o planejamento, você pode observar como os requisitos estão sendo levantados e documentados, uma vez que, no XP, a equipe os insere em cards como histórias do usuário. Quanto à codificação, como você pode implementá-la em pares na sua nova equipe? A metodologia XP defende a integração contínua e a refatoração dos códigos, visando a trazer qualidade ao sistema. Como você pode implementar essas ideias? Suas ações podem variar bastante, já que você está no comando para a realização dessa mudança! Para facilitar sua resolução, utilize os conceitos do Kanban, como o exemplo abaixo:

Quadro 2.1 | Conceitos do Kanban

Funcionalidades que devem ser realizadas	Funcionalidades que estão sendo realizadas	Cada
<ul style="list-style-type: none"><li>• Cadastrar dados.</li><li>• Realizar login no sistema.</li><li>• Alterar dados.</li><li>• Realizar transações financeiras.</li></ul>	Realizar login no sistema.	

Fonte: elaborado pela autora.

Outro exemplo que você pode seguir é o de desenvolver a história do usuário, como mostrado no exemplo.

Quadro 2.2 | Exemplo de história do usuário

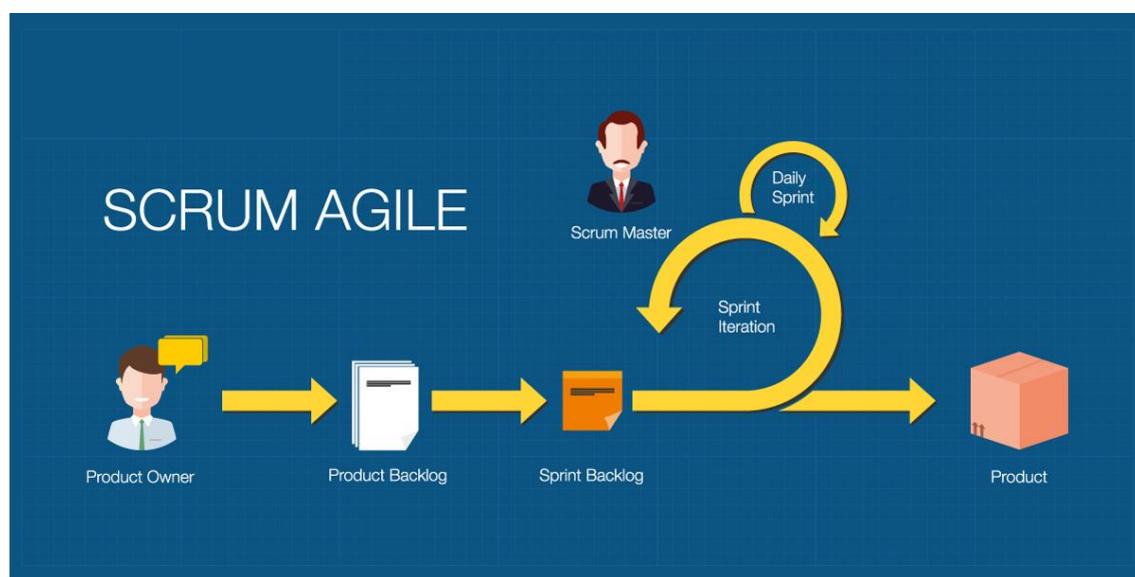
Título da história do usuário: realizar login no sistema	
Descrição:	
	Eu, enquanto usuário, gostaria de realizar login no sistema, inserindo meu CPF como login e minha data d
Fonte:	elaborado pela autora.

Ao alinhar essas atividades com os objetivos da empresa, você conseguirá atingir seus objetivos de forma positiva e resultados notórios.

## Unidade-2 seção-2

# *NÃO PODE FALTAR*

## *DESENVOLVIMENTO DE MÉTODOS ÁGEIS UTILIZANDO SCRUM*



Fonte: Shutterstock.

**Deseja ouvir este material?**

## **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, adicionando novos requisitos do sistema e avaliar suas iterações.
Entrega incremental	O software é desenvolvido em incrementos com o cliente, especificando o que é feito em cada incremento.
Pessoas, não processos	As habilidades da equipe de desenvolvimento devem ser reconhecidas e elas devem ser estimuladas a trabalhar juntas para 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, prepare-se para lidar com mudanças.
Manter a simplicidade	Focalize a simplicidade, tanto do software a ser desenvolvido quanto do processo de desenvolvimento. Tente sempre fazer o que é necessário e nada mais.

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 1970; 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, transparência, 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:

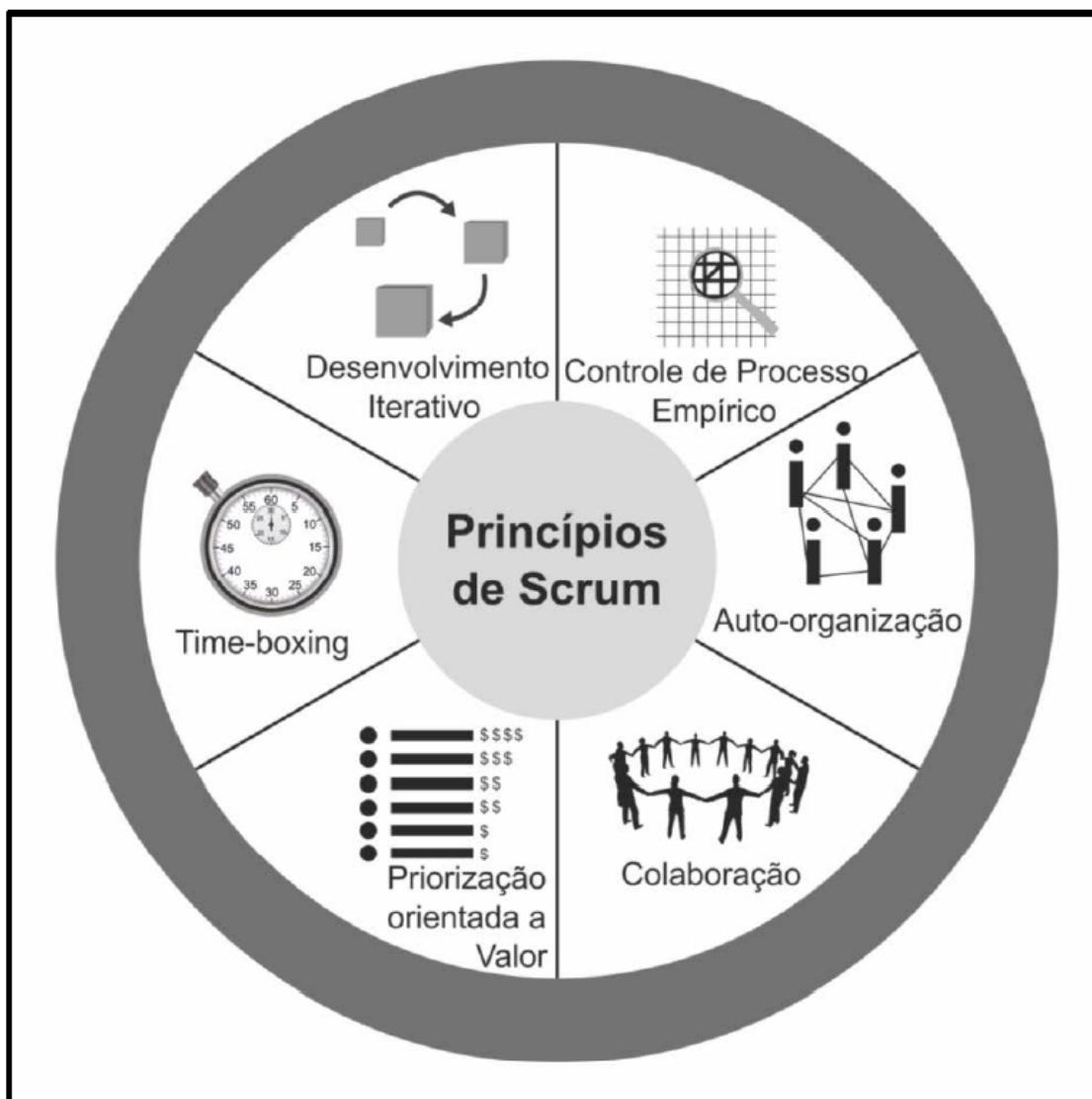
- As equipes podem ser chamadas de "*Times Scrum*". Conforme Cruz (2020, p. 41-42), “o Time Scrum é composto por três papéis: Scrum Master, Product Owner e Time. [...] O Time deve ser interdisciplinar, e geralmente quando é utilizado só esse termo, estamos nos referindo aos desenvolvedores”.
- O indicado 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.
- 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.

Figura 2.4 | Princípios do Scrum



Fonte: SCRUMstudy (2016, p. 8).

#### 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 product owner):</b>	Tem como função trazer uma perspectiva sob o ponto de vista do cliente e cliente quer para aquele produto que será desenvolvido. Ele deve possuir 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 para que o projeto deve atingir. Também é de sua responsabilidade verificar se os conceitos 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.

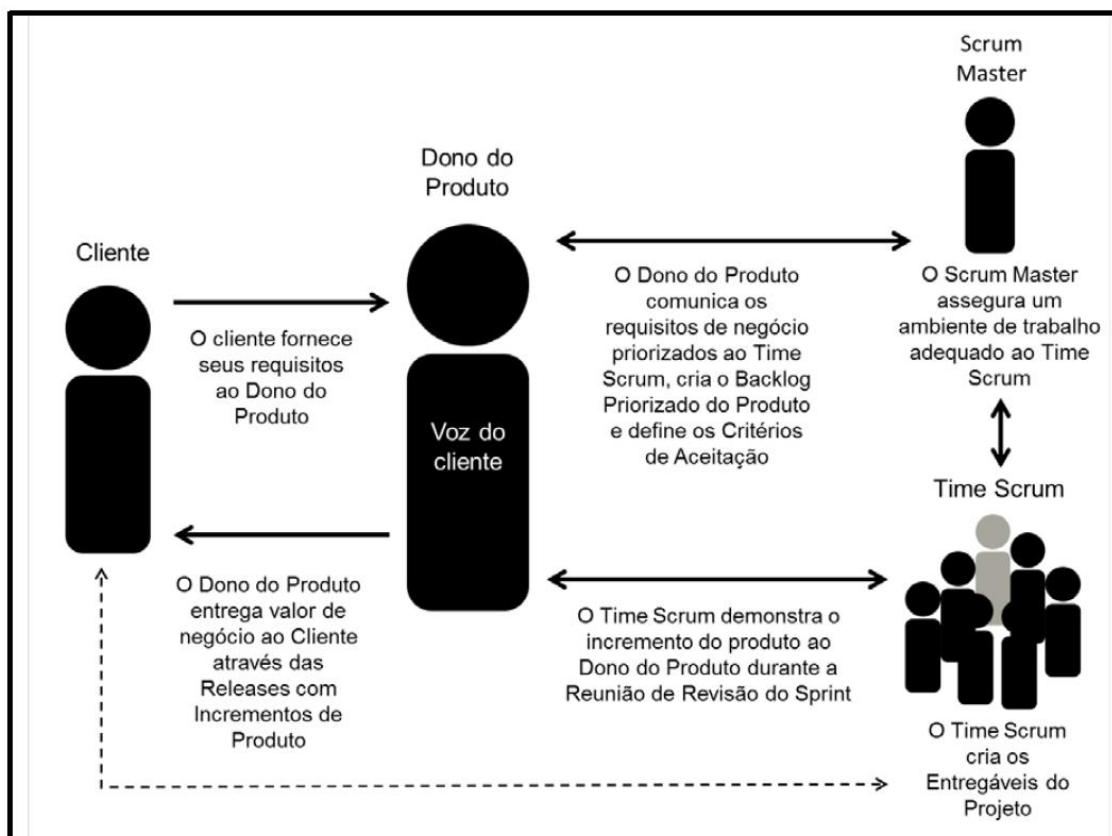
## FASES SCRUM

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).

## 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*.

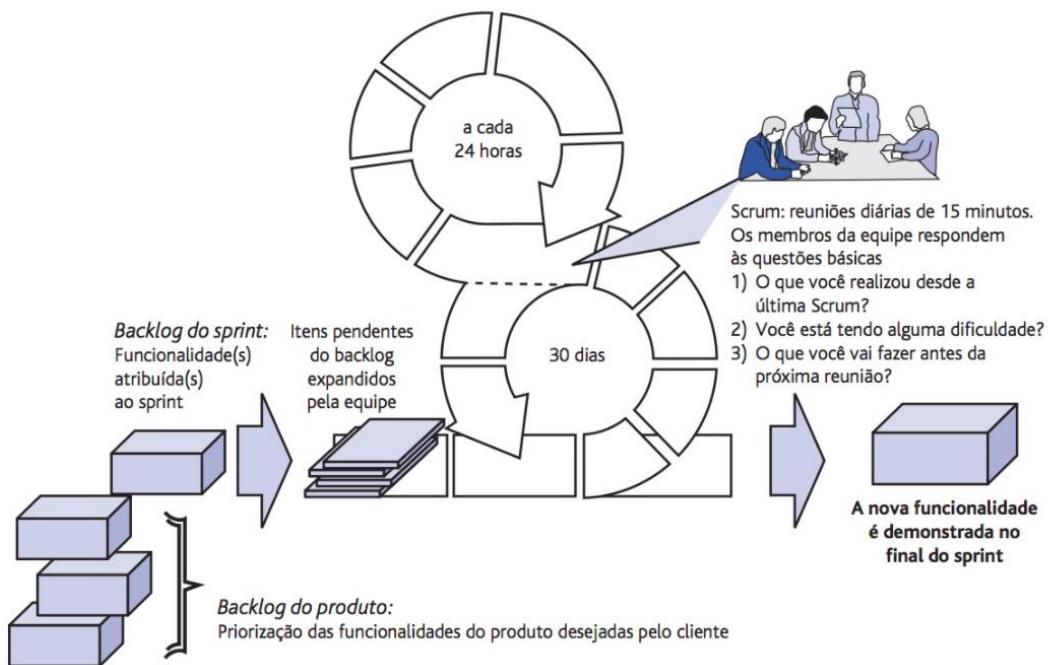
Figura 2.5 | Organização do *Scrum*



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*



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**.

- 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*.

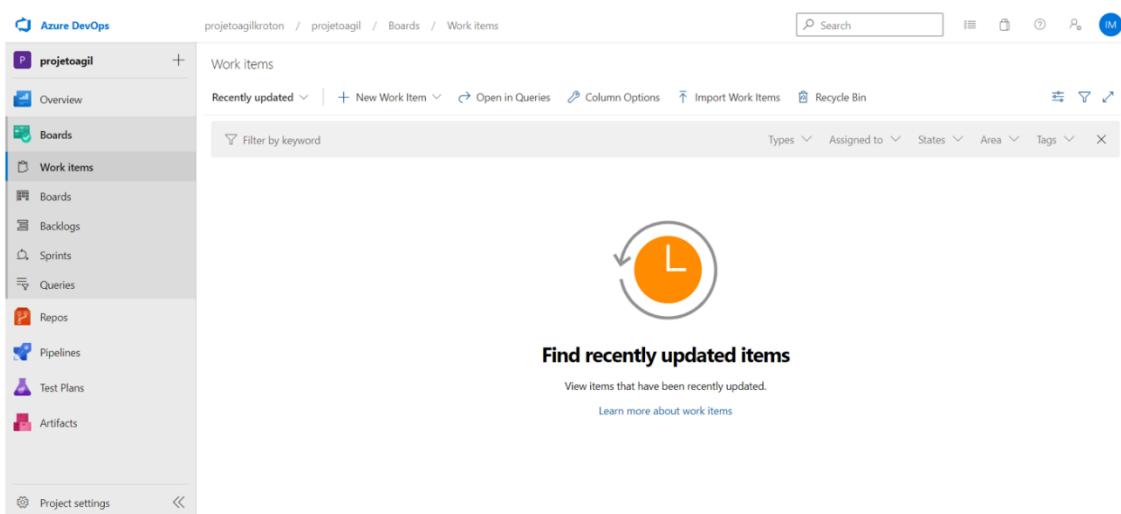
## EXEMPLIFICANDO

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



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

SOBRE VOCÊ > SOBRE O SEU TRABALHO > CONFIGURE O SEU PRIMEIRO PROJETO

Como você está gerenciando o trabalho atualmente — em casa, no escritório ou onde quer que esteja?

Escolha uma ou várias opções a seguir.

- Planilhas, documentos, plataformas de slides**
- Cartões ou quadros no estilo Kanban ou Agile**
- Gráficos de Gantt ou cronogramas visuais**
- Anotações feitas em papel ou listas de tarefas**

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.

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.

## ***NÃO PODE FALTAR***

***ABORDAGENS INOVADORAS E ÁGEIS PARA A GESTÃO DE SOFTWARE***



Fonte: Shutterstock.

**Deseja ouvir este material?**

### ***PRATICAR PARA APRENDER***

Prezado aluno, começamos este conteúdo desafiando você a pensar em pelo menos uma atividade que você executa em seu cotidiano, pode ser uma bem simples, como tomar café da manhã, ou escovar os dentes. Você pode estar se perguntando o que isso tem a ver com software, pois bem, para completar seu desafio, liste alguma outra atividade que você automatizou por meio de um sistema. Vou citar um exemplo: acesso a aplicativos de banco ou de compras de alimentos, remédios, entre outros produtos.

Hoje, provavelmente, isso se torna possível devido à evolução dos recursos tecnológicos e dos processos criativos que existem em torno de suas criações, com isso, o design recebeu mais um aliado, chamado *design thinking*, que é um conceito que pode ser aplicado em várias áreas de atuação, e com a tecnologia não seria diferente: ele atua trazendo inovação e uso de métodos e ferramentas para provê-la. Como resultado, é possível obter o que chamamos de Mínimo Produto Viável (MVP), que é a versão simplificada de algo, em que é possível avaliá-lo em vários aspectos, como os técnicos

de design do produto. Além disso, contempla a possibilidade do uso dos conceitos de pivotagem, em que novas perspectivas sobre o produto são lançadas para serem avaliadas e, até mesmo, postas em prática, sempre com o objetivo de elevar a qualidade e o valor que aquele produto ou serviço gera ao cliente.

Diante disso, relembrar as atividades cotidianas que você listou e imagine novas formas de realizá-las. Por meio dos conceitos aqui aprendidos, você terá grandes chances de conseguir realizar esse desafio, já que terá tempo suficiente para compreender os princípios do *design thinking*, que são: empatia, colaboração, definição de planejamento e, consequentemente, ações sequenciais bem definidas, prototipação, testes, entre outros, uma vez que podem variar conforme o contexto.

O universo do entretenimento ganha, a cada ano, novos tipos de eventos. O objetivo é agradar e chamar a atenção de públicos diversos. Dessa forma, quem trabalha nesse tipo de mercado deve se atualizar quanto às inovações que surgem e que envolvem setores artísticos e tecnológicos, tendo em vista o aumento de uso desses recursos que acaba agregando valores diferenciados à experiência do usuário.

Anualmente, ocorre um festival de música composto por diversos artistas e várias modalidades de apresentações. Como gestor de uma equipe responsável pela organização do evento, ficou a seu cargo apresentar inovações que poderão ser adicionadas à programação predefinida pelos investidores durante a etapa de planejamento inicial do evento.

Com base em seus conhecimentos em projeto de software, você logo percebeu que a tecnologia seria fator-chave para atender a essa demanda; além disso, viu a necessidade de criar um portfólio para apresentar cenários em que a tecnologia poderá ser utilizada para proporcionar novas experiências ao público do evento. O *design thinking* é composto por conceitos e uso de ferramentas e metodologias que podem ser executadas em suas principais etapas, como as de imersão, análise e síntese, ideação e prototipação. Para tanto, você vai usar o **Storyboard** para expor suas ideias sobre a execução da implantação do sistema na fase de prototipação.

Posteriormente, compartilhe seus materiais na nuvem (*Google Drive*, *OneDrive*, *dropbox*, entre outras opções). O objetivo é que os demais colegas possam visualizar suas soluções para a situação apresentada e também contribuir com suas ideias para melhorar sua solução.

Aproveite os conhecimentos adquiridos sobre os temas trabalhados e exerçete a aplicação de outras opções de soluções para o caso.

Este é o momento para colocar sua imaginação para solucionar situações cotidianas, suas e também de outras pessoas. Além disso, lembre-se de que você é o protagonista de seu aprendizado.

#### **CONCEITO-CHAVE**

A tecnologia geralmente é utilizada para solucionar ou prestar suporte à realização de alguma atividade específica. Pense que o presente acaba sendo uma evolução de tudo

que ocorre no nosso dia a dia, ou seja, quando falamos de “software”, podemos afirmar que sua evolução se deu ao longo dos anos, e o seu objetivo sempre foi atender a uma demanda que a sociedade demonstra.

## CONCEITO DE DESIGN

O conceito de design é utilizado por apresentar uma perspectiva direcionada ao aperfeiçoamento e até à criação de coisas novas. Conforme Ambrose *et al.* (2011), envolve um ponto de vista detalhado e criativo, porém direcionado pelo processo, que contribui para a concepção de soluções que atendam às demandas definidas na etapa inicial, uma vez que está relacionado a algumas ações, como definir público-alvo, e ao problema que deverá ser solucionado.

Posteriormente, haverá uma etapa de pesquisas, a fim de se analisar o contexto desse problema e para que ideias sejam geradas. Nesse ponto, motivações e novas necessidades poderão ser visualizadas. Além disso, haverá a possibilidade de criação de protótipos, em que as ideias podem ser apresentadas de diversas formas, dando a possibilidade da seleção e implementação da que melhor se adequar às necessidades do cliente.

## DESIGN THINKING

Dentro desse universo tão amplo do design, surge o *design thinking*, trazendo uma abordagem que caracteriza uma nova forma de pensar e que pode ser aplicada em qualquer área de atuação, já que usa a sensibilidade do design, os métodos e as ferramentas que, juntos, visam a proporcionar inovação. Nesse contexto, seu uso será voltado à tecnologia e ao desenvolvimento de serviços e produtos da área.

Aliás, existe uma definição voltada ao design de serviço que, conforme Stickdorn *et al.* (2014), tem o propósito de planejar experiências por meio da utilização de recursos definidos. Com isso, é possível presenciar pontos positivos relacionados à experiência proporcionada ao usuário. Podemos mencionar, também, os aplicativos para solicitação de veículos ou de entrega de alimentos e outros tipos de produtos, proporcionando, ao mercado do e-commerce, diversos serviços.

Frente a isso, o *design thinking* pode ser visto como a junção do *mindset* e um plano de ação, agregando benefícios e possibilidades de transformar o modo como as empresas se empenham no desenvolvimento de seus processos, serviços, produtos e até estratégias dentro do mercado em que atuam. Compreende-se *mindset* como a forma com que organizamos nossos pensamentos, principalmente voltados ao cotidiano.

Ainda sob a ótica de Stickdorn *et al.* (2014), trata-se de uma combinação entre design, gestão e engenharia de processos, bem como possui alguns princípios que alinharam o design e seu direcionamento ao desenvolvimento de serviços. Entre eles, podemos destacar:

- O usuário como centro de tudo, ou seja, sua participação e perspectiva do que está sendo desenvolvido e o que é obtido como produto final, seja de cada iteração ou ao final de todo processo, são relevantes, pois trazem **empatia** e

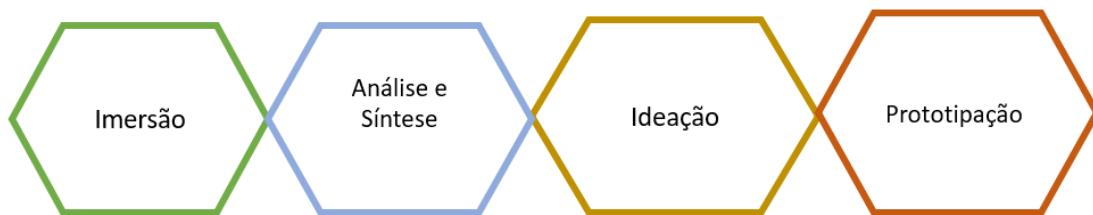
sintonizam todos da equipe, a fim de que haja **definição** de um planejamento voltado a atender aos objetivos vistos.

- Processo criativo coletivo, em que todos da equipe participam do momento de design, trazendo **colaboração**.
- Relacionamento entre **ações sequenciais predefinidas**, pois uma ação acaba contribuindo para outra.
- Todos os aspectos do ambiente são levados em consideração, proporcionando cenários para experimentação da aplicação de novas ideias, podendo, até mesmo, haver uso de **prototipação** para se explanar e **testar** a viabilidade das ideias.

### **FERRAMENTAS, TÉCNICAS E MÉTODOS**

Para atender a esses princípios, é possível fazer uso de ferramentas, técnicas e métodos que auxiliam a todos os envolvidos no projeto a terem transparência na visualização de todos os cenários ativos. Lembre-se de que o processo criativo durante o desenvolvimento de algum produto ou serviço, normalmente, requer uma equipe multidisciplinar, em que cada um contribui da melhor maneira possível, trazendo sua experiência diante de suas habilidades. Entre as diversas possibilidades, é possível mencionar o uso de **algumas etapas** utilizadas durante o processo criativo:

Figura 2.15 | Etapas do *design thinking*



Fonte: elaborada pela autora.

### **IMERSÃO**

A **fase de imersão** é o momento inicial do projeto e envolve a aproximação da equipe com o problema que o cliente retrata. Trata-se de uma fase de extrema importância, pois a situação é vista por meio da descrição do cliente e do usuário final, ou seja, o cliente do cliente.

Conforme Vianna et al. (2012), pode apresentar dois momentos, um de forma **preliminar** e outro de **profundidade**. Na preliminar, ocorre o uso de algumas estratégias, tais como:

- **Reenquadramento**: ocorre por meio da equipe de desenvolvimento e de representantes da empresa do cliente. O objetivo é alinhar as ideias sobre o que vem a ser o problema do cliente e como o produto ou serviço que será desenvolvido deverá atendê-lo. Envolve etapas de captura (por meio da coleta de dados obtidos em reuniões ou momentos de coletividade), transformação (por meio do mapeamento dos dados e de demonstração a partir de mapas mentais) e

preparação (o objetivo é causar sensibilização por meio da demonstração das informações obtidas anteriormente).

#### EXEMPLIFICANDO

Existem diversas técnicas para expor as ideias a todos da equipe. Entre elas, é possível mencionar o uso de mapa mental, que demonstra, por meio de uma representação gráfica, ideias e conceitos ligados a ela.

Para ajudá-lo a organizar as ideias por meio do mapa mental, existem diversas ferramentas, entre elas, a XMind, que é utilizada para o *brainstorming* e foi desenvolvida pela XMind Ltd. Veja o exemplo abaixo, em que um termo que possui maior destaque foi colocado no centro e as demais ideias relacionadas a ele foram sendo inseridas ao seu redor.

Figura 2.16 | Exemplo de mapa conceitual



Fonte: elaborada pela autora.

- **Pesquisa exploratória:** envolve não só elementos internos, mas também externos. Por exemplo: o sistema a ser desenvolvido será voltado ao comércio de alimentos, como os supermercados, e o problema do cliente é a automatização de toda sua logística, desde fluxo de caixa, até recursos humanos, frotas etc. Então, por meio da pesquisa exploratória, é possível estudar também o que envolve esses setores. O de cargas e abastecimento está relacionado até mesmo ao fornecimento da matéria-prima; o de frotas está envolvido com a manutenção dos veículos, e assim por diante. Note que outros aspectos são abordados e, por mais que pareçam ser de cunho apenas externo, acabam afetando o funcionamento interno do negócio.
- **Pesquisa Desk:** esse tipo de pesquisa é fundamental para adequar as ideias do *design thinking*, já que está relacionada à busca de referências, tanto nacionais quanto internacionais, afinal de contas, estamos em um mundo globalizado, que sofre influências de todos os lugares. Logo, quando lidamos com inovação, essa ação é primordial para nos alinharmos às tendências globais.

Por meio dessas atividades, é possível apresentar as restrições que o projeto terá, qual é o perfil do usuário e em quais temas o próximo passo deverá se aprofundar, pois, seguindo a linha de pensamento do exemplo mencionado anteriormente, caso seja um projeto para sistema de supermercado, é preciso limitar o escopo para que não se perca tempo aprofundando em conhecimentos desnecessários para o momento.

Por falar em um próximo passo, com as informações obtidas inicialmente, é possível executar a etapa denominada **imersão de profundidade**, em que serão aproveitados os

resultados documentados anteriormente. Para isso, serão utilizadas técnicas multidisciplinares, como plano de pesquisa, documentação dos perfis de usuários que foram elencados anteriormente etc.; identificando-se as reais necessidades apresentadas pelos envolvidos e as novas oportunidades.

Para obtenção e organização dessas informações, são utilizados alguns instrumentos:

- **Entrevistas:** as entrevistas ocorrem no intuito de se fazer com que o entrevistado exponha, de maneira espontânea, os detalhes sobre as necessidades que o projeto deve atender. Por exemplo, podem ocorrer entrevistas com profissionais que desempenham funções semelhantes em vários locais, a fim de se obter informações a respeito de como as mesmas operações podem ocorrer em contextos diferentes.
- **Cadernos de sensibilização:** possui um objetivo bem semelhante ao da entrevista, porém com a intenção de fazer com que o próprio usuário faça reflexões sobre si. No contexto do desenvolvimento de software, é possível exemplificar sua aplicação por meio da percepção de que a reflexão do cliente pode trazer facilidades no momento da obtenção dos requisitos que o sistema deve ter.
- **Sessões generativas:** essas sessões podem ocorrer após o usuário ter utilizado seu caderno de sensibilização. Note que são técnicas que envolvem abordagens diferentes das que podem ser vistas como mais formais, ou seja, o usuário, que também pode ser visto como cliente, acaba expondo informações de forma indireta, por meio de anotações e de sessões, como a generativa. O maior objetivo dessas técnicas é trazer reflexão para que seja possível definir o que realmente importa ao sistema.

## ANÁLISE E SÍNTSE

É importante salientarmos que a **análise e síntese** é um processo que ocorre durante todo o ciclo, tendo em vista que, em determinados momentos, pode ser capaz de retratar soluções novas e melhorias no que está sendo desenvolvido.

- **Cartões de insights:** São utilizados para identificação de padrões e relação entre os dados já obtidos.
- **Diagrama de afinidades:** é utilizado para identificar ligação entre os temas e as áreas relacionadas ao projeto. Geralmente, esses cartões ficam disponíveis para que a equipe consiga visualizar temas e pontos que podem ser relevantes para o projeto.
- **Mapa conceitual:** note que, nessa etapa, as abordagens envolvem utilizar a representatividade visual, e o mapa conceitual não é diferente; ele traz informações em forma de gráfico e é muito utilizado para auxiliar na organização das informações. O início de um mapa conceitual é realizado por meio da escolha de termos que fazem parte de contextos que envolvem o projeto. A figura 2.4 exemplifica um mapa conceitual.

Figura 2.16 | Exemplo de mapa conceitual



Fonte: elaborada pela autora.

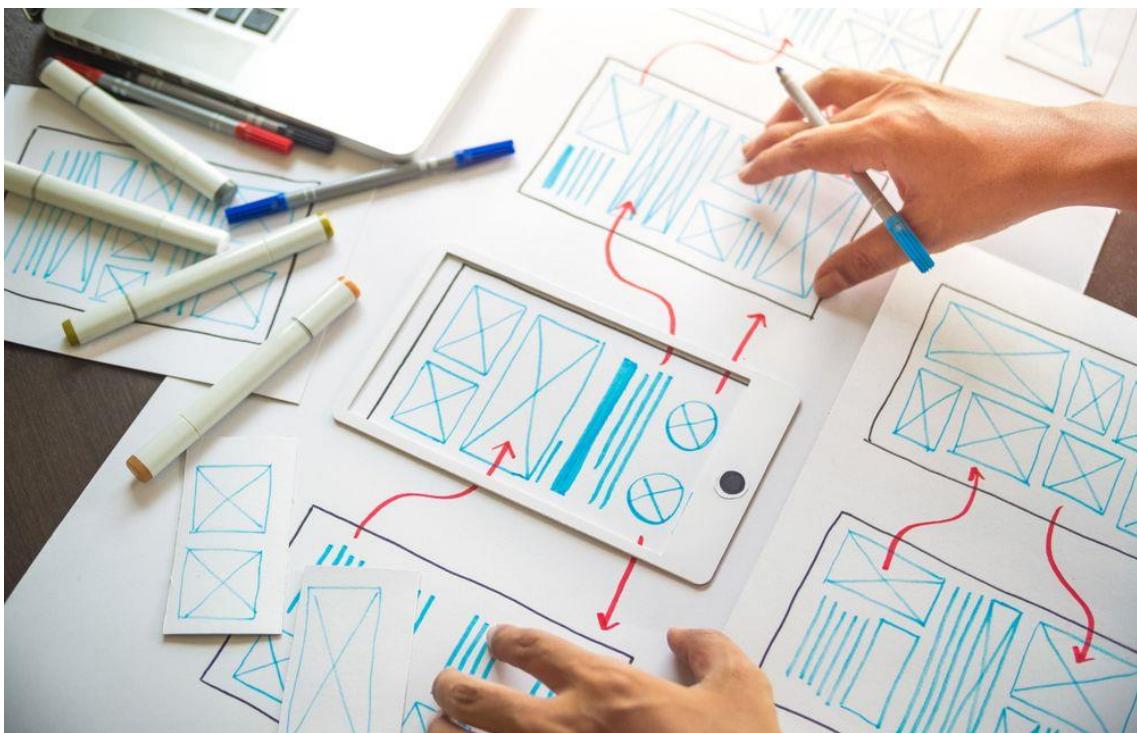
- **Critérios norteadores:** os critérios são importantes, pois visam a garantir que as informações mais relevantes façam parte no projeto. Perceba que cada abordagem documentada traz referências importantes para o projeto de forma diferente e, muitas vezes, uma complementa a outra.
- **Personas:** são personagens fictícios, de acordo com Stickdorn *et al.* (2014), e também podem representar grupos específicos, com perfis semelhantes, e que, em uma perspectiva macro, podem ser diferentes ao extremo. O objetivo é poder mapear possibilidades diferentes para o sistema. Por exemplo: caso queira desenvolver um aplicativo para um canal de filmes, você deve obter informações sobre os tipos de perfis que usarão sua aplicação. Você pode desenvolver personas com diferentes perfis, como idade, sexo, nacionalidade, entre outras características que tornam os usuários diferentes uns dos outros.

## IDEAÇÃO

Neste momento, a equipe expõe suas ideias; é um momento conhecido como ***brainstorming***, em que cada um colabora com suas perspectivas sobre o projeto. As informações reunidas nesse momento são organizadas por meio do uso de algumas práticas.

- **Workshop de cocriação:** é um momento em que se pensa nas soluções para o projeto; ele pode ser compartilhado entre os clientes e usuários.
- **Cardápio de ideias:** após algumas fases, ideias foram discutidas e, nesse momento, ideias sobre oportunidades de inovação e de novos negócios podem ser organizadas. A abordagem recebe esse nome pois podem ser disponibilizadas como cartas, menus etc.
- **Matriz de posicionamento:** essa ação analisa as ideias de forma estratégica para se validar o que foi gerado anteriormente, como as personas, caso tenham sido criadas.

Figura 2.17 | Fase de prototipação



Fonte: Shutterstock.

## ■ PROTOTIPAÇÃO

A prototipação oportuniza a utilização da representação visual de todos os conceitos obtidos nas etapas anteriores. Além disso, ela é utilizada para consagrar se algo faz ou não sentido. Muitas vezes, é nesse momento que fica mais claro se as soluções definidas realmente fazem sentido.

Por exemplo: suponha que a inserção de uma funcionalidade que promove, para o usuário, o retorno do comprovante de uma compra tenha sido definida ao longo do processo para ocorrer em forma de mensagem para o número do celular cadastrado, porém, ao se realizar a prototipação dessa funcionalidade, ficou nítido que ela não fazia muito sentido, tendo em vista que, atualmente, as ferramentas concorrentes utilizam a funcionalidade de outra forma, por meio da opção de compartilhamento via e-mail ou rede social, como mostra a Figura 2.6, em que desenhos, ilustrações, gráficos e simulações trazem essas ideias de forma visualmente mais clara, saindo do abstrato das descrições.

Os protótipos podem ser desenvolvidos com o uso de materiais simples, como papéis e canetas, ou com softwares, que auxiliam por meio da automatização do desenho de telas, botões, entre outros elementos gráficos que podem estar presentes em uma tela.

Outra opção é a elaboração de um **storyboard** cujos retratos se dão por meio de desenhos e descrições sequenciais com a naveabilidade das telas e que podem representar ações do usuário ao utilizar tal aplicação. Por exemplo: ao clicar em um botão, o usuário é direcionado à tela que traz outra funcionalidade, como quando clicamos em “ok”, em uma tela de login, e somos direcionados para outra página.

Conforme De Sá *et al.* (2017), os processos de design (imersão, análise e síntese, ideação e prototipação) não seguem ações lineares, dessa forma, adota uma iteratividade

em suas abordagens, possibilitando a criação de estruturas diversas aos projetos, tendo em vista que a própria análise e síntese pode ocorrer durante as demais etapas.

### MÍNIMO PRODUTO VARIÁVEL (MPV)

Caro aluno, note que todos os processos e as práticas mencionados anteriormente podem ser aplicados em paralelo com as metodologias voltadas ao projeto de um software, independentemente se segue a filosofia ágil ou tradicional. Além disso, existe um artefato denominado **protótipo de alta fidelidade**, que é desenvolvido com o auxílio de ferramentas e pode assumir aparência e funcionalidade próximas ao produto final, bem como ser visto como o um **Mínimo produto viável** (MPV ou, em inglês, *Minimum Viable Product*).

Sob a perspectiva de Massari (2016, p. 111), o MPV “corresponde ao conjunto mínimo de funcionalidades de um produto.” Por exemplo: tratando-se de um projeto de sistema para automatizar as principais atividades de um banco, quais as funcionalidades mínimas para que esse software possa ser utilizado? Nesse caso, ele pode proporcionar ao cliente a possibilidade de visualização de saldo, de realização de transferências e pagamentos, bem como validar e aprovar tudo que foi produzido anteriormente, trazendo, por exemplo, a relevância de adicionar, para cada transação, o pedido de senha ao usuário.

Além disso, por meio das conclusões que podem ser tomadas por meio do acesso ao MPV, pode ocorrer o que se chama **pivotagem**, ou seja, um processo de grande mudança, pois, por mais que várias estratégias tenham sido utilizadas durante o desenvolvimento de um sistema, muitos fatores podem ocasionar mudanças. Mas entenda que toda mudança deve ser analisada, independentemente da metodologia que está sendo utilizada.

Sabbag (2018) diz que sempre ao término de cada ciclo, a equipe deve tomar uma decisão entre pivotar ou perseverar na ideia que veio sendo seguida. Por exemplo, pode acontecer que, após um tempo de uso, um sistema não seja mais adequado, e essa percepção pode acontecer na fase de testes, logo, cabe aos envolvidos decidir se devem seguir com os planos iniciais ou realizar mudanças.

### REFLITA

A sociedade atual associa às suas práticas o conceito da multidisciplinaridade, e no projeto de software não é diferente. Quando há a necessidade de inovar produtos e serviços, o *design thinking* apresenta conceitos bastante relevantes para isso, porém não precisa ser aplicado só. Em diversas situações, ele conta com recursos complementares, como é o caso da metodologia Canvas, que possui uma perspectiva de negócio; além disso, agrupa, com recursos visuais, os detalhes mais relevantes para situações diversas a partir de 4 áreas importantes: clientes, oferta, infraestrutura e viabilidade financeira.

Portanto, é possível notar a importância da escolha de metodologias e abordagens para cada etapa do processo de desenvolvimento de software. Até mesmo porque, atualmente, esse recurso proporciona serviços e inovação de produtos que podem automatizar nossas atividades, das mais simples até as mais complexas.

## FAÇA VALER A PENA

### Questão 1

O design, quando inserido em um contexto social, visa a trazer adaptação e criação de produtos, serviços e ações estratégicas; dessa forma, surgiu outro termo, denominado *design thinking*, que agrega aos pontos citados anteriormente o fator de inovação. Este, por sua vez, possui algumas etapas que norteiam o desenvolvimento do seu processo criativo, entre elas, é possível mencionar a imersão.

Sobre as práticas que podem ser executadas na etapa de imersão, assinale a alternativa correta:

- a. Reenquadramento.
- b. Cartões de insights.
- c. Diagrama de afinidades.
- d. Critérios norteadores.
- e. Personas.

### Questão 2

As estratégias de ação estão sempre sendo utilizadas com o objetivo de trazer inovação aos processos de desenvolvimento de sistemas, produtos e serviços, em que as principais etapas do *design thinking* são: imersão, análise e síntese, ideação e a prototipação.

Considerando as informações apresentadas, analise as afirmativas a seguir:

- I. Os cartões de *insights* são estratégias utilizadas na etapa de imersão.
- II. O reenquadramento é uma estratégia que pode ser utilizada na fase de imersão.
- III. A pesquisa exploratória é utilizada na fase de prototipação.
- IV. As sessões generativas fazem parte da imersão no processo de *design thinking*.

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

- a. I e II, apenas.
- b. II e IV, apenas.
- c. I, II e III, apenas.
- d. II, III e IV, apenas.
- e. I, II, III e IV.

### Questão 3

[...] talvez seja válido observar que, independentemente da intencionalidade do processo, ele terá grande importância para o resultado final do design. A vantagem de articular claramente o processo de design é que isso permite um maior grau de reflexão sobre a influência que o designer teve sobre o resultado do processo de design.  
(STICKDORN, 2014, p. 128)

Sobre os princípios do *design thinking*, assinale a alternativa correta:

- a. A fase de análise e síntese é a fase inicial do projeto e envolve o momento de aproximação da equipe com o problema que o cliente retrata.
- b. A fase de prototipação ocorre após a imersão; além disso, pode ser aplicada em outras etapas do projeto e utiliza a técnica de cartões de insights.
- c. A fase de imersão pode ocorrer de forma preliminar e de profundidade, utilizando abordagens diferentes para obter as informações necessárias.
- d. A fase de ideação utiliza critérios norteadores para a criação de personagens fictícios, que podem assumir papéis de usuários com características iguais.
- e. A fase de validação do projeto ocorre na imersão, tendo em vista que, nas demais etapas, ocorrem testes e entrevistas com os clientes e usuários.

#### **REFERÊNCIAS**

AMBROSE, G.; HARRIS, P. **Design thinking**. Porto Alegre: Bookman, 2011.

DE SÁ, D. *et al.* **Desenvolvendo novos produtos**: conceitos, etapas e criação. Curitiba: InterSaber, 2017.

MASSARI, V. L. **Agile scrum master no gerenciamento avançado de projetos**: base para certificação EXIN Agile Scrum Master. Rio de Janeiro: Brasport.2016.

SABBAG, P. Y. **Inovação, estratégia, empreendedorismo e crise**. Rio de Janeiro: Alta Books, 2018.

STICKDORN, M. *et al.* **Isto é design thinking de serviços**. Porto Alegre: Bookman, 2014.

VIANNA, M. *et al.* **Design thinking**: inovação em negócios. Rio de Janeiro: MJV Press, 2012.

## **FOCO NO MERCADO DE TRABALHO**

**ABORDAGENS INOVADORAS E ÁGEIS PARA A GESTÃO DE SOFTWARE**



Fonte: Shutterstock.

**Deseja ouvir este material?**

### **SEM MEDO DE ERRAR**

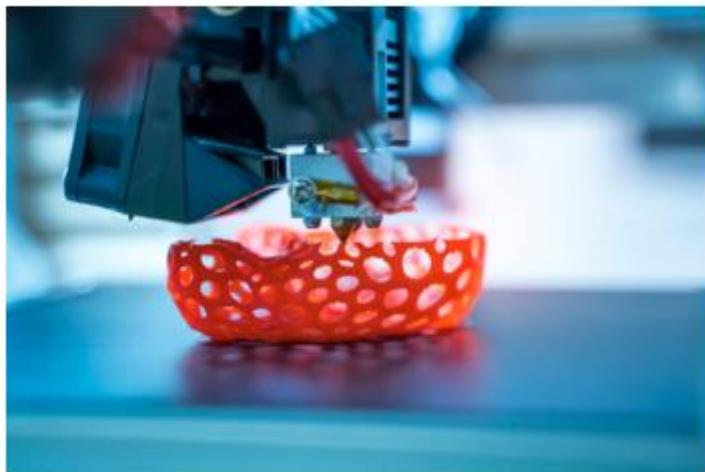
O festival de que fará parte objetiva trazer inovações para a apresentação de artistas e atrações diversas. O uso do *design thinking* será aplicado com o objetivo de agregar valores por meio da inovação, fazendo com que o evento se destaque dos demais por possuir elementos que enfatizam a tecnologia utilizada nesses outros ambientes, como de música, artes, teatro, entre outros.

Você será responsável por proporcionar tudo isso, então, o primeiro passo deverá ser a realização da análise do contexto envolvido na situação, uma vez que está relacionado não só ao uso da tecnologia, mas também à necessidade de agregar valor à experiência do usuário. Para isso, você poderá aplicar seus conhecimentos sobre as etapas do *design thinking* e, ao chegar na etapa de prototipação, usar o *Storyboard* para apresentar aos financiadores e demais envolvidos suas principais ideias.

Figura 2.18 | Tipos de eventos

EVENTO 1	EVENTO 2
 <p>O primeiro evento proporcionará aos usuários acesso à piscina do local, onde serão projetadas imagens holográficas e um show de luzes programadas conforme o som que estiver tocando durante o evento.</p>	 <p>O primeiro evento proporcionará aos usuários acesso à piscina do local, onde serão projetadas imagens holográficas e um show de luzes programadas conforme o som que estiver tocando durante o evento.</p>

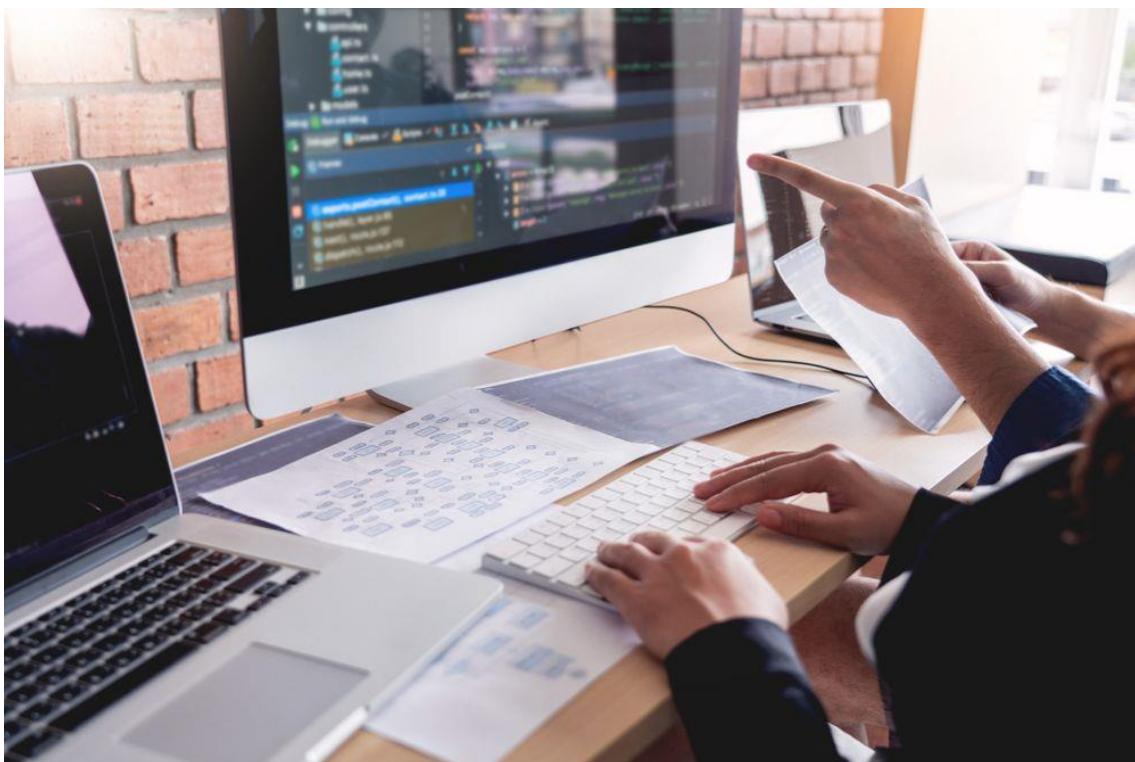
### EVENTO 3



O primeiro evento proporcionará aos usuários acesso à piscina do local, onde serão projetadas imagens holográficas e um show de luzes programadas conforme o som que estiver tocando durante o evento.

## **NÃO PODE FALTAR**

### **MAPEAMENTO DE RISCOS**



Fonte: Shutterstock.

**Deseja ouvir este material?**

#### ***CONVITE AO ESTUDO***

Caro aluno, está intrínseco que dependemos de vários sistemas em nosso dia a dia, sendo que, alguns deles, usamos sem perceber. No entanto, é importante lembrarmos que, para o seu desenvolvimento e funcionamento, foram necessários diversos profissionais que estudaram as necessidades de um sistema para estar intuitivo e integrado de forma incorporada na rotina de seus usuários. Em contrapartida, quando temos de utilizar um sistema cujo manuseio ou mesmo suas funcionalidades possuem falhas, rapidamente começamos a gerar resistência a sua utilização, justificando falta de qualidade e, em alguns casos, levantando suspeitas de sua confiabilidade e segurança.

Dentro desse cenário, vamos nos colocar do lado dos desenvolvedores de sistemas e começar a nos questionar: como evitar falhas no processo de desenvolvimento de software? O que pode dar errado? Como identificar os empecilhos na jornada do desenvolvimento, desde a ideia até a conclusão do sistema com sucesso? Lembre-se de que o conceito de sucesso no desenvolvimento de um projeto está em alcançar as

expectativas do cliente, cumprindo o tempo, o custo, a qualidade e o escopo proposto (COSTA; PEREIRA, 2019).

Sendo assim, para alcançar esse almejado sucesso, a primeira seção destaca a relevância no domínio de técnicas de gestões de projeto, incluindo, nesse processo, a compreensão de modelos diversos de projetos de software, gestão do risco e da qualidade, necessitando de um estudo aprofundado de como reconhecer a natureza do risco, contextualizando as possibilidades por meio de técnicas já reconhecidas e disseminadas a partir dos guias de boas práticas, como é o caso da matriz de risco.

Para o estudo do risco e, por meio dele, mitigar essas ameaças, o profissional da área de TI, para essa etapa do projeto, também terá a necessidade de contar com a experiência da equipe, que tem a missão de resolver problemas complexos por meio de análises diversas, quantitativas e qualitativas, utilizando-se de flexibilidade e adaptações buscando métricas para a qualidade do projeto de desenvolvimento de software, bem como criatividade e inovação para a definição de pronto, em que a premissa de atender à necessidade do projeto, do cliente ou do mercado é real, sem deixar de se preocupar com as limitações tecnológicas, mantendo uma relação direta com a qualidade, buscando planejar e organizar as atividades não apenas durante o desenvolvimento, mas também das manutenções futuras, assunto este tratado na segunda seção.

Isso levará à necessidade de se gerir a qualidade utilizando metodologias adequadas, buscando, por meio de métricas, a qualidade do software. Assim, na terceira seção, trataremos da utilização das documentações necessárias para que o projeto não corra riscos desnecessários devido à falta de informação e de comunicação, sendo relevantes, também, a empatia e a ética no trabalho entre os *stakeholders* do projeto. A tarefa da documentação nem sempre é complexa, mas é decisivo saber quando e por que gerar determinados documentos, a fim de que eles possam ser elementos que ajudem a agregar qualidade ao projeto e não apenas ser um simples diário de anotações ou um “manual técnico” para os desenvolvedores e outro para os usuários.

Devido a todos esses motivos explanados é que iniciaremos a unidade tratando do mapeamento de riscos; em seguida, apresentaremos a gestão da qualidade e, por fim, discorreremos sobre a documentação em projetos, objetivando que você, profissional do setor de TI, entenda as áreas do conhecimento referentes a riscos e qualidade e à importância da documentação em todo o processo do projeto.

Por meio desse aprofundamento, você terá a oportunidade de adquirir competências importantes para ingressar no mercado de trabalho. Vamos correr o risco de alcançar qualidade?

#### ***PRATICAR PARA APRENDER***

Os imprevistos são uma constante realidade do homem, e não faz parte apenas da realidade do séc. XXI e sim de toda história da humanidade. Isso ocorre pois fazemos planos e projetos, criamos e geramos expectativas. Todos esses processos estão inerentes em nossa sobrevivência, na realização de desejos e sonhos; somos capazes, desde pequenos, de analisar e resolver problemas, basta lembrarmos do que fazíamos

quando crianças para conseguir um passeio ou ganhar um presente, bem como agora, já adultos, para realizar um desejo ou sonho.

Pode ser que não tenhamos os conceitos claros ou estejamos utilizando os termos adequados do mundo profissional, mas riscos estão em todas as partes. Para você que está estudando a área de desenvolvimento de sistemas, o risco mais eminente, entre os inúmeros existentes, é a falta de domínio na semântica de uma nova linguagem que você esteja aprendendo, bem como as novas tecnologias, já que elas se renovam com uma velocidade impressionante e, por muitas vezes, apostamos nessa ou naquela nova tendência e dependemos da aceitação do mercado para realmente saber se será disseminada. Para entender esse último risco, faça uma breve pesquisa sobre a invenção da tela *touch screen* (COSTA; PEREIRA, 2019).

O que se pode observar é que as naturezas dos riscos são inúmeras e devem ser categorizadas, a fim de que exista a possibilidade de se criar planos de contingência. Em geral, os riscos afetam as bases de um projeto, o tempo, o custo, a qualidade e, muitas vezes, modifica o escopo do projeto e/ou produto (PMI, 2017), o que torna esse assunto um dos mais complexos a ser tratado, estudado e viabilizado. Para esse contexto, são fundamentais a criatividade e a inovação no processo de análise, bem como a busca constante por métodos e ferramentas adequados para a criação da *checklist*, a fim de se identificar os riscos, definir a natureza, classificar sua relevância por meio de análise da probabilidade e o impacto de sua ocorrência (SOMMERVILLE, 2018).

Para que essa área do conhecimento seja um sucesso, é necessário integração dentro das técnicas de gestão de projetos reconhecidos no mercado e compreensão da gestão do risco, que, por consequência, vai colaborar com a qualidade do projeto de desenvolvimento do software, o que faz o relacionamento interpessoal ser muito bem trabalhado, buscando-se a empatia, a gestão de conflitos (um dos itens a ser desenvolvido pela equipe) e a ética profissional, sendo flexível e adaptativa a todos os níveis, desde os relacionados ao desenvolvimento do projeto, das atividades, do cronograma até mesmo no que se refere ao planejamento e à organização da equipe (CARVALHO, 2018).

Imaginemos a seguinte situação: você é integrante de uma equipe de desenvolvimento de sistemas para a área da saúde e pretende desenvolver uma ferramenta CASE utilizando tecnologia de identificação de gestos e/ou voz com recursos de realidade virtual. Com esse recurso, os médicos, durante um procedimento cirúrgico, poderão analisar tomografias, ultrassonografias e raios-x digitais, minimizando o tempo do procedimento cirúrgico, uma vez que, por meio desse recurso de gestos e/ou voz, o profissional da medicina não terá a necessidade de remover as luvas cirúrgicas para manipular a tela do computador ou as chapas físicas, pouparando uma nova higienização das mãos para voltar ao procedimento, além do paciente ficar menos tempo anestesiado e o centro cirúrgico disponível para mais procedimentos.

Observe, nessa descrição, o escopo do projeto e o escopo do produto a ser desenvolvido; verifique o quanto de riscos estamos propensos no que diz respeito ao desenvolvimento do projeto. É relevante criar uma *checklist* de riscos, classificar sua natureza e definir a ação a ser tomada para cada um dos riscos identificados. Além disso, é importante incluir não apenas os riscos referentes ao desenvolvimento do

sistema e à utilização das tecnologias, mas também ao perfil do futuro usuário e do *sponsor*.

Nesse cenário, perceba que a importância da união de todos os *stakeholders* é relevante para se identificar todos os riscos, dos mais prováveis e de maior impacto até os de menor relevância e impacto, buscando construir todo processo de monitoramento no decorrer do desenvolvimento desse produto que tem um escopo de projeto de significância para os envolvidos.

Portanto, gere o RMMM desse projeto; avalie as tecnologias, as pessoas que estarão envolvidas com você nesse trabalho, categorizando cada um dos riscos identificados; gere a matriz de risco; crie a ordenação de prioridade; aponte meios de mitigar a probabilidade de ocorrência ou a redução do impacto de cada risco; e defina quem será o responsável por acompanhar e agir em cada risco. Todos esses itens trarão um impacto positivo no que diz respeito à minimização dos prejuízos que esses riscos, caso não levantados, podem acarretar, podendo-se evitar o impacto nos custos do projeto. Outro ponto positivo a ser observado é a alocação de recursos, que traz um benefício direto ao cronograma, cumprindo-se o prazo ou mesmo reduzindo. Esse tipo de resultado traz, a toda equipe, uma satisfação de dever cumprido.

Assim sendo, gere o documento que conduzirá a gestão de risco e o monitoramento, que ajudará no processo do planejamento e organização, a fim de se conseguir analisar e resolver os problemas que venham a acontecer de forma prevista ou mesmo imprevista. Para isso, você e sua equipe podem se utilizar de muita inovação e criatividade, buscando atender aos requisitos solicitados, flexibilizando e se adaptando a cada um deles.

Correr riscos faz parte da vida, o que precisamos é aprender a prevê-los, criar meios de mitigar, mas, principalmente, potencializar a existência dos riscos positivos. Para tanto, estudar conteúdos relacionados ao tema é um excelente caminho para conseguirmos esse objetivo, tanto na vida profissional como pessoal.

## **CONCEITO-CHAVE**

### **MAPEAMENTO DE RISCOS**

Quantos riscos corremos na nossa rotina diária? Está aí uma pergunta que não fazemos todos os dias quando acordamos. Afinal, ao falarmos de risco, pensamos, de forma geral, no que pode dar errado. No entanto, é fundamental sabermos que existem autores que defendem a ideia de que o risco que corremos pode ser positivo. É fato que a incerteza é parte corriqueira de nosso dia, ela acompanha a humanidade desde o período da caça até hoje, em que buscamos níveis de satisfação nas necessidades básicas, incluindo a carreira profissional. Na profissão de desenvolvimento de sistemas, temos infinitas incertezas, pois estamos, na maior parte do tempo, aprendendo a entender a necessidade do nosso cliente, que, por muitas vezes, afetará não apenas a dinâmica interna de uma organização, mas, provavelmente, todos os seus clientes diretos e indiretos. O que podemos constatar é que fazemos escolhas o tempo todo, algumas com um grau de risco ou incerteza diferente, mas que podem, também, beneficiar a nós mesmos e/ou o próximo.

Considerando os termos risco e incerteza, existe uma diferença entre eles, onde risco se refere a uma situação incerta ou uma condição, que, caso ocorra, poderá ter resultados positivos ou negativos sobre um dos objetivos, podendo ser o escopo, o cronograma, o custo ou a qualidade. Já a incerteza é uma situação que não se consegue especificar as chances de ocorrer, ou seja, falta conhecimento consciente sobre os resultados que ela virá a provocar (COSTA; PEREIRA, 2019).

Ao falarmos de desenvolvimento de um software, buscamos a compreensão de todas as funcionalidades e de todos os requisitos possíveis de serem obtidos em seu início, por isso, há tantos estudos, nas últimas décadas, voltados às técnicas de gestão de projetos para o conhecimento de todos os processos do plano de desenvolvimento do sistema.

Além disso, faz-se importante compreender mais de um modelo de desenvolvimento de projeto de software e a existência de erros e suas origens, bem como analisar os riscos de forma a geri-los e evitar falhas ou minimizar as consequências, uma vez que esses riscos negativos podem afetar, de forma direta, a qualidade do produto ou serviço relacionado aos projetos de software. No caso dos riscos positivos, deve-se acompanhá-los, a fim de potencializar os benefícios trazidos por eles (WAZLAWICK, 2013).

Conseguir aprender meios de garantir a qualidade do software a ser desenvolvido e mitigar os riscos que podem prejudicar o projeto por meio de ações preventivas ou de uma rápida resposta ao problema previsto ou imprevisto são formas de desenvolver flexibilidade e adaptação, buscando-se um planejamento prévio e a organização dos fatores que são prejudiciais e positivos, analisando e resolvendo problemas dentro dos processos do desenvolvimento de sistemas e do projeto, em que a necessidade do trabalho em equipe, da existência de empatia pelos *stakeholders* envolvidos, da ética no trabalho, do relacionamento interpessoal e da gestão de conflitos são essenciais e devem permanecer.

#### REFLITA

Até onde vale a pena correr riscos? Considerando o desenvolvimento de um sistema de e-commerce, em que são desenvolvidas duas aplicações distintas, uma para uso por meio de navegadores e a outra por meio de APP, em geral, temos características de desenvolvimento com acréscimo de linguagens e API's distintas para cada sistema, levando-se em conta que o tempo de entrega desses sistemas está no seu limite e o surgimento de uma nova API para a busca dos produtos foi inserida no módulo do APP, que interfere na necessidade de mudança da estrutura de Banco de dados.

Frente a isso, como fazer essa integração?

Nesse momento, a minimização do risco ocorre por meio dos métodos de testes, evitando-se, assim, uma cascata de erros. Portanto, qual o caminho: o teste de regressão ou o teste fumaça? Ambos têm o objetivo de analisar a integração de novos recursos ou de alterações. O teste de regressão visa a minimizar os efeitos colaterais que podem ocorrer no que já está funcionando, sendo indicado, principalmente, à integração de um novo componente. Já no teste fumaça, criado para projetos com prazo crítico, o sistema é recriado com todos os arquivos, bibliotecas e o novo componente que necessita ser acrescentado, além de tudo que é necessário para o seu funcionamento, seguindo as orientações da série de testes criada para a simulação de falhas e realizada diariamente, utilizando-se o sistema do início ao fim das funcionalidades criadas, buscando os erros que venham a bloquear o funcionamento do sistema e evitar os atrasos no cronograma.

Mas será que compensa abrir mão dos testes e correr riscos por não disponibilizar um desenvolvedor ou mesmo uma equipe para testar os sistemas diariamente? (PRESSMAN; MAXIM, 2015).

Quando se tratam de riscos, vale rever como eles são contextualizados em algumas referências, como o PMBOK (2017, p.723): “Um evento ou condição incerta que, se ocorrer, provocará um efeito positivo ou negativo em um ou mais objetivos do projeto”. Mais especificamente, o Risco Geral do Projeto, segundo o PMI (2017, p. 723) é:

O efeito da incerteza no projeto como um todo, decorrente de todas as fontes de incerteza, incluindo riscos individuais, representando a exposição das partes interessadas às implementações de variações no resultado do projeto, tanto positivas quanto negativas.

#### **ISO 31000**

A ISO 31000, que tem como escopo o gerenciamento de riscos de qualquer natureza, entende que o risco é o “efeito da incerteza nos objetivos”, em que: “um efeito é um desvio em relação ao esperado. Pode ser positivo, negativo ou ambos e pode abordar, criar ou resultar em oportunidades e ameaças” (ABNT, 2018.,) Apesar de os objetivos de cada projeto serem os mais variados, possuindo diferentes aspectos e categorias no desenvolvimento dos sistemas, que leva a diferentes níveis de riscos, para que a organização possa dirigir e controlar o risco, são necessárias atividades coordenadas, de forma que, ao encontrar e determinar as fontes potenciais que darão origem aos riscos de forma individual ou conjunta, consiga controlar os resultados que venham a afetar os objetivos, sempre analisando a probabilidade da ocorrência por meios de medições, demonstrando a manutenção dos estragos do risco.

A NBR ISO 31000 fornece diretrizes para o gerenciamento de risco baseadas em três pilares:

- Liderança e comprometimento.
- Criação e proteção de valor.
- Processos de gestão de riscos.

Ao consultar o ITIL 4, temos princípios semelhantes que os encontrados na NBR ISO 31000.

#### **COBIT 5**

Enquanto os guias anteriores falam dos riscos de forma geral e são voltados para o desenvolvimento de projetos, produtos e serviços, o COBIT 5 visualiza o risco para a organização a partir da segurança da informação, já que as informações, atualmente, são gestadas por meio de sistemas, ajudando as organizações a criarem valores por meio da TI e o equilíbrio entre a minimização dos riscos, a busca dos benefícios e a utilização dos recursos (COBIT 5, 2019), sendo fortalecido pela Lei Sarbanes-Oxley, que busca garantir, por meio da criação de comitês que têm a função de supervisionar as atividades e operações, de forma a mitigar os riscos aos negócios, identificando e/ou evitando

fraudes, criar mecanismos de auditoria e garantir, com tudo isso, a transparência na gestão da empresa (CARVALHO, 2018).

### **PMBOK**

Para o processo de gerenciamento de riscos, o PMBOK sugere sete processos de gerenciamento de risco: planejar o gerenciamento, identificar, realizar a análise qualitativa e quantitativa dos riscos para se planejar e implementar as respostas e monitorar a implementação dos planos de respostas, bem como acompanhar os riscos identificados, podendo, ainda, identificar e analisar novos riscos, além de avaliar se o processo de risco ao longo do processo está sendo eficaz (PMI, 2017).

### **RISCOS QUE IMPACTAM NO DESENVOLVIMENTO DE SISTEMAS**

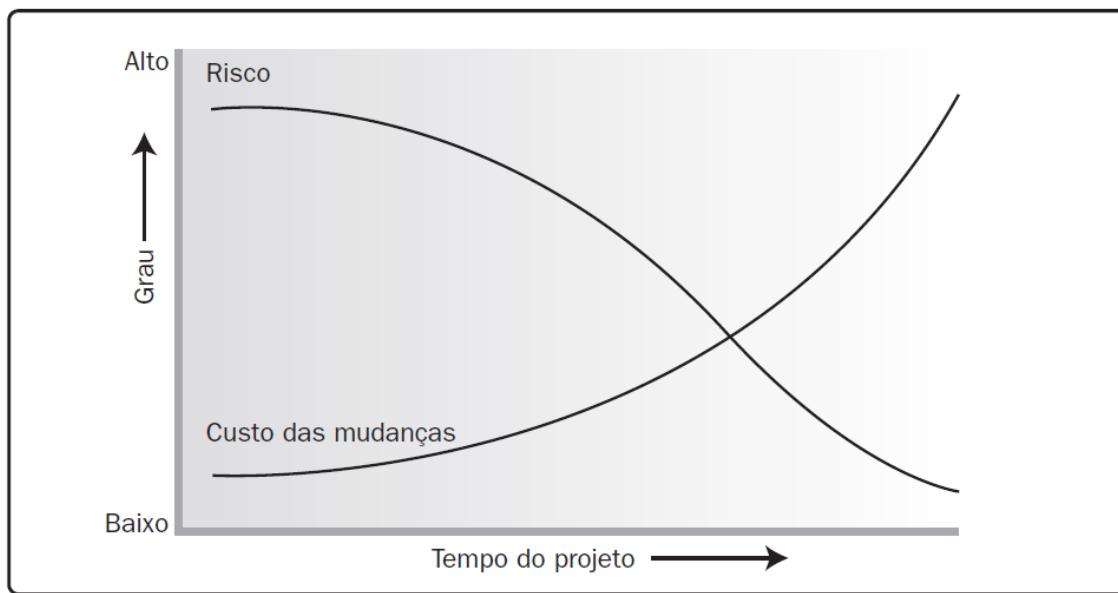
Na prática, é importante definir todos os riscos que podem impactar, de alguma forma, o desenvolvimento de sistemas. Entre os inúmeros tipos de sistemas existentes, todos estão ameaçados pelos mesmos tipos de erros, podendo influenciar o **projeto, o produto e/ou o negócio**. Para que se definir os riscos do projeto de desenvolvimento de sistemas, é possível utilizar as técnicas de *checklists*, reuniões e *brainstormings*, além de análise de cenário e lições aprendidas em projetos anteriores. Essas técnicas são interessantes na fase do planejamento, porém podem ser utilizadas sempre que o gerente de projeto e/ou a equipe entender conveniente (SOMMERVILLE, 2018).

### **GRAU DE INCERTEZA**

Segundo Carvalho (2018), o grau de incerteza está vinculado à falta de informação, pois, no início de um projeto, é comum se ter entre 40% e 80 % das informações necessárias para tomar decisões. E conforme o projeto caminha, as incertezas reduzem, tendo em vista que o domínio de conhecimento das várias etapas aumenta. Essa análise vai de encontro ao gráfico que o PMBOK sugere, segundo os impactos dos riscos, variáveis indefinidas e o tempo de desenvolvimento do projeto, o que é inversamente proporcional aos custos (PMI, 2017).

Considerando o fato de que atua na área de tecnologias, as inovações são constantes e as incertezas são eminentes, portanto, para se evitar as surpresas desagradáveis, quanto mais complexo for o sistema a ser desenvolvido e maior a duração do desenvolvimento, maiores poderão ser os riscos; sendo assim, apenas as reservas de contingência poderão preservar o projeto dos riscos desconhecidos (COSTA; PEREIRA, 2019).

Figura 3.1 | Gráfico que o PMBOK sugere como o impacto de variáveis ao longo do tempo



Fonte: PMI (2017, p. 549).

### PROCESSO DE IDENTIFICAÇÃO DE RISCOS

Nesse processo de identificação, é importante considerar os diferentes tipos de riscos que poderão surgir, e ele pode ocorrer por meio do *brainstorm*, afinal, os riscos são do time de desenvolvimento. Podemos pensar em 6 tipos de riscos para a *checklist* de riscos: de estimativa: a gerência determina os recursos necessários; organizacional: dependerá do ambiente da organização em que o sistema está sendo desenvolvido; pessoas: do time de desenvolvimento; requisitos: ocorrem a partir das mudanças solicitadas pelo cliente e pelo processo de gestão das mudanças e ferramenta do software e dos softwares de apoio para a criação do sistema (SOMMERVILLE, 2018).

Figura 3.2 | Exemplos de diferentes tipos de risco

Tipos de risco	Possíveis riscos
Estimativa	1. O tempo necessário para desenvolver o software foi subestimado. 2. A taxa de correção de defeitos foi subestimada. 3. O tamanho do software foi subestimado
Organizacional	4. A organização foi reestruturada, e uma gerência diferente ficou res...
Pessoal	5. Problemas financeiros da organização obrigam a reduções no orçamento. 6. É impossível recrutar pessoas com as habilidades necessárias. 7. Um membro importante da equipe está doente e indisponível em...
Requisitos	8. O treinamento necessário para a equipe não está disponível. 9. Mudanças propostas nos requisitos exigem uma grande dose de...

Tipos de risco	Possíveis riscos
Tecnologia	10. Os clientes não entendem o impacto das mudanças nos requisitos 11. O banco de dados utilizado no sistema não consegue processar tanto quanto previsto.
Ferramentas	12. Defeitos nos componentes de software reusáveis têm de ser consertados 13. O código gerado pelas ferramentas de geração de código é ineficiente
	14. As ferramentas de software não conseguem trabalhar juntas de maneira eficiente

Fonte: Sommerville (2018, p. 613).

### ASPECTO QUALITATIVO E QUANTITATIVO

Após a identificação de todos os riscos prováveis, cabe agora definir o aspecto qualitativo e quantitativo com a identificação de seus possíveis impactos monetários de tempo, qualidade e produto, de forma a gerar a lista na ordem de priorização dos riscos baseada na combinação dos impactos e da probabilidade de risco.

Para essa análise, é possível utilizar a matriz de risco, e para a criação da matriz, é importante definir valores associados ao impacto e ao efeito dos riscos mediante as questões de custo, escopo, prazo e qualidade do projeto. Portanto, uma das melhores formas de se definir cada um dos riscos identificados é por meio de informações históricas de qualidade, como falado por meio do documento de lições aprendidas, da *expertise* dos profissionais responsável pelo projeto e da organização.

Quanto à análise da probabilidade do risco, ela pode ser classificada como insignificante, baixa, moderada, alta ou muito alta. Os efeitos que cada um dos riscos pode causar no projeto de desenvolvimento de um software: insignificante/muito baixo (não afeta o projeto), tolerável/baixo (abaixo da contingência), médio (nos limites da contingência), grave (significativo, pois ultrapassa a contingência) e gravíssimo (ameaça a existência do projeto) (COSTA; PEREIRA, 2019).

Quadro 3.1 | Matriz de probabilidade de impacto utilizando pontuação para análise de ameaças

Probabilidade	Muito alto 0,90	0,045	0,09	0,18	0,36	0,72	
	Alto 0,70	0,035	0,07	0,14	0,28	0,56	
	Média 0,50	0,025	0,05	0,10	0,20	0,40	
	Baixa 0,30	0,015	0,03	0,06	0,12	0,24	
	Muito Baixa 0,10	0,005	0,01	0,02	0,04	0,08	
		Muito baixo 0,05	Baixo 0,10	Média 0,20	Alto 0,40	Muito alto 0,80	
		Impacto Negativo					

Fonte: adaptada de PMI (2017).

Essa mesma matriz pode ser espelhada para fazer análise das oportunidades. Após essa distribuição é possível analisar a priorização dos riscos apenas observando a matriz, bem como notar que os riscos considerados altos estão acima de 0,20, os moderados entre 0,08 e 0,20 e os baixos menores que 0,08.

## EXEMPLIFICANDO

Pensando na tabela com os 14 possíveis riscos para o desenvolvimento de um sistema, temos:

Quadro 3.2 | Escala de risco

<b>Probabilidade</b>	Muito alto 0,90		2	3	4	6
	Alto 0,70		14	7	1, 9	
	Média 0,50		10, 13	8	12	11
	Baixa 0,30					5
	Muito Baixa 0,10					
<b>Impacto Negativo</b>		Muito baixo 0,05	Baixo 0,10	Média 0,20	Alto 0,40	Muito alto 0,80

Dessa forma, os riscos 6, 11, 4, 1, 9 e 5 são os seis riscos, nessa exata sequência, os quais o gerente de projeto deve acompanhar bem de perto.

De forma geral, escolhe-se os 10 primeiros a serem acompanhados, no entanto, deve-se considerar cada situação. Se tiver 20 riscos listados sendo considerados de alta probabilidade e impacto, será necessário acompanhar os 20. Note que esse processo visa a pontuar questões que, em geral, não se consegue necessariamente mensurar, por isso, a utilização dos níveis e a pontuação respectiva.

Após esse processo de qualificar os riscos, apenas com os que estão na faixa de alto risco, devem ser quantificados, assim concentra-se os esforços no que é de alta relevância, minimizando o tempo gasto neste processo. Não significa que em um outro momento do projeto de software, com o surgimento de novos riscos e até mesmo a extinção de outros, seja necessário reavaliar todos o processo novamente. Até esse momento, se tem categorização; priorização; definição das atenções especiais; a lista de quem necessita de respostas em curto prazo e a os pontos de alerta para as possibilidades de tendências de novos riscos (CARVALHO, 2018).

A análise quantitativa se dá por meio de métodos estatísticos que têm como objetivo complementar o processo anterior. Verificando em valores, as técnicas mais utilizadas são: *expected monetary value* (EVM) ou análise do valor monetário esperado, simulação de Monte Carlo, árvore de decisão, entre outras. Para essas análises, é necessário um profissional experiente, além disso, é possível utilizar alguns softwares como na simulação de Monte Carlo os sistemas Crystal Ball, @Risk e Simul8, que são utilizados de apoio para o processo de tomada de decisão e definição das metas da organização. Por meio deles, os gestores conseguem responder qual a chance de um determinado resultado ocorrer antes de uma decisão definitiva, e é indicado utilizá-los em qualquer momento do planejamento estratégico, oferecendo suporte à análise das decisões para médio e longo prazos.

## **ANÁLISE DA ÁRVORE DE DECISÃO**

Para a análise da árvore de decisão, são indicados os sistemas *Tree Plan* ou *Precision Tree*, que constroem um mapa de possíveis resultados a partir de uma série de escolhas relacionadas, o que permite à empresa ou ao gerente de projeto definir o que fazer mediante análise de custo, benefícios e probabilidade, podendo ser utilizados para conduzir discussões informais ou desenhar um algoritmo que prevê, matematicamente, a melhor escolha.

É possível encontrar inúmeras empresas no mercado ofertando suporte a esses processos de análise quantitativa de risco, e a experiência do profissional é significativa nesse processo, assim como da equipe de desenvolvimento (COSTA; PEREIRA, 2019).

## **PLANEJAMENTO DE RISCO**

A partir dessas etapas concluídas, passa-se para o planejamento de risco, em que é preciso definir que atitude tomar no caso do risco virar realidade. Há as opções de: **eliminar** – estratégia conservadora, evita-se o risco e altera o projeto, podendo aumentar o prazo ou o recurso, reduzir o escopo ou até cancelar o projeto em última situação; **transferir** – quando se tem a opção da contratação de outra empresa para assumir a parte do projeto da qual a equipe não sabe tratar o problema; **mitigar** – fazer o possível para reduzir as chances e/ou consequências do evento que venha a provocar o risco, para isso, é necessário proatividade, realizar testes e simulações com mais constância e criticidade e utilizar processos redundantes ou menos complexos; e **aceitar** – o plano de contingência deverá estar bem elaborado ou, de forma mais passiva, a equipe deverá tomar as decisões da melhor forma possível (CARVALHO, 2018).

## **ASSIMILE**

Planejamento é um conceito que abrange todas as áreas e em todos os momentos e para todos aqueles que desejam lançar objetivos e metas. Ou seja, planejar é uma atitude corriqueira do ser humano que sonha, deseja ou mesmo quando alguém lhe “obriga” a executar uma atividade. E para cada ação que se necessita fazer ou opta por não fazer, essa decisão influenciará consequências futuras, podendo ser positivas, negativas ou neutras. Então, quem atua na área de TI entende claramente o significado da palavra “SE” acompanhada do “SE NÃO”, e para planejar meios de eliminar, mitigar, transferir e simplesmente aceitar os riscos, é preciso usar a palavra “SE”. Portanto, em um projeto de desenvolvimento de software, os riscos de maior importância devem, sem dúvida, ter um plano de forma sumária e com as ações precisas para reduzir sua probabilidade ou impacto no seu produto de software. Os riscos médios devem estar listados e ter seu plano de ação escrito, porém à vista, para que, caso venha a aumentar sua probabilidade ou o grau de impacto, algo seja feito o quanto antes. Já os riscos de baixa importância devem ser monitorados, pois se vierem a ter alterações para o nível acima, um plano de ação deverá ser imediatamente preparado.

Em suma, o risco do qual se sabe a eminência deve estar sempre em sua supervisão, mas, sem dúvida alguma, tanto equipe quanto gerente do desenvolvimento devem estar alertas às incertezas (WAZLAWICK, 2013).

Para um efetivo planejamento de risco, é interessante a criação de documento apropriado para cada empresa e equipe. Portanto, esse documento deve ser elaborado em grupo, como mencionado no início, para que os participantes possam visualizar um exemplo prático.

Veja, num projeto fictício, onde será desenvolvido um sistema que visa à solicitação do pedido que será retirado pelo aluno na cantina, no horário do intervalo, a fim de organizar e agilizar a produção dos pedidos a serem preparados, evitando-se, também, a produção em excesso de um determinado produto e a falta de outro para o bom atendimento da cantina. Para esse sistema, será necessário que o cliente tenha acesso à internet, uma identificação mínima para o acesso e o controle dos pedidos, a criação de promoções e as modificações do cardápio com agilidade, pode meio de aparelho celular, computadores ou totem fixado ao lado da cantina.

A partir da análise feita pela equipe de desenvolvimento (alunos da área de TI da própria instituição), pelo cliente e pelo gerente do projeto (professor), foram identificados os riscos desse projeto com a sigla “e” – estimativa, “o” – organizacional; “p” – pessoas, “r” – requisitos, “t” – tecnologia e “f” – ferramentas.

Quadro 3.3 | Lista de riscos com identificação e análise de P - probabilidade, I - impacto e E - exposição/importância; todos classificados de forma simplificada em relação à Matriz de risco como B (baixo), M (médio) e A (alto)

<b>Id</b>	<b>Causa</b>	<b>Risco</b>	<b>Efeito</b>	<b>P</b>	<b>I</b>	<b>E</b>
p1	Falta de domínio das ferramentas e tecnologia, necessárias para o desenvolvimento.	Recruitar pessoas com o conhecimento necessário.	Aumentar o tempo para o desenvolvimento do sistema.	A	A	A
p2	Não se sabe se será possível pagar treinamento para, ao menos, um integrante.	Necessidade de treinamento.	Atraso no cronograma e custos com treinamento.	A	M	A
e1	Requisitos muito instáveis.	Mudança nos requisitos.	Necessitando de retrabalho, atraso no cronograma.	A	M	A
p3	Doença, trancar o curso.	Membro da equipe ficar indisponível.	Perda de desenvolvedores, necessidade de substituição.	M	M	M
o1	Aumento do valor dos produtos e/ou redução do número de alunos.	Problemas financeiros da organização.	Redução do orçamento.	B	A	M
f1	Geração de código ineficiente.	Código gerado pela ferramenta de geração de código.	Dificulta a correção do código na manutenção.	B	M	B
t1	Utilização de banco de dados gratuito, com limitações de armazenamento	Banco de dados não atende à demanda.	Sistema necessitará de atualização; aumento de gastos.	B	B	B

Fonte: adaptado de Wazlawick (2013) e Sommerville (2018).

A partir da identificação dos riscos em uma primeira reunião, gera-se um plano de redução de probabilidade e impacto focando as causas do risco. Já o plano de redução de impacto visa a reduzir o impacto que os riscos podem causar no desenvolvimento do sistema, diminuindo os seus efeitos. Para isso, é necessário o plano de mitigação,

contingência e redução de impacto com atividades definidas, prazos e responsáveis para a imediata intervenção no primeiro sinal de o risco se tornar realidade, incluindo nos planos do projeto, sendo contabilizadas essas ações no custo e no tempo (WAZLAWICK, 2013).

Quadro 3.4 | Plano de redução da probabilidade baseado na causa

<b>Id</b>	<b>Causa</b>	<b>Risco</b>	<b>Plano de redução de probabilidade</b>
p1	Falta de domínio das ferramentas e tecnologia, necessárias para o desenvolvimento.	Recrutar pessoas com o conhecimento necessário.	Realizar pesquisas e cursos necessários para aprofundar o conhecimento.
p2	Não se sabe se será possível pagar treinamento para, ao menos, um integrante.	Necessidade de treinamento.	Cursos gratuitos, busca por colegas e professores para auxiliar.
e1	Requisitos muito instáveis.	Mudança nos requisitos.	Planejar protótipos, verificar a existência de produtos semelhantes, realizar reuniões de elucidação de requisitos.
p3	Doença; trancar o curso.	Membro da equipe ficar indisponível.	Reorganizar a equipe para as atividades a serem cumpridas e todos estarem a par do trabalho um do outro.
o1	Aumento do valor dos produtos e/ou redução do número de alunos.	Problemas financeiros da organização.	Relatar o quanto o projeto colaborará para a recuperação da cantina.
f1	Geração de código ineficiente.	Código gerado pela ferramenta de geração de código.	Verificar a existência de ferramentas amigáveis para o processo de manutenção futura.
t1	Utilização de banco de dados gratuito, com limitações de armazenamento.	Banco de dados não atender à demanda.	Busca por um banco de dados com bom custo-benefício.

Fonte: adaptado de Wazlawick (2013) e Sommerville (2018).

No Quadro 3.4, foi descrito o plano para a redução da probabilidade dos três últimos itens, no entanto, não existe essa necessidade, a não ser que o grau de probabilidade de ocorrência venha a aumentar.

Visualize o Quadro 3.5 para análise de minimização dos impactos que poderão ocorrer. Observe que o último item não foi elencado devido ao fato de o seu grau de impacto ser baixo.

Quadro 3.5 | Plano de redução de impacto baseado no efeito

<b>Id</b>	<b>Risco</b>	<b>Efeito</b>	<b>Plano redução de impacto</b>
p1	Recrutar pessoas com o conhecimento necessário.	Aumentar o tempo para o desenvolvimento do sistema.	Busca por auxílio de colegas e professores.
p2	Necessidade de treinamento.	Atraso no cronograma e custos com treinamento.	Buscar por bibliografias e treinamentos. Inserir esses recursos no custo.
e1	Mudança nos requisitos.	Necessidade de retrabalho, atraso no cronograma.	Implementar sistema de versionamento. Definir, o quanto antes, a arquitetura-base.
p3	Um membro da equipe ficar indisponível.	Perda de desenvolvedores, necessidade de substituição.	Sobreposição de trabalho entre os integrantes da equipe.
o1	Problemas financeiros da organização.	Redução do orçamento.	Demonstrar que cortes não trarão o custo-benefício adequado ao projeto.
f1	Código gerado pela ferramenta de geração de código.	Dificulta a correção do código na manutenção.	Verificar a existência de ferramentas que sejam amigáveis ao processo de manutenção.

Fonte: adaptado de Wazlawick (2013) e Sommerville (2018).

Todo esse processo de análise e geração da tabela com os riscos identificados, organizados por ordem de prioridade, a partir da análise quantitativa e qualitativa, definição dos planos de mitigação, monitoramento e controle também pode ser encontrado como RMMM (*risk, mitigation, monitoring and management*) gerenciamento e monitoramento na mitigação de riscos (PRESSMAN, 2016).

### **GERENCIAMENTO E MONITORAMENTO DOS RISCOS**

Como foi citado, o gerenciamento de risco é um processo interativo, ou seja, precisa ocorrer durante todas as etapas do projeto, contar com a constante participação de todos os *stakeholder* e, principalmente, ter sua comunicação efetiva.

Para haver o gerenciamento ou monitoramento dos riscos, é fundamental que o gerente do projeto monitore os indicadores dos fatores que venha a aumentar ou diminuir a probabilidade de o risco ocorrer, bem como o seu impacto, que acontece no grupo de processo de monitoramento em paralelo com o grupo de processo de desenvolvimento do projeto. Para isso, é importante monitorar cada um dos riscos que está no início da lista definida no planejamento. Em desenvolvimento de software observa-se, principalmente, no aumento do número de alterações dos requisitos, este fornecerá pistas referente à probabilidade da ocorrência de boa parte dos riscos analisados (WAZLAWICK, 2013; SOMMERVILLE, 2018).

Em suma, faz parte do processo de monitoramento e controle dos riscos a reavaliação constante. Pode-se dizer que dependendo da lista, fazer um auditória da validação de importância destes riscos e do planejamento proposto a eles, principalmente nos relacionados a custo, analisar todas as variações e tendências de cada uma das variáveis definidas pela equipe, medições continuas e o desempenho técnico dos riscos e possuir a prática de pequenas reuniões referente ao andamento em relação aos riscos, muito interessante este último processo quando se vincula ao desenvolvimento de software por meio das metodologias ágeis (CARVALHO, 2018).

Após análise de todo o conteúdo apresentado, estudado e desenvolvido nesta seção, cabe a reflexão de que os autores citados, apesar de elencarem em ordem distinta as sequências de análise dos riscos ou de criarem categorias um tanto quanto diferentes na análise da natureza do risco, ainda assim, todos têm o mesmo objetivo: buscar garantias de minimizar os riscos negativos ou, ao menos, neutralizar e potencializar os positivos, o que torna todo o nosso estudo de altíssima relevância e de prática diária.

### FAÇA VALER A PENA

#### Questão 1

Ao desenvolver um projeto de desenvolvimento de software, uma nova tecnologia é inserida no mercado de trabalho cuja principal função é trazer um sistema de segurança para os dados do cliente ao utilizar ferramentas e-commerce, e a empresa em que você está atuando opta por fazer alteração no projeto visando a agregar o que há de melhor e atual.

Considerando as informações apresentadas, analise as afirmativas a seguir:

- I. Deverá ser revista a análise de risco.
  - II. Toda a equipe deverá passar por treinamento.
  - III. Ocorrerá atraso na entrega do projeto.
  - IV. Haverá um aumento de custo do projeto.
- 
- a. I e II, apenas.
  - b. I, III e IV, apenas.
  - c. II, III e IV, apenas.
  - d. III e IV, apenas.
  - e. I, II, III e IV.

#### Questão 2

Um gerente de projeto de uma empresa multinacional tem, ao seu lado, um gerente de riscos que irá para um treinamento fora do país e lá ficará por uma semana. Frente a isso, passará as suas atribuições ao gerente de projeto, no entanto, são necessários alguns alinhamentos referentes ao assunto.

Tomando como referência a análise de risco, determine: V (Verdadeiro) ou F (Falso):

- ( ) A matriz probabilidade x impacto classifica os riscos envolvidos.
- ( ) As estratégias de respostas ao risco são meios de minimizar a ocorrência.
- ( ) O impacto do risco é a sua probabilidade de ocorrência.

Assinale a alternativa que apresenta a sequência CORRETA.

- a. V – F – V.
- b. V – V – F.
- c. F – F – V.
- d. V – F – F.
- e. F – V – F.

### Questão 3

No decorrer do desenvolvimento de um sistema para uma empresa de móveis planejados classificada como pequena empresa, existem alguns requisitos que, ao serem agregados ao sistema, traz agilidade e maior optimização do tempo de criação do projeto, refletindo na montagem dos móveis e minimizando o tempo de entrega do produto. Para esse sistema, foi solicitado o preenchimento dos dados básicos do projeto e a adição de fotos e das medições efetuadas no local. O sistema deverá funcionar por meio de um smartphone para ser possível o compartilhamento de informações com o projetista, com a produção e com o montador, atuando em todos os processos referentes aos móveis.

Baseado nos conceitos de risco e oportunidades, analise as asserções a seguir:

- I. O risco do projeto está relacionado ao fato de o cliente possuir recursos para o armazenamento dos arquivos do sistema.
- II. Oportunidade: a empresa de desenvolvimento de software fechou uma parceria com a empresa de *Cloud Storage*, oferecendo a biblioteca de controle do armazenamento dos arquivos.

A respeito dessas asserções, assinale a alternativa CORRETA.

- a. As asserções I e II são proposições verdadeiras e a II soluciona a I.
- b. As asserções I e II são proposições verdadeiras, mas a II não soluciona a I.
- c. A asserção I é uma proposição verdadeira e a II é falsa.
- d. A asserção I é uma proposição falsa e a II é verdadeira.
- e. As asserções I e II são proposições falsas.

### REFERÊNCIAS

ABNT — Associação Brasileira de Normas Técnicas. **NBR ISO 31000**. Rio de Janeiro: ABNT, 2018.

AXELOS. **ITIL foundation**: ITIL 4 edition. London: AXELOS, 2019.

CARVALHO, F. C. A de. **Gestão de projetos**. 2. ed. São Paulo: Pearson Education do Brasil, 2018.

COSTA, A. B. da. PEREIRA, F. da S. **Fundamentos de gestão de projetos: da teoria à prática – como gerenciar projetos de sucesso**; Curitiba. Intersabers, 2019.

ISACA. **COBIT 5 Implementation**. Rolling Meadows: ISACA, 2012.

NEWTON, R. **O gestor de projetos**. 2. ed. São Paulo: Pearson Prentice Hall, 2011.

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. **Software engineering: a practitioner's approach**. 8. ed. New York: McGraw-Hill Global Education Holdings LLC, 2016.

SOMMERVILLE, I. **Engenharia de software**. 10. ed. São Paulo: Pearson Prentice Hall, 2018.

WAZLAWICK, R. S. **Engenharia de software: conceitos e práticas**. Rio de Janeiro: Elsevier Editora Ltda., 2013.

## FOCO NO MERCADO DE TRABALHO



Fonte: Shutterstock.

**Deseja ouvir este material?**

### **SEM MEDO DE ERRAR**

Quando falamos de tecnologias que podem auxiliar os profissionais da área da saúde, é perceptível o grau de importância e interesse de inúmeras pessoas em relação ao projeto, afinal, nunca se espera precisar, mas também não podemos dizer que não nos beneficiará. Como exemplo, temos o projeto em que médicos, durante o procedimento cirúrgico, poderão manipular os resultados dos exames pré-operatórios sem a necessidade de remover as luvas cirúrgicas e, ainda, refazer a higienização das mãos

<b>Id</b>	<b>Risco</b>
1	Equipe sem o conhecimento necessário ou sem interesse na área.
2	Um membro da equipe sair da empresa.
3	Resistência dos usuários na utilização.
4	Tamanho do projeto estimado maior que o real.
5	Tecnologia não atende às expectativas.
6	Usuários poderão modificar os requisitos.
7	Utilização de rotinas prontas com necessidade de ajustes.

para o retorno ao procedimento, tudo com o objetivo de otimizar o procedimento, beneficiando a todos os envolvidos na cirurgia e os que estão no aguardo da liberação do espaço para o próximo procedimento.

Para esse sistema, o checklist de riscos são dos mais variados e, podemos dividi-los em tamanho do produto, impacto no negócio, características do cliente, definição do processo, ambientes de desenvolvimento e tecnologias para a construção e composição do pessoal; podendo ser riscos internos ou externos à organização (COSTA; PEREIRA, 2019).

Você pode iniciar com os questionamentos relacionados ao cliente: o cliente concorda em “perder” tempo com você? O cliente tem ideias sólidas a respeito dos requisitos? Quais são as expectativas realísticas? Mas também pode partir para questionamentos relacionados a pessoas: os membros da equipe têm as habilidades necessárias? A equipe possui o treinamento necessário? A equipe está comprometida com toda a duração do projeto? Há pessoas suficientes à disposição? Qual será a rotatividade entre os membros da equipe?

Existem, ainda, os riscos relacionados à organização: estão dispostos a investir em equipamentos adequados? Existe espaço no ambiente cirúrgico para a instalação dos equipamentos? Os profissionais da organização estão interessados em novos ambientes?

É preciso pensar, também, em outros tipos de riscos, como mudança de gestão, mudanças tecnológicas, concorrências de produtos, cronograma adequado a todo o processo de desenvolvimento, testes e implantação, comunicação eficiente e outros (a equipe define como categorizar melhor) (WAZLAWICK, 2013).

Portanto, após a criação da lista de riscos com suas devidas identificações, haverá o processo de análise qualitativa, em que será definida a probabilidade de ocorrer entre baixo, médio e alto nível e o impacto que terá sobre o projeto, caso ocorra. Assim, é possível gerar uma lista organizada de riscos de maior possibilidade e impacto para, na sequência, por meio da matriz de risco, definir os riscos que deverão ser analisados

Quadro 3.6 | Lista de riscos com identificação

Fonte: adaptado de Wazlawick (2013).

No Quadro 3.7 estão algumas sugestões de riscos. Existem inúmeros outros riscos a serem encontrados e alguns dependem da característica da organização e da equipe que irá desenvolver, no caso, inerentes ao seu domínio de conhecimento.

É possível utilizar a Matriz de risco a fim de se buscar uma classificação por meio de valores predefinidos. No exemplo anterior, perceba que todas devem estar na região de risco

Quadro 3.7 | Matriz de probabilidade de impacto utilizando pontuação para se analisar as ameaças

Probabilidade de Impacto	Muito alto 0,90			4	6	1
	Alto 0,70			5	2	7
	Média 0,50				3	
	Baixa 0,30					
	Muito Baixa 0,10					
		Muito baixo 0,05	Baixo 0,10	Média 0,20	Alto 0,40	Muito alto 0,80
<b>Impacto Negativo</b>						

Fonte: adaptado de PMI (2017).

Após inserir todos os riscos listados na matriz e selecionar todos que estão listados como médio até muito alto, bem como seus impactos, passamos para a lista de identificação do tipo de risco e descrição de sua causa e efeito.

Quadro 3.8 | Lista de riscos com identificação “e” – estimativa, “o” – organizacional, “p” – pessoas, “r” – requisitos, “t” – tecnologia, “f” – ferramentas, análise de P - probabilidade, I - impacto e E - exposição/importância. Todos classificados como B (baixo), M (médio) e A (alto)

Id	Causa	Risco	Efeito	P	I	E
p1	Falta de treinamento no uso da ferramenta.	Equipe sem o conhecimento necessário ou sem interesse na área.	Ajustar o tempo de desenvolvimento do sistema; criar meios de estimular o estudo.	A	A	A
p2	O usuário não ter total conhecimento de sua necessidade e das tecnologias.	Os usuários poderão modificar os requisitos.	Atraso no cronograma e custos com retrabalho.	A	M	A
f1	Reutilização de código instável.	Utilização de rotinas prontas com necessidade de ajustes.	Necessidade de retrabalho; atraso no cronograma.	A	M	M
p3	Rotatividade de profissionais.	Um membro da equipe sair da empresa.	Perda de desenvolvedores; necessidade de substituição.	M	M	B
e1	Falta de experiência da equipe com a tecnologia	Tamanho do projeto estimado maior que o real.	ajuste do orçamento e do cronograma.	M	M	B
o1	Necessidade de conhecer novos recursos	Resistência dos usuários na utilização.	Cliente não conseguir implantar a nova tecnologia.	B	M	B
t1	Custo x benefício em relação à tecnologia adquirida.	Tecnologia não atende às expectativas.	Aumento dos custos; adaptação com o recurso existente.	B	B	B

Fonte: adaptado de Wazlawick (2013) e Sommerville (2018).

Após esse processo deverão ser criados os planos de mitigação e redução da probabilidade e do impacto. Lembre-se de que o plano de redução da probabilidade é baseado na causa e o de redução de impacto é baseado no efeito.

Para essa etapa, cabe a você definir como realizar essa tarefa, pois, com certeza, está diretamente relacionada a sua realidade, de domínio de conhecimento na área de desenvolvimento em TI, e à forma como sua organização é governada, além das realidades do projeto, mediante a tecnologia escolhida pela equipe, o que demandará a busca de conhecimento específico ou prévio.

E para concluir este estudo, é importante que observe o gerenciamento de risco como um aliado a impedir a “perda” de tempo ou o retrabalho. E por tanto gerar um plano de monitoramento e controle, é bem interessante definir o período de revisão dos riscos, os responsáveis por acompanhar cada um deles e os integrantes que conseguirão agir para a contingência de cada um deles.

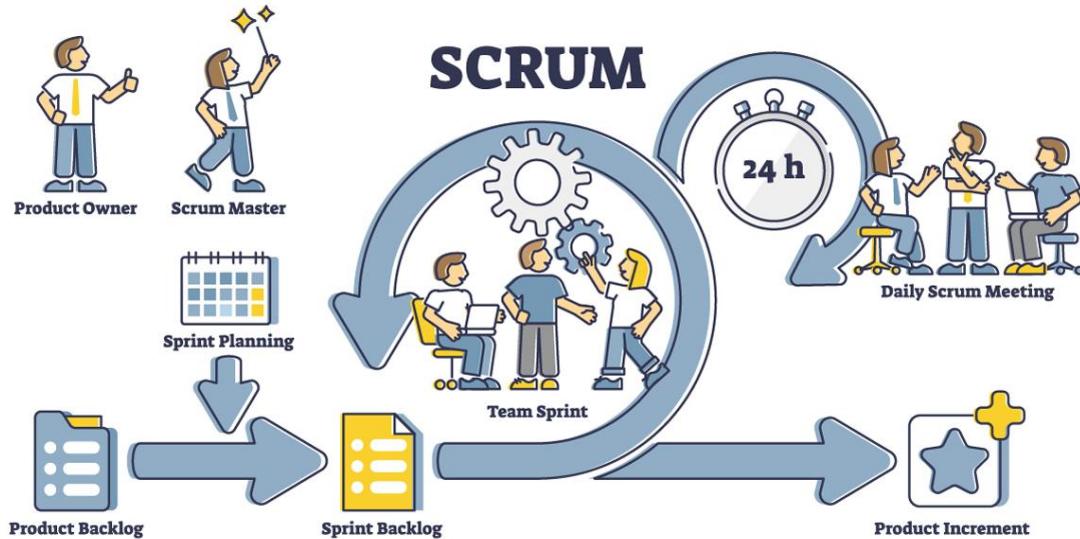
Esperamos que tenha compreendido a importância de se gerenciar os riscos dos projetos de desenvolvimento de software utilizando técnicas de projetos normatizadas. É certo que o crescimento profissional a partir desses conhecimento capacitará todos os profissionais para gerir não só os riscos de natureza técnica, de projeto, de tempo, de qualidade, de negócio e todos os outros que surgirem por variáveis, muitas vezes, incontroladas pela organização ou equipe, mas para lidar com as situação de relacionamento interpessoal, focando a empatia e a ética no trabalho, ou seja, a gestão de conflitos.

## Unidade 3- seção 2

# *NÃO PODE FALTAR*

## *GESTÃO DA QUALIDADE*

Daniela Teresa Rossignoli Uebele



Fonte: Shutterstock.

**Deseja ouvir este material?**

### **PRATICAR PARA APRENDER**

Caro aluno, quando buscamos por qualidade, logo pensamos em custo alto. Creio que seja uma relação parcialmente equivocada, no entanto, qualidade no desenvolvimento de software é, hoje, um requisito mercadológico indiscutivelmente essencial.

No ponto de vista do seu cliente, bem pago, para o desenvolvimento do sistema, já inclui, impreterivelmente, a questão da qualidade. Quando entende que a necessidade dele é um sistema que gerenciará ou otimizará os processos de sua empresa, é porque ela se encontra em um momento de estagnação e necessita de um diferencial, ou seja, de **QUALIDADE**, que tem que fazer parte do pacote do software a ser desenvolvido, bem como muitas análises de riscos, a fim de que aja o controle de cada um deles, além do comprometimento com o tempo, o custo e o escopo do produto.

A empresa contratante do desenvolvimento de um sistema personalizado espera que seu novo projeto lhe traga diferenciais estratégicos ou resolva problemas recorrentes. Não estamos falando de mais um sistema e sim daquele que manterá a empresa competitiva no mercado em que atua, talvez até revolucionar a área de atuação.

Portanto, visando a atender às expectativas e aos anseios de uma organização, a gestão da qualidade deve ser implantada no processo de planejamento, execução e controle dessa qualidade esperada. Sendo assim, faz-se necessária a implantação de

metodologias de projetos buscando meios de se definir, medir e controlar o sistema de qualidade do Software.

A equipe de desenvolvimento e o cliente, por meio dos documentos gerados no processo de iniciação do projeto, do termo de abertura, do planejamento das atividades, dos riscos, da comunicação e de todas as 10 áreas do conhecimento trabalhadas pelo PMBOK, terão de definir como garantir a qualidade. Para tanto, existem inúmeras normativas que podem ser acrescidas nesse processo, aqui veremos a ISO 25010.

Serão trabalhados modelos de qualidade para se buscar ferramentas já utilizadas, facilitando esse processo de definição de pronto, sendo visto como um fator de qualidade. Será por meio deles e de outras técnicas que será possível traçar métricas, definir o que pode ser dito como qualidade no desenvolvimento de um software, já que este não sofre desgaste, porém necessita de muito planejamento para uma longa vida útil; sua manutenção deverá ser eficiente e eficaz; a segurança do sistema, os dados e as informações deverão ser garantidos, como muitas outras variáveis que serão estudadas.

Voltemos a estudar o projeto em que você e a sua equipe terão que desenvolver um sistema. A fim de atender à área da saúde, essa ferramenta será utilizada dentro da sala de cirurgia para que os médicos possam manipular radiografias de raio-X, tomografias e ultrassonografias sem a necessidade de remover as luvas, portanto, serão utilizados recursos tecnológicos de identificação de movimentos e/ou voz com a junção de realidade virtual, visando à minimização do tempo do procedimento cirúrgico, pois o profissional será poupar de ter que, novamente, higienizar as mãos para voltar ao procedimento, trazendo benefícios também ao paciente que passará menos tempo anestesiado, liberando o centro cirúrgico para outros procedimentos.

Para promover a qualidade necessária, visando à utilização de sistema de qualidade de software, faz-se necessária a criação dos requisitos funcionais necessários para o desenvolvimento do sistema, criando, a partir desses requisitos, a definição dos requisitos não funcionais, que facilitará a análise dos aspectos externos e internos do desenvolvimento do sistema. Com essas informações definidas, é viável a criação dos critérios de qualidade para cada um dos requisitos funcionais e não funcionais, visando à geração da definição de pronto (DoD).

Utilizando as características definidas pela ISO 25010 e suas subcaracterísticas, verifique as aplicáveis aos sistemas e como seriam efetuados os testes para responder os questionamentos, a fim de se verificar se o sistema desenvolvido está atendendo a tais características sugeridas.

Esses processos devem envolver a equipe de desenvolvimento, portanto, solicitar a ajuda de todos os envolvidos será de grande importância para o sucesso da busca pela qualidade. Crie o quadro com o conceito de Kanban, com 3 colunas divididas em TO DO (Fazendo), *Work in Progress* (WIP) desenvolvendo e *DONE* (pronto); insira as atividades a serem executadas inicialmente e escreva, logo abaixo, os critérios de pronto, assim, tornaremos transparente o processo de atividade concluída.

A busca por qualidade e a compreensão dos conceitos e das formas de aplicá-la no desenvolvimento do software levarão você a ser um profissional não apenas capacitado, com habilidades técnicas, mas com qualidade.

## **CONCEITO-CHAVE**

### **GESTÃO DA QUALIDADE**

**QUALIDADE!** O que essa palavra representa para você? Qualidade pode ser observada por inúmeros olhares. Buscando uma interpretação ampla nos dicionários, é definida pela forma como se pode distinguir pessoas e objetos devido às suas propriedades e aos seus atributos de forma a possuir diferenciação significativa segundo a sua natureza, ou seja, tudo que fazemos ou criamos pode e deve ter qualidade. Se recorrermos ao conteúdo da seção anterior, a mitigação de riscos compreende uma das formas de se buscar a qualidade, já que, no momento em que se reduzem os riscos que levam ao atraso ou aumento dos custos, talvez até à mudança do escopo, já temos a perda da qualidade, e essa compreensão nos permite uma análise ampla sobre a qualidade de um projeto de desenvolvimento de software (COSTA; PEREIRA, 2019).

A fim de gerirmos a qualidade, observaremos, no estudo desta seção, que a qualidade tem inúmeras formas de ser mensurada e que existe a possibilidade de se **adequar essas métricas em diferentes metodologias de projeto**.

Ao analisar o contexto de qualidade do ponto de vista do mercado de desenvolvimento de software, que é altamente competitivo, e levando em conta que na década de 1950 não se imaginava o quanto importante seria um software, possuindo a imensa capacidade de manipular um enorme volume de dados e gerando informações de alto grau de complexidade, comprovando a Lei das “consequências não pretendidas” (PRESMAN, 2006).

Portanto, o gerenciamento da qualidade inclui analisar os aspectos tangíveis (aspectos que são tocados / visíveis / usável / mensurado) e intangíveis (os valores que não podem ser tocados ou medidos), e depende diretamente da relação de entrega do sistema e/ou dos serviços que é capaz de oferecer, ou seja, um produto diferenciado é fundamental para fortalecer o nome da empresa desenvolvedora.

Dentro do atual sistema mercadológico não se fala apenas em entrega de produtos dentro do prazo e custo, que funciona conforme as expectativas do cliente, mas também em vender uma experiência, de forma que ela se torne inesquecível, ofertando ao seu cliente qualidade do início ao fim do processo de venda, o que nos leva a mais um questionamento: como garantir a qualidade em projetos de desenvolvimento de software? Podemos pensar da mesma forma que pensamos ao buscar a qualidade para qualquer outro projeto? Resposta: utilizando-se a gestão do triângulo de ferro, conservando o custo, o tempo e o escopo do projeto, conforme previsto, assim, atendendo à qualidade no que diz respeito ao aspecto tangível, aquele que foi previamente acordado mediante as restrições iniciais de qualquer projeto (COSTA; PEREIRA, 2019).

### **IMPORTÂNCIA DA QUALIDADE**

Para entender a importância que a qualidade vem tomando desde os anos 1980. Existem inúmeras instituições reconhecidas que disponibilizam normas para permitir o planejamento, desenvolvimento e controle da qualidade, inclusive específicas para softwares. Vamos citar algumas normas nacionais e internacionais:

- NBR 13596: versão nacional da ISO 9126 falando sobre a qualidade de produtos de software, fazendo parte da família ISO 9000.
- NBR ISO 9001: padronização para a garantia da qualidade em projetos, desenvolvimento, instalação e processos – Sistema de qualidade.
- NBR ISO 9000-3: utilização da ISO 9000 no processo de desenvolvimento de software, ou seja, gestão e garantia da qualidade.
- ISO 15498: é um guia que traz diretrizes para avaliação dos produtos de software, criado a partir da ISO 9126.
- ISO 12119: desenvolvido para os softwares genéricos, vendidos em prateleiras, em que a empresa se adequa ao produto e não o produto à empresa; normas que determinam as características de qualidade de pacotes de software.
- IEEE P1061: *Standard for Software Quality Metrics Methodology* - normativas para tratar métodos de padronizar a qualidade de software, incluindo algumas formas de medição.
- CMMI *Capability Maturity Model Integration*: Modelo da SEI (Instituto de Engenharia de Software do Departamento de Defesa dos USA): não se enquadra nas normas ISS, porém é amplamente aceita para a avaliação da qualidade dos processos de desenvolvimento do software;
- SPICE ISO 15504 ou apenas SPICE: melhora os processos de desenvolvimento de software, semelhante a ISO 12207, referente à parte de ciclo de vida de Software.

Além dessas, existem muitas outras, por isso a importância de um amplo conhecimento sobre o que é qualidade, como aplica-la, controlar, medir. São inúmeros contextos entrelaçando planejamento, execução e monitoramento (WEBER *et al.*, 2015).

#### **EXEMPLIFICANDO**

A busca por qualidade no desenvolvimento de software é bem visível quando falamos sobre os sistemas web e os apps dos bancos; não basta nos oferecer os serviços e recursos necessários para a movimentação financeira. Os clientes, no início, tiveram uma grande resistência a sua utilização, afinal, não tinham como saber se o acesso via rede mundial era seguro. Foi necessário mais que um software bem desenvolvido; a segurança e os meios de demonstrar ao cliente que o banco também se preocupava com esse aspecto intangível foram essenciais, pois só era possível mensurar a inexistência de falhas de segurança – sinônimo de qualidade.

Tudo isso se transformou em um ciclo de “não uso, pois não sei se é seguro” e “preciso que usem para comprovar a qualidade do sistema mediante a segurança”. Para isso, a maioria dos bancos, inicialmente, investiu na divulgação; se houvesse problema, o banco assumiria a responsabilidade. Após esse processo, a busca, hoje, está pelas funcionalidades, pelo fácil acesso, pelo rápido entendimento e pelo acesso eficiente.

#### **PROCESSOS RELACIONADOS AO GERENCIAMENTO DA QUALIDADE DO PROJETO**

A principal preocupação quanto ao gerenciamento da qualidade durante o desenvolvimento de software é a busca por garantir que o sistema de software a ser desenvolvido seja “adequado para seus objetivos”, atendendo às necessidades dos usuários, proporcionando confiabilidade, usabilidade e eficiência em seu funcionamento. Nos últimos 20 anos, a utilização da gestão da qualidade, de novas

tecnologias e métodos de testes por meio de softwares trouxe melhorias importantes para o aumento da qualidade do produto desenvolvido (SOMERVILLE, 2018).

O PMBOK de 2018 indica três processos relacionados ao gerenciamento da qualidade do projeto:

- Planejar o gerenciamento da qualidade.
- Gerenciar a Qualidade.
- Controlar a Qualidade.

### MÉTRICAS DA QUALIDADE

Para alcançar uma compatibilidade com a qualidade, é importante criar meios de medi-la. A satisfação do cliente está no atendimento de suas expectativas, portanto, compreender, controlar e influenciar as necessidades dos clientes são caminhos para atender à qualidade desses *stakeholders*. A gestão da empresa possui a responsabilidade de garantir a qualidade, uma vez que deve fornecer recursos adequados para que o planejamento seja executado. A qualidade do produto não está desvinculada dos interesses organizacionais, como o custo e prazo (COSTA; PEREIRA, 2019).

A melhoria contínua é uma necessidade do ciclo conhecido como PDCA, que significa:

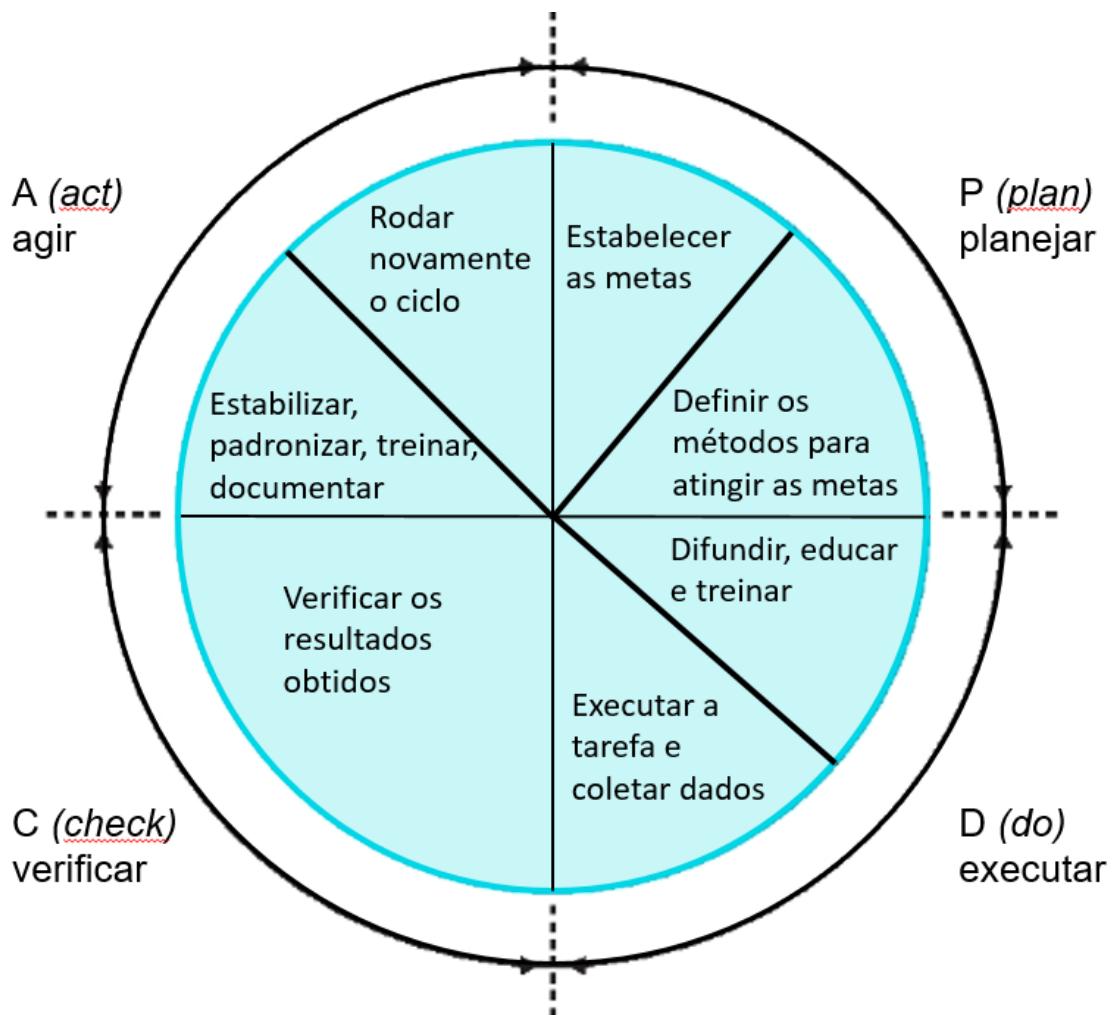
**Plan** = planejar – Estabelece metas e define os métodos necessários para se obter os objetivos do processo.

**Do** = fazer – Executa, difundi, treina, educa as tarefas e coletas os dados.

**Check** = verificar – Faz a verificação dos resultados.

**Act** = agir – Realiza ações corretivas utilizando-se de padronizações, treinamentos e documentos dos processos.

Figura 3.3 | Ciclo PDCA: planejar (*plan*), executar (*do*), verificar (*check*) e agir (*act*)



Fonte: Carvalho (2018, p. 187).

Esse ciclo deve ser utilizado constantemente e a qualquer momento do projeto, pois o objetivo de melhoria contínua exige essa constância; além disso, esse recurso se utiliza de outras ferramentas capazes de trazer meios de se medir a qualidade (CARVALHO, 2018).

#### GERENCIAMENTO DA QUALIDADE TOTAL (GQT)

A utilização do **gerenciamento da qualidade total** (GQT) visa à conscientização da qualidade em todos os processos de uma empresa, incluindo os *stakeholders* externos (fornecedores, distribuidores e parceiros); o uso do **seis Sigma** ( $6\sigma$ ), que se refere à eliminação de defeitos por um conjunto de boas práticas sistêmicas na melhoria de processos (DMAIC - definir, medir, analisar, melhorar e controlar), visa a resolver problemas de causa desconhecida; e **Lean Seis Sigma**, que é a junção da metodologia seis sigma com as ferramentas e os conceitos do **Lean Manufacturing**, traz a melhoria dos resultados, minimizando as variações nos processos, eliminando desperdícios e as atividades que não agregam valor ao projeto, bem como aprimora o gerenciamento e a qualidade do produto, pois, para o desenvolvedor, as características internas do desenvolvimento do sistema está na legibilidade, testabilidade e eficiência da criação e manutenção dos sistemas desenvolvidos. A prevenção em vez da inspeção é uma

necessidade real, pois o custo do retrabalho é muito maior do que a prevenção da existência de falhas (SOMERVILLE, 2018).

Uma das principais ferramentas para o processo de garantia da qualidade é a auditoria, permitindo uma revisão totalmente independente e benefícios como: as lições aprendidas que trazem benefícios para os futuros e atuais projetos; visualização de lacunas, não conformidades e deficiências do planejamento; avaliação dos fornecedores por meio das verificações de atendimento dos requisitos; e a melhoria contínua da organização por meio do giro do ciclo (PDCA) (COSTA; PEREIRA, 2019).

### GESTÃO DA QUALIDADE FORMAL

Observe, mais uma vez, que a qualidade não está embasada exclusivamente na entrega do software dentro do prazo, do custo e do escopo predefinido, mas é importante que se tenham boas práticas durante o desenvolvimento, que sejam criadas documentações adequadas, não apenas um manual do usuário, e linhas de código documentadas, bem como se tenha uma organização estrutural das bibliotecas de desenvolvimento e que haja utilização das boas práticas para a análise antes mesmo da codificação, gerando um sistema de fácil manutenção.

Portanto, quando se pensa em equipes de desenvolvimento de um software, logo vem à mente a ideia de um sistema dito como grande, devido à quantidade de funcionalidades ou mesmo por se utilizar de tecnologias inovadoras, já que tais sistemas, possivelmente, terão longa vida de utilização e levarão alguns anos para ser concluídos. Para esses casos, a gestão da qualidade formal é crucial para o sucesso do projeto, e isso demandará a necessidade de recursos compartilhados entre todos os membros, oferecendo um recurso de acesso constante e documentações para se definir o que é qualidade (CARVALHO, 2018).

Existe uma certa dificuldade em definir se um sistema cumpre ou não as especificações. A interpretação dos requisitos pode ser equivocada em sua descrição tanto pelo cliente quanto pelos desenvolvedores, tornando o processo de descrição dos requisitos equivocado e incompleto. Por existir inúmeros *stakeholders* envolvidos no projeto, é provável que alguns requisitos fiquem fora do desenvolvimento, dessa forma, para alguns, o sistema será de baixa qualidade. O processo de manutenção, por exemplo, não dá para ser totalmente mensurado, o que torna inviável a medição de algumas características do sistema desenvolvido (SOMERVILLE, 2018).

Para uma exemplificação de métricas de qualidade, pode-se criar uma tabela utilizando ferramentas de planilha eletrônica ou softwares específicos, desde que estes permitam o acesso compartilhado com todos os membros. O exemplo para visualização será feito no formato retrato, sendo melhor desenvolvido em paisagem, pois não haverá apenas 2 requisitos como no exemplo do Quadro 3.9.

Cód.	1	
Requisito	Consistência e validação dos dados a serem inseridos ou alterados.	Criação de relatórios
Indicador	IQ01 – verificação das validações de tipo de dado e sua validade no momento da digitação ou finalização do campo.	IQ02 –
Tolerância	>= 99%	
Métodos de medição	Testes manuais. Verificação de código utilizando bibliotecas e API's predefinidos pela equipe. O usuário, ao utilizar o sistema, verifica as entradas de dados.	Cronômetro para criação do relatório de criação
Periodicidade	Diária	
Quem mede	Equipe de desenvolvimento, usuário final.	Equipe
Doc. de registro	Formulário FIQ01	
Ação corretiva*	Orientar os atendentes para atenção ao processo de cadastro. Anotação de qualquer tipo de falha.	Identificar os resultados do processo
Quando medir*	A cada medição.	
Responsável*	Gestor do projeto e equipe desenvolvedora.	Supervisor

Fonte: adaptado de Carvalho (2018).

#### REFLITA

Você já pesquisou os bugs que causaram não apenas perdas financeiras, mas também perdas humanas? No mundo moderno da informação e da informatização, erros computacionais são, por vezes, destruidores.

Existem alguns bugs que são realmente assustadores, e as falhas estão ligadas à falta de métodos de análise do sistema, de planejamento de testes e utilização de métricas de qualidade, o que leva à falta de qualidade no processo de desenvolvimento. O erro do hardware foi, por muitas vezes, um dos mais preocupantes, porém já se vão alguns anos que o software tem apresentado preocupações maiores.

O processo de qualidade para todo e qualquer sistema é de fundamental importância, pois os prejuízos financeiros, psicológicos e de saúde também podem ser causados por consequência de sistemas mal projetados, testados e entregues com descasos.

#### Evolução dos sistemas de qualidade de software

No final dos anos 1980, surgiu o método de programação orientada a objetos; já em 1997, surgiu o método de análise *Unified Modeling Language* (UML), amplamente utilizado por permitir que a análise do sistema seja desenhada e discutida entre a equipe de gestão e de desenvolvimento, reduzindo o tempo de trabalho de codificação e do retrabalho. Em 2001, por sua vez, a criação do manifesto ágil trouxe a flexibilização do

enrijecimento causado nos anos anteriores, em busca da qualidade na produção de sistemas de software com a visão equivocada embasada no conceito da qualidade de um produto de linha de produção; por fim, a partir de 2002, inúmeros processos ágeis foram sugeridos para que o cliente pudesse interagir no decorrer do desenvolvimento do software, buscando-se o alinhamento e o dinamismo do escopo com o mercado e, por consequência, a entrega de um sistema funcional, visando ao sistema de qualidade de software (ROCHA, 2001; WEBER, 2008; SOMERVILLE, 2018).

Dentro dessa evolução em busca de **sistemas de qualidade de software**, passou-se a existir novas formas de se trabalhar. A metodologia Scrum surgiu para agregar mais produtividade nos processos; o conceito Lean visa a eliminar toda atividade que não é prioridade, evitando o comprometimento do cronograma; já o conceito cascata visa a um controle de processo linear de forma rigorosa, começando uma nova etapa apenas quando a anterior está concluída, opondo-se à metodologia ágil, que visa ao próprio projeto ou produto, permitindo melhorias e mudanças com maior frequência, baseadas no feedback do cliente. E além dos métodos gerenciais, existem inúmeras linguagens para desenvolver sistemas adequados a cada cliente, com muita personalização e personificação, como: sistemas web, *web services*, *cloud computing* (software como serviço); aplicativos mobile, IoT (*Internet of Things*) entre outros.

Os inúmeros processos de desenvolvimento de software utilizam atividades como: **análise de requisitos** – momento em que se toma conhecimento do real problema para o desenvolvimento do sistema; **design do software** – momento de planejar a solução do problema apontado na documentação dos requisitos; **código** – transforma em código o que foi planejado no designer do sistema; e **teste** – a busca por defeitos/falhas e um meio de se medir e controlar a qualidade do sistema desenvolvido (JALOTE, 2005).

Nesse processo de análise de requisitos, a metodologia UML é uma excelente forma de se modelar (desenhar) e documentar analisando os requisitos funcionais, ou seja, quais são as funcionalidades que o seu sistema terá de oferecer ao usuário. Exemplo de requisitos funcionais para um consultório de atendimento médico:

- RF01 – O sistema deve gerar a lista de consultas do dia para cada um dos médicos.
- RF02 – Os usuários devem ser identificados por um código de 6 dígitos.
- RF03 – As secretárias podem fazer pesquisas referentes às consultas de qualquer um dos médicos.
- RF04 – O sistema deve gerar, no final da semana, um relatório contendo quantas consultas foram e como os planos de saúde.

Também devem ser definidos os requisitos não funcionais, que estão relacionados a restrições como tempo, imposições de padronizações, entre outros. Existem métricas para se especificar os requisitos não funcionais, o que leva a incluir e determinar as métricas de qualidade.

Tabela 3.1 | Métricas para especificar requisitos não funcionais

Propriedade	Métrica
Velocidade	Transações processadas/segundo.

Propriedade	Métrica
	Tempo de resposta do usuário/evento.
Tamanho	Tempo de atualização da tela. Megabytes/número de chips de ROM.
Facilidade de uso	Tempo de treinamento.
Confiabilidade	Número de quadros de ajuda. Tempo médio até a falha.
	Probabilidade de indisponibilidade.
	Taxa de ocorrência de faltas.
Robustez	Disponibilidade. Tempo para reiniciar após a falha.
	Porcentagem de ventos, causando falhas.
Portabilidade	Probabilidade de corromper dados em uma falha. Porcentagem de declarações dependentes do sistema-alvo.
	Número de sistemas-alvo.

Fonte: Sommerville (2018, p. 94).

Os requisitos não funcionais pensando no sistema do exemplo acima, temos:

- RNF01 – Validação da entrada de dados do paciente.
- RNF02 – Apenas usuários autorizados têm acesso às respectivas funcionalidades.
- RNF03 – Tempo de criação dos relatórios deve ocorrer em <=30segundos.

Esse processo está ligado diretamente com a questão da qualidade, principalmente no que se refere aos objetivos do sistema, portanto, o momento de análise por meio do UML permite que os desenvolvedores busquem ser objetivos e claros a partir de pequenos textos e desenhos de diagramas do sistema a ser desenvolvido, sendo utilizado por toda a equipe de desenvolvimento e pelo cliente. A ferramenta é considerada bastante eficiente no que diz respeito à comunicação entre os membros da equipe de desenvolvimento de sistemas.

### ■ PLANEJAMENTO DE GESTÃO DE QUALIDADE

Para se definir um planejamento de gestão de qualidade, executar e monitorar, é preciso entender os objetivos do projeto, ou seja, conhecer os requisitos (conforme apresentado no exemplo da metodologia UML), as normas e as diretrizes em que o escopo do projeto está sujeito. A partir desse entendimento, é possível definir os requisitos e as especificações que atenderão à qualidade do produto a ser desenvolvido, atestando, assim, a qualidade em busca do efetivo gerenciamento (CARVALHO, 2018).

Além do guia PMBOK, é preciso utilizar outros processos das áreas de conhecimento, como o acesso ao termo de abertura e o plano de projeto, em que também temos as informações dos custos estimados e do prazo para o encerramento de cada atividade. Com essas informações, é possível identificar os requisitos, as respectivas datas de entregas e definir a qualidade dessas atividades, visando a atender aos padrões definidos pela equipe (COSTA; PEREIRA, 2019).

A escrita do plano de qualidade deve ser mantida com descrições curtas; caso isso não ocorra, provavelmente, as pessoas deixarão de ler o documento e, evidentemente, o propósito desse processo será perdido (SOMERVILLE, 2018).

Após compreender o planejamento do gerenciamento da qualidade integrado às demais áreas do conhecimento, é preciso buscar ferramentas para a análise por meio dos modelos de qualidade, como a análise de custo-benefício, fluxograma, análise de lacunas entre expectativa e percepção, entre outros métodos existentes, bem como escolher o mais adequado para a equipe e a organização (PMI, 2017).

A partir da utilização de modelos de qualidade, temos alguns produtos resultantes do planejamento da qualidade, como: o plano geral de gerenciamento da qualidade; *checklist* que proporciona ao profissional a entrega dos requisitos; planos de melhorias que evitam que processos saiam do controle; e atualizações que ajudam no gerenciamento das comunicações das partes interessadas (CARVALHO, 2018).

As métricas de qualidade definem os requisitos de sucesso do projeto, apontando medidas reais de como a entrega será considerada concluída. Elas fazem parte do plano de gerenciamento da qualidade e devem descrever com detalhes os atributos/as funcionalidade do produto ou processo, ou seja, quando uma tarefa é considerada concluída e quem será o *stakeholder* responsável pela sua aprovação. Os padrões e critérios de aceitação foram predeterminados para cada requisito no processo inicial (COSTA; PEREIRA, 2019).

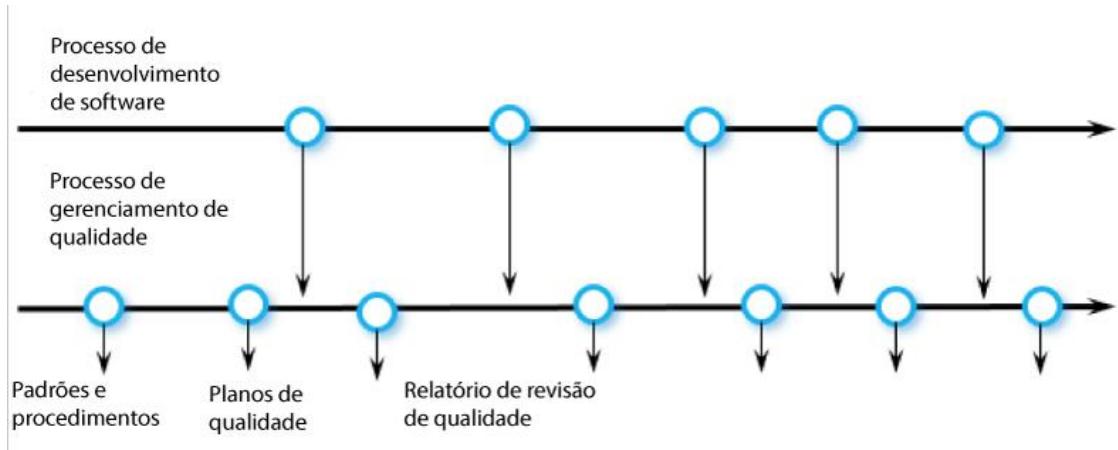
### **GARANTIA DA QUALIDADE DO PROJETO**

Na etapa seguinte, temos a garantia da qualidade do projeto, em que ocorrem as auditorias internas e/ou externas com as revisões de qualidade, esse processo visa à melhoria contínua. O último item a ser pensado diz respeito ao controle da qualidade do projeto, em que são descritos os procedimentos de inspeção para que o monitoramento ocorra conforme o planejamento desse ciclo de vida do projeto (COSTA; PEREIRA, 2019).

Buscar a excelência do sistema de qualidade de software por meio de padronizações é um caminho proposto desde a Segunda Guerra Mundial e foi potencializado com os métodos Ford para criar padronizações na produção e conseguir avaliar, ao final da produção, se o produto possui a qualidade necessária para ser ofertado ao cliente. “ISO”, em grego, significa igual; já em inglês, significa *International Organization for Standardization* (Organização Internacional para a Padronização).

Baseadas nesses conceitos, foram criadas formas de se gerenciar o desenvolvimento de software. A equipe de gerenciamento da qualidade deve ser independente da equipe do processo de desenvolvimento.

Figura 3.4 | Linha dos processos de desenvolvimento de software e de gerenciamento da qualidade



Fonte: Somerville (2018 p. 664).

A equipe de qualidade verificará os resultados a fim de garantir que estejam conforme os padrões e as metas da organização. Essa verificação ocorrerá por meio de documentos, a fim de se verificar se as tarefas relevantes formam todas concluídas ou se ocorreu alguma suposição incorreta referente às atividades de outro grupo.

#### ASSIMILE

A gestão da qualidade de software visa a garantir que um software possua poucos defeitos, atingindo um padrão aceitável e pré-acordado de manutenção, confiabilidade, segurança e todos os requisitos que definem qualidade em sistemas de software (SOMMERVILLE, 2018; NEWTON, 2011).

Essas etapas descritas servem para todo e qualquer projeto de forma genérica, mas quando falamos de desenvolvimento de software, devemos acrescentar a qualidade subjetiva, que está baseada nas características não funcionais, que refletem na experiência do usuário. Caso a funcionalidade não atenda às suas necessidades, arrumará meios de contornar a deficiência e atingir seu objetivo. Porém, se o software não for eficiente ou mesmo confiável, isso fará com que o usuário não alcance seus objetivos.

Portanto, não basta saber se a funcionalidade está implementada e funcionando, é necessário que os atributos não funcionais atendam às expectativas do cliente.

Tabela 3.2 | Atributos não funcionais da qualidade de software

Segurança ( <i>safety</i> )	Compreensibilidade
Segurança da informação ( <i>security</i> )	Testabilidade
Confiabilidade	Adaptabilidade
Resiliência	Modularidade

Segurança ( <i>safety</i> )	Compreensibilidade
Robustez	Complexidade

Fonte: Somerville (2018, p. 667).

Infelizmente, é inviável que se consiga atender plenamente a todos esses atributos não funcionais, por exemplo: a perda de desempenho poderá ser reduzida caso aumente de mais o sistema de segurança. No caso do plano de qualidade para o desenvolvimento de software, devemos incluir a definição do processo de avaliação da qualidade mediante a escolha do atributo de maior relevância, portanto, se a escolha se der pelo desempenho, outros fatores poderão ser sacrificados em prol da escolha.

Outra questão a ser analisada é o desenvolvimento que se dá por meio de um processo criativo, o que significa que a experiência e as habilidades individuais influenciam diretamente nos resultados, o que, fatalmente, poderá sofrer influência por fatores externos, como a pressão para o encerramento do projeto ou o lançamento de novos produtos no mercado. Embora a base da qualidade se apoie em padrões e processos, os gerentes de qualidade se destacam ao reconhecer a existência dos aspectos intangíveis, como a elegância, legibilidade, confiabilidade, manutenibilidade etc., que não conseguem ser inseridas aos padrões. Devendo ser estimuladas por meio do comportamento profissional da equipe em todos os membros (SOMERVILLE, 2018).

Considerando tudo o que foi definido até aqui, é preciso garantir a qualidade, monitorar e controlar a execução desse plano, de forma a assegurar o andamento das atividades até o seu encerramento, contribuindo com a minimização dos defeitos de planejamento ou eliminando-os nas inspeções, durante o andamento do projeto (COSTA; PEREIRA, 2019).

As inspeções de programas ocorrem por meio das revisões feitas por pares, em que os bugs são encontrados pelos próprios integrantes da equipe, fazendo, assim, parte do processo de verificação e validação de software, os sistemas em desenvolvimento contam com a possibilidade de verificação, utilizando as representações dos modelos UML. Nas reuniões de inspeções, os membros que realizaram a revisão linha a linha do código-fonte os descrevem para a equipe. Para esta atividade é recomendado o uso de um checklist, podendo ser criado pela equipe ou utilizando modelos (templates) já sugeridos, estes itens irão sofrer variações conforme a linguagem de desenvolvimento (SOMERVILLE, 2018).

## MÉTODOS DE MEDIÇÃO DE QUALIDADE

Uma das grandes dificuldades da gestão da qualidade de desenvolvimento de software são os métodos de medição de qualidade, afinal o software não sofre desgaste. A ISO/IEC 25010, de 2011, propõe um modelo que define oito categorias de características sobre a qualidade, sendo estas divididas em subcaracterísticas, conforme Quadro 3.10.

Quadro 3.10 | Modelo de qualidade de produto de sistemas proposto pela Organização Internacional para Padronização e pela IEC - ISO/IEC 25010, que se relacionam com propriedades estáticas do software e propriedades dinâmicas do sistema de computador

Características	Subcaracterísticas	Significado
Adequação funcional: satisfaz as necessidades explícitas e implícitas para se atingir a finalidade do produto?	Completude funcional	É adequado às necessidades do usuário?
	Correção	Faz o que é proposto de acordo com a necessidade?
	Adequação funcional	As funções realizam as tarefas de forma objetiva e fácil?
Eficiência: os recursos e os tempos utilizados são compatíveis com o nível de desempenho requerido para o produto?	Comportamento do tempo	O tempo de resposta e de processamento atende aos requisitos?
	Utilização de recursos	Quanto recurso utiliza? Durante quanto tempo?
	Capacidade	Os limites máximos atendem aos requisitos?
Compatibilidade: a troca de informações entre produtos, sistemas, componentes e/ou execução das suas funcionalidades ao compartilhar o mesmo ambiente de hardware ou de software.	Coexistência	As funcionalidades são eficientes ao compartilhar o ambiente comum e os recursos com outros produtos? Existe o impacto com algum outro produto?
	Interoperabilidade	Dois ou mais sistemas, produto ou componentes trocam informações e as usam com eficiência e segurança?
Usabilidade: é fácil de usar o software com eficiência, eficácia e satisfação em um contexto de uso específico?	Reconhecimento de adequação	O usuário reconhece que o produto ou sistema é adequado as suas necessidades?
	Capacidade de aprendizagem	O usuário consegue aprender a usar o produto ou sistema com eficácia, eficiência, isenção de risco e satisfação em um contexto de uso especificado?
	Operabilidade	É fácil operar e controlar?
	Proteção ao erro do usuário	O sistema evita que o usuário cometa erros?
	Estética da interface do usuário	A interface do usuário permite uma interação agradável e satisfatória?
	Acessibilidade	O sistema pode ser usado por pessoas com a mais ampla gama de características e capacidades,

Características	Subcaracterísticas	Significado
		(deficiências) para atingir uma meta especificada?
Confiabilidade: durante um período, continua funcionando de acordo com as condições preestabelecidas?	Maturidade	Atende às necessidades de confiabilidade sob operação normal?
	Disponibilidade	O tempo em que o sistema está operacional e acessível para uso?
	Tolerância a erro	Apesar das falhas de hardware ou software, o sistema permanece funcionando?
	Recuperabilidade	É capaz de recuperar dados após falha e restabelecer o sistema?
Segurança: o sistema protege as informações e os dados, fornecendo o grau de acesso adequado para seu tipo e nível de autorização.	Confidencialidade	Existe a garantia de que os dados estão restritos a determinados usuários?
	Integridade	O sistema impede o acesso não autorizado ou modificações no código?
	Não repúdio	Existe um controle das atividades dos usuários de forma que eventos ou ações não sejam repudiados posteriormente?
	Prestação de contas	As ações de uma entidade podem ser rastreadas exclusivamente para essa entidade?
	Autenticidade	Há garantia de que o usuário ou recurso pode ser comprovado?
Manutenibilidade: há eficácia e eficiência para correções, atualizações e alterações?	Modularidade	A mudança em um componente tem impacto mínimo em outros componentes?
	Reutilização	Um ativo pode ser usado em mais de um sistema ou na construção de outros ativos?
	Analisabilidade	É eficaz a avaliação do impacto de uma alteração pretendida ou o diagnóstico de uma falha quando ocorre? E a modificação?

Características	Subcaracterísticas	Significado	Fonte:
	Modificabilidade	É eficaz e eficiente modificar sem inserir defeitos ou degradar a qualidade do produto existente?	
	Testabilidade	Há eficácia e eficiência nos critérios de testes realizados para se determinar se esses critérios foram atendidos?	
Portabilidade: é utilizável o produto em plataformas diferentes com pequeno esforço de adaptação?	Adaptabilidade	É fácil adaptar a outros ambientes sem aplicar outras ações ou meios além dos fornecidos para essa finalidade no software?	
	Capacidade para ser instalado	É eficaz e eficiente instalar e desinstalar em ambiente específico?	
	Capacidade para substituir	É fácil substituir por outro software que possui a mesma finalidade ao atualizar?	

adaptado de ABNT (2011).

Foi apresentado apenas o modelo de qualidade do produto; falamos de uma única normativa de um conjunto sugerido pelo ISO/IEC 25010, portanto, como se observar no PMBOK, é importante o devido aprofundamento para cada necessidade, e sua implementação deve ocorrer de forma parcial ou completa e adequada a cada realidade.

### CONTROLE DA QUALIDADE

No processo de controle da qualidade é onde ocorre o registro dos resultados da execução das atividades, a avaliação do desempenho e as recomendações necessárias para a melhoria contínua. Para essa etapa, os resultados que podem ser obtidos são: as validações das mudanças, as informações referentes ao desempenho do desenvolvimento, os documentos com as solicitações de mudanças, a atualização dos documentos conforme necessidade e a verificação das entregas. Portanto, esse processo de controle de qualidade deve ocorrer nas fases de execução e finalização, para que se possa mostrar ao cliente que todos os critérios definidos inicialmente foram cumpridos (COSTA; PEREIRA, 2019).

Para uma ilustração desse processo de encerramento, vamos observar um documento com solicitação de mudanças e os resultados obtidos com ele.

Quadro 3.11 | Sugestão básica para relatório de controle de mudança com versionamento visando ao controle da qualidade

Controle de Versões			
Versão	Data	Autor	Notas da Revisão
1.0	27/05/2020	Sr. XXX (sponsor)	Elaboração Inicial – primeiro rascunho.
1.1	10/06/2020	Sr. XXX (sponsor)	Revisão pós-reunião com fornecedores e responsável pelo setor de compras.
2.0	31/03/2012	Sr. XXX (sponsor)	Revisão final.
2.1	31/03/2013	Sr. XXX (sponsor)	Inclusão de mais 2 indicadores IQ02, IQ03.
2.2	31/05/2015	Sr. XXX (sponsor)	Migrado para o plano de gerenciamento da qualidade.
2.3	29/06/2015	Sr. XXX (sponsor)	Revisão do programa de melhoria contínua.

Nº Solicitação	SM02	Solicitantes	Sr XXX (sponsor); Sr XXX (Responsável pelo setor).
Prioridade [0-Maior prioridade. 5- Menor]	3	Setor solicitante	Compras

#### **Descrição da mudança**

Na criação do relatório de solicitação de orçamento por período e fornecedores, incluir o status de resposta e atendimento adequado aos produtos adquiridos referente ao cumprimento de prazo, custo e qualidade.

#### **Justificativa**

Relatório no âmbito de maior complexidade de análise não previsto inicialmente. Sua necessidade surgiu a partir da implementação da gestão total de qualidade em processo de implantação na empresa. Após a inserção da funcionalidade de avaliação dos fornecedores, quanto ao cumprimento dos requisitos de atendimento as compras efetuadas.

#### **Classificação de impacto no projeto**

Análise de Impacto	Descrição
Esforço Estimado (Horas)	16 horas de um desenvolvedor; 4 horas de outro desenvolvedor para teste.
Custo Estimado (R\$)	R\$ 3.000,00.
Impacto no Prazo (Dias)	Mais três dias.

Requisito de Qualidade	Ações para atingimento	Indicadores
Tempo de resposta do relatório <= 30seg	Verificação do código de desenvolvimento; Sistema de gerenciamento de banco de dados adequado; Tempo de resposta da estrutura do servidor.	IQ01 – revisão em pares.
Entrega dentro do prazo	Comprometimento da equipe.	IQ02 – testes pela equipe de teste.
Criação da documentação	A equipe de desenvolvimento e testes deve ter fácil acesso aos documentos compartilhados.	IQ03 – auditoria.
Manutenibilidade	Código comentado. Documentos atualizados.	IQ04 – teste com os usuários.
Definição de pronto	Alinhar o documento que define quando a funcionalidade é considerada concluída.	

Aprovações		
Participante	Assinatura	Data
Patrocinador do projeto		
Gerente do projeto		

Fonte: adaptado do PMI (2017).

Existe uma definição de pronto (*Definition of Done*, no original, ou *DoD*) para a entrega de cada atividade que é tratada pelo método *Scrum* como uma criação desenvolvida pela equipe durante a *sprint* (período em que a equipe desenvolve uma parte do produto final), tendo que ser submetida a todas as etapas do desenvolvimento e validação, a fim de que possa ser dada como pronta (DONE). Esse processo de verificação reforça a transparência entre os integrantes da equipe, já que é o principal pilar da metodologia *Scrum*, tornando público o conhecimento do significado DONE. Para que

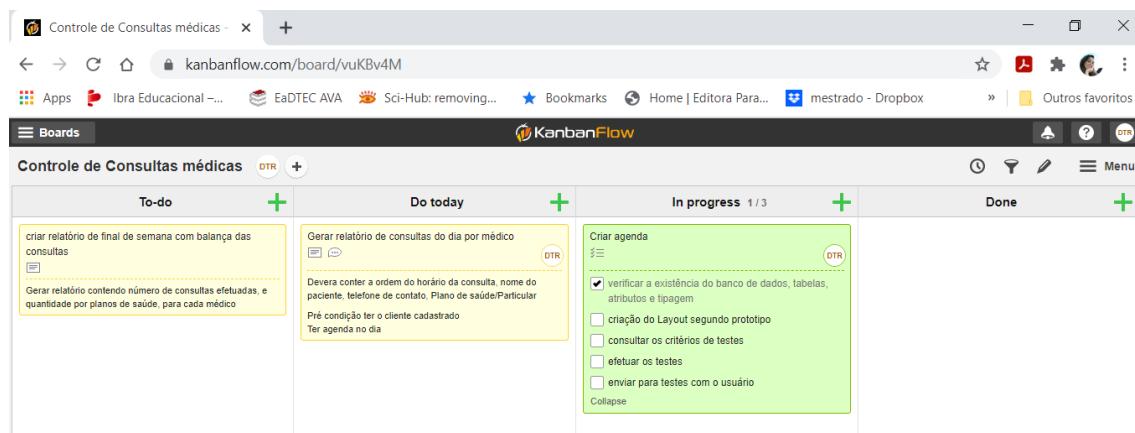
exista uma definição de pronto bem elaborada, é necessário o envolvimento do Time *Scrum* por completo, a fim de que sejam definidos os aspectos e a construção do produto, de testes e homologação de cada um dos itens do *backlog* da *sprint* (DUARTE, 2019).

Para ilustrar um exemplo de DoD, podemos ter: codificação, revisão do código, aprovação nos testes individuais, ser aprovado nos testes de funcionalidades manuais e/ou automatizados, versionamento e, por fim, a homologação que, em geral, dá-se pelo *Product Owner*, seu cliente no método *Scrum*.

Pensando nesse processo, é possível utilizar o método Kanban, com 3 colunas divididas em *TO DO* (fazer), *Work in Progress* (WIP – desenvolvendo) e *DONE* (pronto).

A fim de se ampliar o conceito de pronto para que haja uma boa definição, é preciso pensar em critérios de qualidade e incluir os requisitos não funcionais, como já discutido anteriormente na checklist proposta pela norma ISO 25010, no final do desenvolvimento é fundamental que se tenha passado por todas as análises e testes previamente definidos. Como resultado desse processo, a cada sprint trabalhada, haverá uma evolução, o que trará o aumento da qualidade no desenvolvimento do sistema, fazendo com que o empirismo inerente a cada ciclo gere a melhoria contínua, produzindo, assim, softwares com maior qualidade (DUARTE, 2019).

Figura 3.5 | Exemplo de quadro de KanBan durante o desenvolvimento de um sistema de controle administrativo de consultas médicas



Fonte: adaptada de Costa e Pereira (2019).

Para a utilização desse quadro, basta acessar o site e fazer o cadastro no sistema web. Um quadro pronto aparecerá para que você possa editá-lo, e já será apresentado um breve tutorial interativo de como funciona a ferramenta. Modifique o nome do quadro para um nome adequado com o projeto. Para cada tarefa incluída, poderão ser definidos os responsáveis, o tempo estimado e o tempo realizado, um detalhamento da tarefa, bem como inserir a descrição das tarefas, principalmente no que se refere ao processo de definição de pronto e métricas de qualidade.

Existem inúmeras outras ferramentas gratuitas para a utilização do quadro de Kanban, como o site Padlet.

Com todo esse conhecimento referente à qualidade no desenvolvimento de sistemas, incluindo o que já foi aprendido anteriormente sobre como mitigar riscos, ou seja, gestar todos esses quesitos, possuir conhecimento das técnicas de gestão de projeto, conhecer os cinco grupos de processo de gestão com normalizações reconhecidas nacionalmente e internacionalmente, saber escolher como será gestado o seu projeto mediante as características da equipe e da organização envolvida e tento algumas ferramentas para auxiliar nesses desafios do mundo de desenvolvimento de software, entendemos que, neste momento, você passa a ser um profissional capaz de resolver problemas e analisá-los cuidadosamente, planejar e organizar as atividades, bem como promover um relacionamento interpessoal saudável, com empatia e ética no trabalho, sabendo gerir os conflitos em todos os níveis e ser criativo e inovador em todos os processos, atendendo às necessidades de adaptação e, portanto, sendo flexível conforme a circunstância.

### FAÇA VALER A PENA

#### Questão 1

Para alcançar uma compatibilidade com a qualidade, é importante reconhecer a gestão da qualidade por meio de aspectos diferentes. Esses aspectos possuem pontos de vista distintos, a depender diretamente do atendimento ou não às expectativas do *stakeholder*.

- I. A Satisfação do cliente está no atendimento de suas expectativas.
- II. A gestão da empresa possui sua responsabilidade quanto a garantir a qualidade.
- III. A qualidade do produto está desvinculada dos interesses organizacionais como o custo, o prazo e o escopo.

Considerando o contexto apresentado, assinale a alternativa correta.

- a. Apenas as afirmativas I e II estão corretas.
- b. Apenas a afirmativa I está correta.
- c. Apenas a afirmativa II está correta.
- d. Apenas as afirmativas I e III estão corretas.
- e. As afirmativas I, II e III estão corretas.

#### Questão 2

Uma das grandes dificuldades da gestão da qualidade de desenvolvimento de software são os métodos de medição de qualidade, afinal, o software não sofre desgaste. A ISO/IEC 25010, de 2011, propõe um modelo que define oito categorias de características sobre a qualidade, sendo elas divididas em subcaracterísticas.

Característica	Exemplo
I. Adequação funcional	a. O sistema compartilha informações com os demais sistemas ao compará-las.
II. Eficiência	b. O funcionamento se dá de acordo com o que foi estabelecido por um lado.
III. Compatibilidade	c. Facilidade no processo de correções e manutenção do sistema.
IV. Usabilidade	d. O desempenho requerido pelo produto é atendido no que diz respeito ao usuário.

Característica	Exemplo
V. Confiabilidade	e. Cada usuário tem o seu nível de acesso definido.
VI. Segurança	f. Com pouca adaptação, é possível utilizar em SO diversos.
VII. Manutenibilidade	g. O sistema atende a todas as necessidades implícitas do software.
VIII. Portabilidade	h. Dentro de um contexto de uso, o software possui eficiência, eficácia e s

Assinale a alternativa que apresenta a associação CORRETA entre as colunas.

- a. I-a; II-h; III-g; IV-d; V-e; VI-c; VII-f; VIII-b.
- b. I-d; II-a; III-h; IV-b; V-e; VI-c; VII-f; VIII-a.
- c. I-f; II-g; III-d; IV-a; V-h; VI-b; VII-e; VIII-c.
- d. I-g; II-d; III-a; IV-h; V-b; VI-e; VII-c; VIII-f.
- e. I-c; II-f; III-g; IV-d; V-a; VI-h; VII-b; VIII-e.

### Questão 3

Em um projeto de desenvolvimento de sistema, visando à qualidade do sistema a ser desenvolvido, utilizando-se da metodologia Scrum, é relevante a existência de critérios de avaliação de cada uma das entregas a ser efetivada. Porém, para que essa tarefa seja considerada concluída, a equipe precisa ter a compreensão do significado de pronto (DoD). Esse processo é relevante para que não exista dúvidas sobre quando o desenvolvedor pode dizer “ACABEI!”

Tomando como referência o contexto apresentado acima, julgue cada uma das afirmativas a seguir como (V) verdadeira ou (F) falsa.

- ( ) A equipe Scrum cria os critérios de DoD para cada *sprint* no início da *sprint*.
- ( ) O processo de DoD reforça a transparência entre os integrantes, o principal pilar do *Scrum*.
- ( ) O envolvimento de toda a equipe para a definição dos aspectos de construção, testes e homologação torna a definição de pronto melhor elaborada.

Assinale a alternativa que apresenta a sequência CORRETA.

- a. V – F – V.
- b. F – V – V.
- c. V – V – F.
- d. V – F – V.
- e. F – F – V.

#### *REFERÊNCIAS*

ABNT — Associação Brasileira de Normas Técnicas. **NBR ISO/IEC 9126-1. Engenharia de software - Qualidade de produto - Parte 1: Modelo de qualidade.** Rio de Janeiro: ABNT, 2003.

CARVALHO, F. C. A. (org.). **Gestão de projetos.** São Paulo: Pearson Education do Brasil, 2015.

CARVALHO, F. C. A de. **Gestão de projetos.** 2. ed. Pearson Education do Brasil, 2018.

COSTA, A. B. da; PEREIRA, F. da S. **Fundamentos de gestão de projetos:** da teoria à prática – como gerenciar projetos de sucesso. Curitiba: Intersabers, 2019.

DUARTE, L. F. **Agile coaching:** um guia prático. 2. ed. [S.l.]: eBook Kindle, 2019.

ISO — International Organization for Standardization. **ISO/IEC 25010. System and Software engineering - System and software Quality Requirements and Evaluation (SQuaRE) - system and software quality models.** Geneva: ISO, 2011.

JALOTE, P. **An integrated approach to software engineering.** 3. ed. New York: Springer, 2005.

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

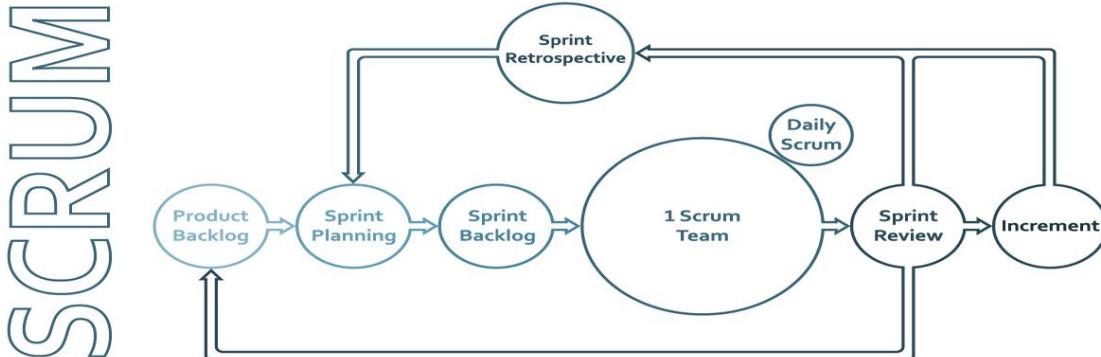
PRESSMAN, R. S. **Engenharia de software.** 6. ed. Rio de Janeiro: McGraw-Hill, 2006. 720p.

ROCHA, A. R. C.; MALDONADO, J. C.; WEBER, K. C. **Qualidade de software: teoria e prática.** São Paulo: Prentice Hall, 2001. 303p.

SOMMERVILLE, I. **Engenharia de software.** 10. ed. São Paulo: Pearson Prentice Hall, 2018.

WEBER, K. C. Tecnologia da informação: programa brasileiro da qualidade e produtividade em software. [S.l.: s.n.], 2008. 485 p.

# FOCO NO MERCADO DE TRABALHO



Fonte: Shutterstock.

**Deseja ouvir este material?**

## SEM MEDO DE ERRAR

Para que se consiga desenvolver o sistema de visualização e manipulação de radiografias obtidas por meio de raio-X, tomógrafo e ultrassom, utilizando-se do recurso de identificação de gestos e/ou voz, faz-se necessário definir qual será a tecnologia utilizada, a fim de se verificar na equipe um profissional com conhecimentos do uso das bibliotecas do equipamento a ser adquirido. Para esse item, é preciso analisar custo x benefício, além das limitações, o tempo de resposta e as questões de definição da captura dos movimentos ou da interpretação da voz, lembrando que parte das informações se encontra no termo de abertura e na documentação referente ao planejamento das atividades.

Devem ser incluídas nesse processo de planejamento a análise dos riscos e as respostas dadas a eles. Além disso, a definição da qualidade pode ser definida antes de se iniciar a próxima *sprint*, minimizando a perda de tempo no processo de planejamento, uma vez que poderá haver inúmeras alterações nas atividades anteriores.

Após o processo de definição do equipamento e análise de cada requisito funcional e não funcional, é possível dar início ao desenvolvimento, como:

- RF1 – cadastro do paciente.
- RF2 – cadastro do procedimento.
- RF3 – inserção dos arquivos referentes às radiografias do exame.
- RF4 – criação da área de acesso e movimentação das radiografias com interação do uso de gestos e voz.
- RNF1 – garantir a segurança das informações apenas aos funcionários pertinentes.
- RNF2 – o sistema terá eficiência e eficácia na manipulação das imagens.

Figura 3.6 | Kanban DoD – definição de pronto

The image shows a digital Kanban board titled "Kanban DoD - Definição de pronto". The board has three columns: "TO DO - À fazer", "Work in Progress (WIP)", and "DONE (pronto)".

**TO DO - À fazer:**

- Inserção dos arquivos das radiografias.  
RNF3.1 – garantir que os arquivos inseridos estão no formato de leitura adequado ao sistema;
- RNF3.2 – em caso de tomografia permitir a montagem da imagem 3D com definição.
- RNF3.2 – Nas imagens 3D verificar a eficiência de resposta do processamento da imagem e dos dados durante o procedimento.
- Critérios de qualidade:  
revisão do código individual;  
testes individuais incluindo os RNF;  
testes de funcionalidades;  
teste de sobrecarga;  
versionamento por GitHub;  
homologado pelo Product Owner;
- Adicionar comentário

**Work in Progress (WIP):**

- Cadastro do procedimento  
RNF2.1 – Agilidade na inserção dos dados do procedimento por meio do CID.
- RNF2.2 – Inserção automática dos materiais cirúrgicos a serem utilizados definidos previamente por meio do CID.
- Critérios de qualidade:  
revisão do código individual; ok  
testes individuais incluindo os RNF; ok  
testes de funcionalidades; ok  
teste de sobrecarga; ok  
versionamento por GitHub; ok  
homologado pelo Product Owner; ok
- Passou pelos testes de adequação funcional; eficiência, compatibilidade; confiabilidade; Segurança; Manutenibilidade; foi dispensado da portabilidade.
- PENDÊNCIA**  
usabilidade É NECESSÁRIO  
APERFEIÇOAMENTO NO QUE DIZ  
RESPEITO A DISTRIBUIÇÃO DAS  
FUNCIONALIDADES DE  
MOVIMENTAÇÃO POR GESTOS;
- Adicionar comentário

**DONE (pronto):**

- Cadastro do paciente  
RNF1.1 – validação das informações como data de nascimento, cpf, etc.
- RNF1.2 – agilidade no processo de cadastro com biblioteca de pesquisa de CEP, fazendo auto preenchimento dos dados de residência.
- Critérios de qualidade:  
revisão do código individual; ok  
testes individuais incluindo os RNF; ok  
testes de funcionalidades; ok  
teste de sobrecarga; ok  
versionamento por GitHub; ok  
homologado pelo Product Owner; ok
- Passou pelos testes de adequação funcional; eficiência, compatibilidade; usabilidade; confiabilidade; Segurança; Manutenibilidade; foi dispensado da portabilidade.
- Adicionar comentário

Fonte: captura de tela do Padlet elaborada pela autora.

Note que aqui foram tratadas as funcionalidades triviais e ainda é necessário um grande refinamento dos quesitos referentes aos critérios de aprovação, não apenas da funcionalidade, mas também das 8 características de análise de qualidade de um software proposto pela ISO 25010, podendo ser agregadas as análises de inspeções, PDCA, auditoria e utilização de sistemas para gestão da qualidade.

Acreditamos que você tenha realmente entendido não apenas como utilizar as ferramentas, mas, principalmente, a importância desse processo dentro da profissão de desenvolvedor de sistemas. Não estamos falando exclusivamente de criar qualidade para o produto, mas levar qualidade aos profissionais que utilizarão o sistema e beneficiar, por consequência, seus clientes.

## ***NÃO PODE FALTAR***

### ***DOCUMENTAÇÃO DO PROJETO***

*Daniela Teresa Rossignoli Uebele*



Fonte: Shutterstock.

**Deseja ouvir este material?**

#### ***PRATICAR PARA APRENDER***

Caro aluno, você já deve ser usuário de inúmeros sistemas de software, não é? Mas quando foi que você procurou o manual para aprender a usar um sistema? E o quanto o manual ou o help realmente lhe ajudou?

É dentro dessa realidade que vale a pena refletir não somente sobre o que documentar, mas como e quando fazer isso, afinal, investir o seu tempo em gerar documentos que não serão utilizados com certeza não faz parte dos seus planos. Para aqueles que amam o processo de desenvolvimento de sistema, documentar costuma ser um martírio, no entanto, faz-se necessário a mudança de pensamento e principalmente de hábito, a fim de que o processo de documentar se torne útil e consequentemente agradável.

Para isso, temos, hoje, o recurso tecnológico a nosso favor, não é verdade? Podemos documentar e acessar as documentações em tempo real e compartilhá-las, o que traz a você, futuro desenvolvedor de uma equipe criadora de software, grandes facilidades, no entanto, agora, é preciso entender os motivos que tornam a documentação um elemento

que define e melhora a qualidade do seu trabalho e de sua equipe, perceber o quanto a sua falta ou a má documentação gera grandes riscos de se ter projetos inadequados ou mesmo projetos encerrados sem a sua conclusão.

Entender que a documentação é um guia não apenas para a equipe desenvolvedora, mas para a equipe gestora, de qualidade e, principalmente, para o cliente, que vê nesse processo profissionais preparados para planejar e organizar suas atividades, promove um relacionamento interpessoal favorável, apresentando flexibilidade e adaptação ao processo de mitigar os riscos que venham a comprometer o sucesso do projeto de software.

A documentação traz garantias a todos os envolvidos no projeto; delimita tudo o que será executado e gerenciado, deixa claro tudo o que não está incluído e evita atrasos no cronograma. Além disso, levando-se em consideração a necessidade da mudança, esse documento também é gerido pela equipe que busca meios criativos e inovadores de transpor desafios maiores, de levar ao cliente o que há de mais atual em ferramentas, técnicas de desenvolvimento e gestão de projetos, promovendo um produto dentro do escopo, do custo, do tempo e da qualidade previstos, bem como funcional às necessidades de sua empresa, alinhado com todos os setores, visando a um atendimento de qualidade aos clientes que a organização atende.

Considere que sua equipe será responsável pelo projeto de desenvolvimento do sistema de uma ferramenta CASE que utilizará recursos tecnológicos de identificação de gestos e/ou voz com realidade aumentada, levando a equipe a efetuar uma análise aprofundada, a fim de propor a resolução do desenvolvimento da ferramenta CASE. Os médicos poderão utilizar esses recursos durante o procedimento cirúrgico para analisar tomografias, ultrassonografias e raio-x digitais, minimizando o tempo do procedimento cirúrgico, uma vez que, por meio desse recurso de gestos e/ou voz, o profissional da medicina não terá a necessidade de remover as luvas cirúrgicas para manipular a tela do computador ou as chapas físicas, poupano a constante higienização das mãos para voltar ao procedimento. Além disso, o paciente ficará menos tempo anestesiado e o centro cirúrgico ficará disponível para mais procedimentos.

Considerando a importância de se documentar todas as funcionalidades do sistema a ser desenvolvido e visando apresentar para os profissionais que atuarão por meio sistema que o hospital pretende oferecer. A criação de um Termo de Abertura do Projeto, a padronização de um documento de solicitação de mudanças, a definição de como será administrado o cumprimento das atividades por meio de um software colaborativo, o local que será o repositório desses documentos, o software para as documentações de alto nível e os arquivos comuns para o desenvolvimento do sistema, que se mostram propícios para que esse projeto tenha o seu início formalizado, demonstrando planejamento e organização da equipe, a flexibilidade para mudanças e adaptação aos riscos, além de apresentar soluções criativas e inovadoras.

Para tanto, cabe a utilização de todas as análises de risco e do planejamento do monitoramento já realizado como os requisitos previamente sugeridos, as definições de pronto e as ferramentas definidas para a gestão do andamento do projeto, visando-se à busca da qualidade do produto a ser desenvolvido, demonstrando uma postura ética e empatia no trabalho.

Utilizar os conhecimentos aprendidos para se desenvolver o TAP com detalhamento de sua importância do início ao fim do projeto, sendo adequadamente documentado e atualizado, evitando-se o desperdício do tempo em documentações excessivas, porém sem deixar de fora as informações que trará as devidas garantias para o sucesso do projeto e a segurança de que toda a equipe está alinhada à proposta de implantação de um novo recurso com toda segurança necessária.

Vale lembrar a importância de se utilizar a documentação do código, visando a manutenções futuras ou novas atualizações, podendo manter o produto do projeto ativo por muitos anos, adequando-o conforme a evolução tecnológica e os sistemas operacionais e hardware.

Com certeza você e sua equipe trarão propostas enriquecedoras a esse projeto, buscando o que existe de mais atual e moderno no mercado tecnológico na área da saúde.

Busque, nesta seção, muito mais do que processos ou documentos padronizados e repletos de informações; inclua valor nessas informações, dê significado ao processo documental, entenda esse recurso como um benefício para o desenvolvimento de sistemas.

#### **CONCEITO-CHAVE**

#### **RISCOS PELA AUSÊNCIA DE DOCUMENTAÇÃO**

A utilização de documentos pessoais no dia a dia é essencial, não importa aonde vamos, é sempre necessário um documento de identificação (RG), de habilitação (CNH) e o passaporte, caso necessite sair do país. Em praticamente tudo que fazemos nos é exigido uma identificação formalizada, e além desses, necessitamos de documentos que indiquem a nossa trajetória, a nossa história, que comprovem a existência, a qualidade e minimizem os riscos de falsificações e fraldes, por isso, esses documentos são emitidos e verificados por órgãos responsáveis, que se utilizam de tecnologia, softwares e métodos para garantir a veracidade e a segurança das informações. Analise a Universidade em que estuda. Ela é reconhecida e certificada pelo MEC e sempre busca comprovar sua excelência nos inúmeros cursos ofertados, bem como capacitando funcionários e/ou selecionando-os a partir de processos de contratação.

O grau de exigência do mercado vem se elevando a cada dia, por isso, discute-se, há tempos, a maturidade no gerenciamento dos projetos dentro das empresas de consultoria em TI, já que, de forma geral, o triângulo de ferro de um projeto vem sendo comprometido frequentemente, afetando, diretamente, a qualidade do produto e do projeto (CARVALHO, 2018).

Esse fato se dá por inúmeros motivos; um dos grandes riscos é a ausência ou mesmo a má documentação, que traz complicações ao processo de implementação, tornando complexo o trabalho de um desenvolvedor quanto à inexistência ou mesmo às poucas informações sobre o que se deve executar, podendo levar à criação de funcionalidades inadequadas em relação às necessidades do cliente, havendo, por consequência, o desperdício de tempo, devido ao retrabalho ou ao fato das informações estarem restritas

e/ou concentradas no código fonte desenvolvido, com acesso exclusivo ao desenvolvedor, levando a equipe a refazer códigos que poderiam ser reaproveitados (SOMMERVILLE, 2018).

Podemos observar, a partir dessas situações, a importância da formalização de alguns processos, a ausência de documentos que definam, de forma clara e objetiva, os requisitos previamente analisados, seus riscos quantitativos e qualitativos, além de como controlar ou combater a forma e o meio de comunicação e a definição de pronto (DoD), não se limitando a apenas esses itens. Tratam-se de documentações de real valor, que devem sim ser mínimas, objetivas, claras e de fácil acesso a toda equipe, em especial ao gerente do projeto, pois, por meio dessa documentação de planejamento, ocorre a execução e o controle, fazendo com que a equipe busque atender à entrega do escopo, custo e tempo adequados com a devida qualidade (COSTA e PEREIRA, 2019).

Cabe ressaltarmos a problemática existente pela ausência ou má documentação relacionada à gestão do prazo, caso não exista nenhum documento que defina os prazos para a conclusão de cada uma das tarefas, tornando-se complicados o comprometimento e o controle do desenvolvimento do sistema, desencadeando inúmeros outros problemas: a equipe provavelmente passará a viver um estresse psicológico devido à pressão, além do desgaste físico, precisando, talvez, fazer hora extra para compensar a falta da gestão do tempo, havendo, ainda, a redução da produtividade, a queda da qualidade e o aumento nos custos do projeto, além de inúmeros outros agravantes em todas as áreas de conhecimento envolvidas no processo de gestão de projetos em desenvolvimento de software (NEWTON, 2011).

Existe outro quesito muito importante, que faz parte da realidade da maioria das pessoas, principalmente das que vivem no mundo da tecnologia, que diz respeito à quantidade de informações que recebemos, que, por ser tão grande, a chance de esquecermos algo importante é eminente. Portanto, uma das grandes diferenças entre os desenvolvedores profissionais e amadores está na questão da criação de documentações.

Dentro de todo o nosso estudo, não houve a imposição de se criar um determinado documento desse ou daquele jeito, afinal, não existe uma regra específica para a documentação de um software; o aperfeiçoamento dessas documentações ocorrem em um processo de melhoria continua, conforme a colaboração e produtividade da equipe envolvida no desenvolvimento do software. Para que esse processo ocorra, existem inúmeras ferramentas de desenvolvimento e de documentação que permitem a criação desses arquivos de forma simultânea e coordenada, assim, todos da equipe têm acesso às informações em tempo integral. Essas ferramentas de integração, conhecidas também como repositório, geram o versionamento do sistema em desenvolvimento, além de um excelente controle de alterações.

Portanto, se o programa será para uso particular ou mesmo para algum projeto de uso pontual (muito raro de ocorrer), não haverá a necessidade de inúmeras documentações, mas, ainda assim, será de grande valia comentar o código e gerar versões diferentes. No entanto, quando o sistema for para outras pessoas utilizarem, podendo, inclusive, haver manutenção por outros desenvolvedores, será necessária a existência de informações extras, além do código (SOMMERVILLE, 2018).

A documentação se torna ainda mais relevante para manter as atividades da equipe alinhadas aos objetivos da organização. Portanto, o estudo sobre a questão da

governança e da estratégia das organizações utilizando-se ITIL e COBIT para o processo de gestão dos recursos da área de TI em relação ao alinhamento com todos os outros setores da empresa cita também a necessidade de haver um processo ágil, ainda sugerindo a criação de documentos de formalização.

#### ASSIMILE

Primeira parte de uma sugestão de termo de abertura de projeto: observe no texto a alinhamento estratégico com ITIL quando cita a necessidade do sistema que visa a melhorar e agilizar o agendamento pelos pacientes com maior flexibilidade quanto ao horário para o agendamento da consulta e visualização da disponibilidade dos horários na agenda do profissional.

Quadro 3.12 | Agendamento

Termo de Abertura do Projeto				LOGO
Nome do Projeto – tudo que estive grifado substitui				AUTOMATIZANDO AGENDAMENTO DE CONSULTAS
Controle de Versões				
Versão	Data	Autor	Notas da Revisão	
1.0	01/12/2020	Izabel Santos	Daniela Uebele	
1.1	07/12/2020	Daniela Uebele	Izabel Santos	

- Objetivos do documento (Descreva quando e como esse documento será utilizado)

Oficializar o início do projeto, definir os responsáveis, as entregas essenciais, documentar requisitos iniciais, premissas e restrições.

- Cenário atual e a justificativa do projeto (Breve histórico, realidade atual. Descreva a atual situação e o motivo do projeto.)

A secretaria precisa estar sempre à disposição para atender aos pacientes por telefone, o que a impede de fazer outras atividades externas. Logo, o objetivo do projeto é o autoagendamento dos pacientes, a fim de que a secretaria tenha disponibilidade para os afazeres do consultório. Alinhando esse processo de agendamento a um atendimento mais eficiente ao paciente, oferecendo uma melhor visualização de sua agenda pessoal com a do consultório, com agendamento 24h ou, ainda, o agendamento tradicional via telefone nos horários de atendimento.

O sucesso do projeto será alcançado ao atender a todos os critérios de aceitação das entregas, às restrições e ao cronograma.

- Sistema Web de agendamento, pacientes visualizam os horários de consultas livres.
- Efetuar o autoagendamento com breve cadastro.
- Confirmação automática 24h antes da consulta automatizada por mensagem.
- Equipe de desenvolvimento web, DBA, sistema de segurança e privacidade.
- Sistema concluído em 60 dias úteis.

(PMI, 2017).

Considerando esses fatos, diga-me, então, por que não documentar um projeto de desenvolvimento de sistemas? Quanto custa um sistema bem documentado? Quanto você valoriza documentar e organizar o seu trabalho? Se você for continuar o trabalho de alguém, imagina como seria mais simples a compreensão de tudo que já foi feito, discutido, pensado, o que deu certo e o que já deu errado!. A economia de tempo é um ponto relevante para criarmos o hábito de documentar (SOMMERVILLE, 2018).

É interessante olhar esse cenário relacionado às documentações. Em geral, a utilização das metodologias ágeis não enfatiza o ato de documentar os processos, porém não o descarta. No entanto, documentar pode ser o **diferencial** para se aumentar a qualidade e a produtividade da equipe de trabalho. Conhecer as metodologias e as técnicas de gestão de projetos utilizando-se de normatizações é um caminho bem promissor.

Na gestão ágil, falamos de menos documentação, porém não da extinção dela. Ao discutirmos as dez áreas do conhecimento propostas por PMBOK, metodologia ainda fundamentada na gestão de projeto de empresas mais tradicionais, atualizada constantemente conforme a evolução do mercado e já adequada para coexistir com outras metodologias, em todas as áreas observamos a importância da criação de documentos; aliás, muitos alimentam o próximo grupo de processo do ciclo de vida do desenvolvimento de software (COSTA e PEREIRA, 2019).

#### REFLITA

Voltando a alguns assuntos já trabalhados, a gestão ágil veio como uma alternativa à gestão tradicional, mas nunca foi dito que se descartaria todo o aprendizado anterior! O objetivo maior dessa metodologia que surgiu no início do século XX, a partir do sistema de produção Toyota, tem como foco eliminar desperdícios de toda natureza e fazer entregas rápidas, contínuas e com qualidade total (TQM – total *quality management*) (SHIMOKAWA; FUJIMOTO, 2009). Baseou-se no fato de que imprevistos, problemas e riscos são inerentes a todo e qualquer projeto, independentemente de suas proporções de complexidade e tamanho, certo? O que importa mesmo é absorver as inevitáveis e constantes mudanças? Fazer as integrais de forma constante, permitindo que o seu cliente participe do processo? Que o produto a ser entregue seja realmente útil para o que foi proposto? A documentação faz parte de todo esse processo, a qualidade e os riscos estão na falta da documentação i? Muitos questionamentos, não é? Vale a busca por mais experiências (COSTA; PEREIRA, 2019).

#### QUANDO DOCUMENTAR

Um bom exemplo de documento simples, “rápida” de preencher é o TAP (termo de abertura do projeto), este documento ele é o 1º a ser gerado e percorre todas as etapas do projeto, sendo consultado e alterado do início ao final. É gerado no grupo de processo inicial do projeto, momento em que firma o comprometimento entre as organizações para o desenvolvimento (START do projeto), que traz informações orientadoras para todo o projeto, com o escopo pré definido, que se documentado errado com certeza trará sérios problema de qualidade (CARVALHO, 2018).

Segundo Newton (2011), criar documentos padronizados, a exemplo o “termo de abertura” sugerido por PMI ou usar um “documento de iniciação de projeto”. É um

raciocínio que embasa a criação destes documentos, sendo este o ponto importante, e não os templates prontos que trazem apenas lacunas a serem preenchidas (Figura 3.7).

Com certeza, essas sugestões foram apontadas devido à experiência de inúmeros gestores, em infinitos projetos que deram certo, porém nunca se limitaram. Como o próprio PMI cita em inúmeros momentos do Guia PMBOK, os modelos podem variar conforme cada organização, cada projeto e cada equipe (NEWTON, 2011; PMI, 2017).

Observe que o termo de abertura possui versões, pois, ao passar para o processo de planejamento, sofrerá um alinhamento com questões não pensadas logo no início do projeto.

Para tanto, são necessários outros documentos, como a EAP e o seu Dicionário, o cronograma com as interdependências e as datas marcos. Os requisitos funcionais deverão ser detalhados, e, em geral, utiliza-se a metodologia UML para se criar essa modelagem inicial. Observe estes itens listados em uma versão inicial e apenas os principais itens constarão no termo de abertura.

Figura 3.7 | Segunda parte do termo de abertura do projeto mostrando a análise por EAP

<p>✓ Estrutura Analítica do Projeto – Fases e principais entregas (Principais entregas, versão inicial.)</p> <p>1 Análise</p> <p>1.1 Prototipagem – esboço das telas com a definição de navegação e campos necessários para o agendamento.</p> <p>1.2 Documentação – TAP; cronograma; análise de risco; planejamento de comunicação; análise do sistema UML.</p> <p>2 Desenvolvimento</p> <p>2.1 Layout – criação dos layouts de tela com configuração de identidade visual.</p> <p>2.2 Banco de dados - criação do CRUD do sistema e testes com pesquisas e relatórios.</p> <p>2.3 Sistema de segurança - controle de log dos usuários da clínica.</p> <p>3 Gestão de recursos – definir e controlar os recursos necessários para o desenvolvimento do sistema.</p>
<p>✓ Principais requisitos das principais entregas/produtos (Principais requisitos dos produtos/entregas a serem atendidos identificados na EAP acima.)</p> <p>1 Visualização e agendamento, conforme a liberação de horário pelo médico;</p> <p>2 Confirmação da consulta automática um dia antes;</p> <p>3 Criar relatório da agenda do dia;</p> <p>4 Gerar relatório no final da semana com o resumo dos atendimentos realizados por médico e plano de saúde.</p>

Fonte: adaptada de PMI (2017).

Além desses itens, temos a utilização de algumas ferramentas disponíveis para análise de risco e definição das ações para os classificados como prioritários; para a definição do planejamento de qualidade, decidindo-se como serão feitas as avaliações da qualidade e quais os métodos e as ferramentas a serem utilizadas. Portanto é necessário definir como serão feitos os testes no sistema, e após todo o processo de planejamento e geração das documentações pertinentes, vale ressaltar que a utilização desses

documentos e das ferramentas para esses processos será definida pela equipe conforme a necessidade de cada projeto e as características da organização (CARVALHO, 2018).

Cabe voltar às sessões anteriores e verificar o quanto falamos de analisar os riscos visando a mitigar, eliminar e controlar os riscos por meio de documentações simples e objetivas para não se perder o objetivo de acompanhar cada um dos riscos elencados a partir do estudo do escopo do desenvolvimento do sistema, que ocorre de forma detalhada no **grupo de processo de planejamento**. É nesse processo que ocorre o plano de gerenciamento, e os documentos do projeto são gerados para se ter o planejamento dos riscos, da qualidade, do tempo, do custo, das comunicações e dos recursos necessários, podendo haver um replanejamento conforme ocorrerem as mudanças (CARVALHO, 2018).

Para que esse processo de mudanças ocorra de forma “tranquila” e controlada, é bastante interessante a utilização de um documento para controle de mudanças, assim, é possível haver um log das solicitações, ou seja, um controle de tudo que for solicitado e a situação em que as alterações se encontram com o status de: Ok, cancelada, pendente ou em andamento, é fundamental que na análise da solicitação se classifique a prioridade de 0 a 4 sendo respectivamente altíssima e muito baixa a prioridade, definidos a partir da análise do impacto que a alteração trará para o desenvolvimento do sistema em relação a tempo, custo, escopo e qualidade. Cabe definir se a mudança será corretiva, preventiva, de reparo de Defeitos, fora do escopo ou apenas informativa (MONTES, 2017).

Figura 3.8 | Exemplo de solicitação de mudanças

**DOCUMENTO DE SOLICITAÇÃO E CONTROLE DE MUDANÇAS**

Controle de Versões			
Versão	Data	Autor	Notas da Revisão
1.0	10/12/2020	Izabel Santos	Elaboração Inicial – primeiro rascunho
1.1	15/12/2020	Daniela Uebele	Revisão pós reunião com a funcinária

Solicitante	Izabel Santos	Prioridade [0-Maior ↔ 4-Menor]	2
Status	Pendente	Classificação	Fora do escopo

**✓ Descrição sumária**

[Descreva os requisitos e as características dos produtos com as mudanças que devem ser entregues.]

Inserção de um relatório semestral por plano de saúde contendo levantamento de atendimentos, procedimentos efetuados e valor recebido.

**✓ Justificativa**

[Porque a necessidade da mudança.]

Verificação do número de pacientes, os valores que estão sendo oferecidos por cada plano de saúde, e se a permanência no plano de saúde é pertinente.

**✓ Classificação de impacto no projeto**

[Preenchido pelo solicitante ou GP conforme definido no plano de gerenciamento de projetos.]

Análise de Impacto	Descrição
Esforço Estimado (Horas)	40 horas de desenvolvimento.
Custo Estimado (R\$)	R\$10.500,00
Impacto no Prazo (Dias)	Mais uma semana de desenvolvimento

Aprovações		
Participante	Assinatura	Data
Patrocinador do Projeto	Constantino Souza	16/12/2020
Gerente do Projeto	Daniela Uebele	16/02/2020

Fonte: adaptada de PMI (2017).

Lembre-se de que, por mais que se analise e busque levantar todos os requisitos no início do projeto, não será possível saber com precisão a realidade do cliente, vivemos em um mundo em constantes mudanças, inclusive políticas e sociais; somos seres imperfeitos, e sua função como gestor de projeto é buscar a entrega de um software aceitável pelo cliente, portanto, faça perguntas mesmo que pareçam simples, pois são esses questionamentos que trarão um conhecimento aprofundado para possibilidades impensadas.

Todos os processos do planejamento que geram documentações visam a uma forma de apoio para tomada de decisões acertadas. Verificando-se a qualidade e as modificações que o projeto venha a sofrer, criam-se parâmetros para que a equipe e os sponsor consigam determinar se os requisitos do sistema foram atendidos, sendo esse um momento oportuno para se utilizar a documentação no grupo de processo de encerramento, pois é por meio da documentação inicial que auxiliou o planejamento que é possível obter a definição de pronto para os membros da equipe, a aceitação do

cliente, revisando o final de cada fase e do projeto total, ter as lições aprendidas documentadas é um meio de criar um histórico para os projetos futuros (CARVALHO, 2018; COSTA; PEREIRA, 2019; SOMMERVILLE, 2018).

Observe na Figura 3.9 as restrições, as premissas e os riscos iniciais analisados. O exemplo está bem simples, porém é suficiente para entender que o cliente tem que ter ciência do que foi acordado e que ele possui responsabilidade também pelo sucesso do projeto. As premissas deixam claras as questões de comunicação e de infraestrutura necessárias, bem como os gastos para se manter o sistema em funcionamento.

Figura 3.9 | Parte final do termo de abertura contendo as restrições do projeto, as premissas, os riscos e o orçamento final

- ✓ Restrições (limitação aplicável ao projeto, que afetará o seu desempenho. Limitações reais: orçamento, recursos, tempo de alocação)

Valor do projeto de R\$ 5.000,00 (cinco mil reais)

Servidor do sistema, página web, segurança, banco de dados, sistema desenvolvido, toda a responsabilidade da equipe.

Atendimento à questão de tempo de resposta do sistema, disponibilidade para os usuários, manutenção dentro do tempo acordado com o cliente.

- ✓ Premissas (considera-se como fatos verdadeiros sem prova para fins de planejamento.)

Os usuários terão disponibilidade para atender à equipe e fazer testes 50% do tempo do projeto.

Já possui infraestrutura e conhecimento da utilização do sistema solicitado.

A comunicação se dará por meio de e-mail ou sistema de conversa instantânea e reuniões, e se houver necessidade de alterações, com documentação pertinente.

- ✓ Riscos (Descreva os principais riscos do projeto.)

Mudança no cenário financeiro inicial.

Usuários perderem o interesse no sistema.

Surgimento de sistema comercial equivalente.

- ✓ Orçamento do Projeto (Estimativa preliminar dos custos do projeto representada pelo orçamento ou pelo fluxo de caixa com suas principais entradas e saídas financeiras. Base para a aprovação financeira do projeto e da formação da linha de base dos custos)

#### Aprovações

Participante	Assinatura	Data
Patrocinador do Projeto	Consultório ??????????	
Gerente do Projeto	Daniela Teresa Rossignoli Uebel	14/12/2020

Início do Projeto: xx/xx/20xx

Fim Previsto do Projeto: xx/xx/20xx

Fonte: adaptada de PMI (2017).

## **POR QUE DOCUMENTAR?**

Note que todos esses autores e muitos outros concordam com a importância de se criar documentações mínimas, porém necessárias para que se tenha um desenvolvimento de sistema adequado à necessidade. Perceba que a integração de todas as áreas que o PMBOK cita é um meio de se buscar o conhecimento e a compreensão da importância de todas as etapas do projeto. Assim, ao desenvolver um sistema, a gestão do risco e da qualidade, na criação de um software, não está apenas relacionada à criação de um programa ou um conjunto de programas, faz parte do software a documentação necessária para que o usuário do sistema, a equipe que analisará a qualidade e os desenvolvedores possam ter parâmetros para o desenvolvimento de um sistema de qualidade.

Gerando a partir das documentações técnicas e do projeto a manutenibilidade, normalmente armazenada em meio eletrônico, a dependabilidade e a segurança da informação, a eficiência e a aceitabilidade de um sistema concluído, atendendo a todos os requisitos que foram solicitados no início do projeto e detalhados no planejamento, executados e controlados para que, ao fim, houvesse meios de aprovar um sistema adequado às necessidades e ao propósito inicial do cliente (SOMMERVILLE, 2018).

Descrevemos a abordagem tradicional da importância da documentação, em que é comum que a definição aconteça no início, ou seja, antes do desenvolvimento, trazendo consigo as especificações dos requisitos, os casos de uso detalhados, os modelos de projetos, entre inúmeros outros elementos oferecidos para a modelagem de um sistema. O que deve ser ressaltado é que esses documentos não devem ser apenas mantidos, mas principalmente atualizados até o fim do desenvolvimento do sistema, momento em que é entregue o produto ou serviço final.

É importante a criação de documentos enquanto ainda se define o que deve realmente ser criado, sendo, provavelmente, o pior momento do trabalho, já que existe uma incerteza quanto ao que se deseja realmente. Nesse momento é preciso refletir sobre o custo do processo. No início, ele teve um grande valor e o custo era viável, porém deve-se analisar o custo para as atualizações (CARVALHO, 2018).

Considerando que na metodologia ágil primeiro se comunica e depois se realiza, deixamos para gerar a documentação em momentos adequados, podendo ser durante ou após o desenvolvimento das funcionalidades do sistema, considerando que a compreensão e o desenvolvimento se encontram em um processo estável, o que reduzi o custo para se manter a documentação atualizada.

## **ASSIMILE**

Vamos reforçar apenas dois dos doze princípios ágeis:

- Nossa maior prioridade é satisfazer o cliente por meio da entrega **adiantada** e **contínua** de *software* com valor agregado.
- Mudanças nos requisitos são bem-vindas, mesmo tardiamente no desenvolvimento. Processos ágeis tiram vantagem das mudanças visando à vantagem competitiva para o cliente.

Portanto, são 2 momentos distintos para se gerar a documentação necessária, o de **incremento**, gerado durante o desenvolvimento, interessante para que se adicione o critério de definição de pronto (*Definition of done*), garantindo que nada estará pronto

sem a devida documentação; ou a documentação de **backlog**, em que o *Product Owner* fará análise de prioridade, decidindo o momento de disponibilizar para a equipe de desenvolvimento. O que é possível notar é que a documentação passou a ser um resultado do desenvolvimento e não mais um insumo.

O código 3.1 é um exemplo de documentação técnica; um código fonte PHP com conexão ao banco de dados comentado durante o desenvolvimento de um estudante. Neste exemplo, ainda existe um excesso de informação, o ideal são os códigos extremamente limpos, contendo informações como nome, contato, descrição sucinta do que faz o código e a data de conclusão.

Código 3.1 | Código de um estudante e código de um programador – clean code

```
<?php

require_once ("connection-config.php");

/*
Compatibilidade: PHP 5.2, 5.3, 5.4, 5.5, 5.6, 7.0, 7.1, 7.2
*/

try {
    //StringConexao = $driver. ":host=" . $server . ";dbname=" . $database;
    //conn = new PDO($stringConexao, $user, $password);
    //Padrão sem se preocupar com acentuação
    //conn = new PDO("$driver:host=$server;dbname=$database", $user, $password);
    //Novo e obrigatório formato para que possamos usar os acentos!
    //-----$conn = new
    PDO("$driver:host=$server;dbname=$database;charset=utf8", $user,
    $password,
    array(PDO::MYSQL_ATTR_INIT_COMMAND => "SET NAMES utf8"));

    /*
    PONTE ENTRE O PHP E O BANCO DE DADOS
    -----
    Instancia um novo objeto do tipo PDO
    Indicando que deverá este objeto conectar ao banco, utilizando o driver do sgbd mysql,
    Acessando a base de dados chamada dbphp, no servidor indicado como localhost e utilizando
    as credenciais de acesso (user=root/password = usbw)
    */
    $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
```

```

/*Modifico atributos /propriedades do objeto de conexÃ£o informando que os erros devem
ser tratados utilizando/enviando EXCEPTION

Outros possÃveis tipos de Atributos de ERRO!!

PDO::ERRMODE_SILENT -- Fica quietinho e nÃ£o fala que deu erro -- MAS GERA LOG!
PDO::ERRMODE_WARNING -- ENVIA uma mensagem de ALERTA -- MAS GERA LOG!
PDO::ERRMODE_EXCEPTION -- Envia uma EXCEPTION que serÃ£o tratada dentro de um try
catch! -- MAS GERA LOG!

*/
}

catch( PDOException $Exception ) {

echo "Erro: " . $Exception->getMessage() . " - CÃ³digo: " . $Exception-
>getCode( );

die;
}

< ?>

CÃ“DIGO FONTE PROFISSIONAL

//Daniela Uebele - cel.:(99)99999-9999 - data da conclusÃ£o 13/02/2018
//MÃ©todo mais indicado para conexÃ£o por ser o mais seguro.
//http://php.net/manual/pt_BR/class.pdo.php
//Alterado por Andre Silva â€“ cel (99)99999-9999 - data da conclusÃ£o 20/12/2020
//para funcionar nas versÃµes atuais do PHP

<?php

require_once ("connection-config.php");

//Compatibilidade: PHP 5.2, 5.3, 5.4, 5.5, 5.6, 7.0, 7.1, 7.2

try {

//Novo e obrigatÃ³rio formato para que possamos usar os acentos!

$conn = new PDO("$driver:host=$server;dbname=$database;charset=utf8", $user, $password,
array(PDO::MYSQL_ATTR_INIT_COMMAND => "SET NAMES utf8"));

/*PONTE ENTRE O PHP E O BANCO DE DADOS - Instancia objeto PDO, driver do sgbd mysql,
bdif dbphp, credenciais de acesso (user=root/password = usbw)

*/
$conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);

```

```

}

catch( PDOException $Exception ) {

echo "Erro: " . $Exception->getMessage() . " - CÃ³digo: " . $Exception-
>getCode( );

die;

}

?>

```

Fonte: Sommerville (2018).

Para esses documentos, é interessante a existência de três elementos:

- **Contextualização do problema:** definindo o que ocorreu durante o processo de desenvolvimento.
- **Detalhamento técnico:** explicitar como, quando e a causa da intervenção, a fim de deixar os responsáveis cientes, em especial da existência de determinadas linhas de código.
- **Contextualizar soluções:** explicitar as alterações efetuadas no código para se obter correção do problema ou modificação em algum processo (COSTA; PEREIRA, 2019).

Lembrando que é relevante constar o nome de quem fez cada um dos processos, em todas as etapas do desenvolvimento do software (Figura 3.8). Essa abordagem diferenciada visa a gerar documentação que agregue valor real.

## MOTIVOS PARA CRIAR DOCUMENTOS IMPRETERIVELMENTE

Se for um código baseado em regulamentações; se a regra do negócio possuir uma complexidade em que o cliente entende ser importante uma formalização para minimizar problemas futuros ou mesmo informar aos usuários do sistema; simplesmente porque o cliente informou que deseja essa ou aquela documentação (CARVALHO, 2018).

Vamos retomar algumas perguntas: documentar é importante? Pode parecer exagero, mas registrar todas as etapas do projeto, as tomadas de decisões, por que foram tomadas e ter um histórico de tudo é sem dúvida muito importante. Gerar uma proteção para toda a equipe em relação ao processo de desenvolvimento talvez seja o principal motivador, mas ainda assim será comum encontrar projetos com pouca ou nenhuma documentação, o que provavelmente afeta a qualidade do produto ou serviço (COSTA; PEREIRA, 2019).

Dentro do assunto documentação, existe uma preocupação que é bem pouco citada, já que, em geral, em sala de aula, falamos de pequenos projetos, em que o desenvolvimento ocorre de forma individual ou dentro de um grupo de alunos, simulando uma pequena empresa, onde os desenvolvedores estão no mesmo espaço,

trabalhando lado a lado, mas é mais do que notório que a cada dia o *home office* acontece com maior frequência, então, levantamos outra questão!

Se não estamos em um mesmo espaço, como fazer esse controle e a atualização de documentações?

A utilização de ferramentas de software é fundamental; quando se inicia o desenvolvimento de um sistema, a equipe deve definir os padrões a serem utilizados, e a partir dessas convenções, é importante que se escolha os *frameworks*, as linguagens para o desenvolvimento e sistema para o controle de alterações no projeto, a fim de se poupar tempo, reduzir esforços e difundir com agilidade as informações sobre o projeto, funcionando como um repositório de informações, dessa forma, havendo uma base sólida para o trabalho a ser desenvolvido (SOMMERVILLE, 2018).

Vai muito além da utilização de ferramentas de processamento de texto para se gerar os documentos padrões de formalização, são necessárias planilhas eletrônicas ou softwares adequados para o controle da gestão de cronograma, de custo, de desempenho, por isso, são criados *frameworks* para a gestão de projetos, que são inúmeros, alguns oferecem serviços parcialmente gratuitos, como o Monday.com, Wrike, GoodDay entre outros, que trazem recursos para todas as áreas do conhecimento de forma integrada e o compartilhamento de atualizações preparado para o trabalho em equipe remota.

Para a gestão de cronograma, desempenho e visualização das atividades pendentes, em desenvolvimento e desenvolvidas em todas as suas etapas, o Trello é um software bastante conhecido, concorrente direto do Taskworld. Ambos oferecem inúmeros recursos baseados nos conceitos DoD e Kanban, como vimos na seção anterior, quando falamos do software kanbanFlow (COSTA; PEREIRA, 2019).

Para a documentação técnica durante a análise, são utilizadas as ferramentas de modelagem. Em geral, utiliza-se a metodologia UML, que traz inúmeras notações para a construção de diagramas, podendo, assim, representar diferentes visões do sistema, permitindo uma ampla análise do que será desenvolvido, trazendo a facilidade da representação visual, o que a torna bastante intuitiva. Ela é fortemente indicada para o desenvolvimento orientado a objeto, mas nada impede de ser utilizada para outras formas de programação.

Algumas ferramentas disponíveis são de uso gratuito, inclusive algumas *online*, quando partimos para alguns softwares que possuem recursos gratuitos, mas quando pagos, oferecem funcionalidades baseadas na engenharia reversa (cria os diagramas a partir do código fonte existente, importante é que esteja bem limpo e padronizado) ou, ainda, criam o código-fonte a partir dos diagramas, o que não significa que minimizou todo o trabalho de desenvolvimento, ainda será necessário o desenvolvimento de funcionalidade, mas, com certeza, oferecem um adianto.

Três exemplos de ferramentas bastante utilizadas para a criação dos diagramas UML:

- Astah UML: antigo JUDE, versões gratuitas e pagas desenvolvidas pela Change Vision;

- Enterprise UML: desenvolvido pela Sparx Systems; compatibilidade com linguagens como Java e C#; possui a função de criar código e engenharia reversa.
- Visio: está integrado ao pacote Office da Microsoft; possui outras representações gráficas suportadas.

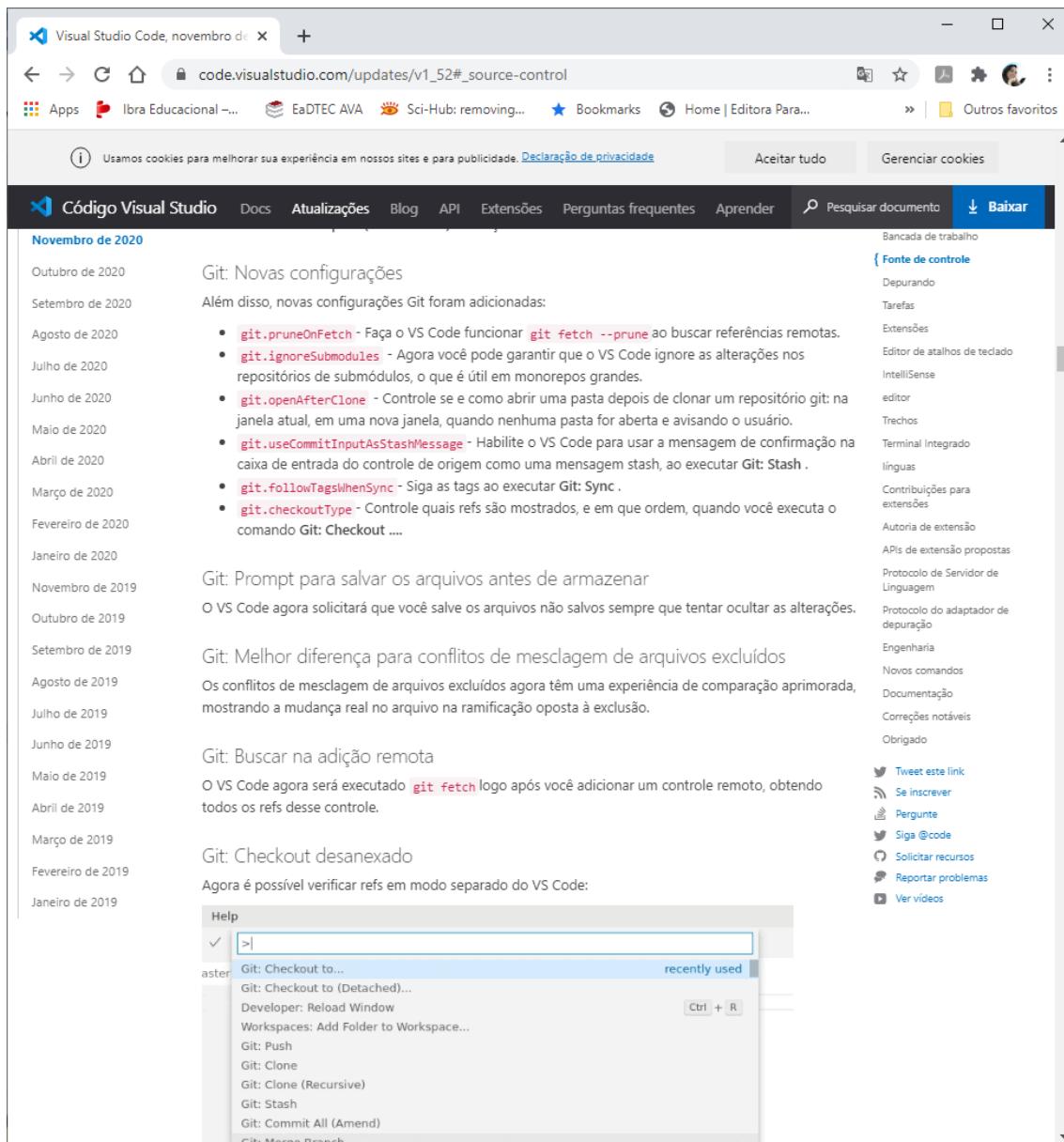
Todas essas ferramentas oferecem o recurso de exportação dos diagramas para arquivos de imagem, podendo ser inserida essa análise em arquivos de editoração de texto, gerando ou atualizando a documentação do desenvolvimento do sistema (SOMMERVILLE, 2018).

Após a criação da documentação técnica e análise, passa-se à utilização de editores para o desenvolvimento de software. Verifique que as ferramentas são constantemente tratadas como documentos independentes, e esse fato não procede; muitos deles oferecem recursos que são ignorados, como o de versionamento ou a criação automática de manuais técnicos e de usuários, que podem ser rastreados dentro do código. O Visual Studio Code traz esse recurso, entre muitos outros que são ofertados e desconhecemos (Figura 3.10 e 3.11).

Figura 3.10 | Visual Studio Code trazendo as últimas atualizações



Fonte: captura de tela do Visual Studio Code elaborada pela autora. Figura 3.11 Atualização com novas funcionalidades para o uso do Git, detalhando e oferecendo pequenos vídeos demonstrativos, observe no final da imagem



Fonte: captura de tela da plataforma Visual Studio Code elaborada pela autora.

Observe e acesse o *help on line* da ferramenta e navegue observando as melhorias; note que eles geram um versionamento bem claro e definido para que desenvolvedores e usuários desse recurso saibam o que está acontecendo e, ainda, possam participar das sugestões de melhorias do software. Esse é, sem dúvida, um bom exemplo a ser seguido para o desenvolvimento dos nossos sistemas.

A questão não é apenas documentar, mas como e onde documentar. São utilizadas em sala de aula algumas ferramentas desenvolvidas para um Ambiente Virtual de Aprendizado (AVA), como o *Teams*, da Microsoft, integrado com os aplicativos Office 365, além de recursos para a gestão de projetos, chat, entre outros recursos, bem como o Google Classroom integrado com o Google Drive, o Docs, o Sheets, o Slides e o Hangout, trazendo uma agenda comum à equipe. Tratam-se de plataformas que oferecem recursos de reuniões *online*, compartilhamento de arquivos, programas web de organização, editores de textos, planilhas eletrônicas, entre inúmeros recursos que

atende à questão de trabalho colaborativo entre os membros de uma equipe (SOMMERVILLE, 2018).

O que essas ferramentas trazem é um repositório, no entanto, existem alguns recursos que são integrados e ele, e esse é um ponto importante a ser trabalhado. Além dessas ferramentas voltadas para as instituições de ensino, existem os softwares profissionais que atuam também com o versionamento, como o conhecido Git, o Mercurial, TFS (*Team Foundation Server*), Subversion e CVS, que foi a primeira ferramenta de controle de software, desenvolvida em 1968, para o controle de versionamento. Mas por que fazer um versionamento de sistema? Quais os benefícios? (WAZLAWICK, 2013).

Nos projetos de desenvolvimento de sistemas na área de TI, é comum haver várias versões de sistemas, além, é claro, de desenvolvedores trabalhando na mesma atividade, não sendo muito incomum a existência de atualizações após o sistema estar implantado, o que torna o registro e o controle necessários, buscando-se garantir a compreensão de todos os membros da equipe e o andamento do projeto, além de evitar a perda de dados, facilitar a gestão e a conservação de um histórico das alterações do código e reduzir as inconsistências entre os vários documentos que compõem a documentação de um projeto de software (SOMMERVILLE, 2018).

#### [ASSIMILE](#)

Como fazer um versionamento no software?

Quem estabelece o versionamento são os próprios desenvolvedores ou responsáveis pelo projeto, ou seja, não existe uma padronização. O primordial é desenvolver um sistema de fácil compreensão, fundamentado e que traga a essência da proposta com a qual o software está sendo desenvolvido. Dessa forma, é possível gerar uma hierarquia de ordem crescente, composta por um conjunto de algarismo com significados, sendo o primeiro deles o de maior relevância, que estará vinculado a mudanças abruptas e de grande significado para o software; na sequência, serão definidos para as mudanças pontuais, como melhorias na segurança, podendo ser ajustes de *bugs*, *upgrades* da identidade digital do produto, entre inúmeras outras opções para haver um controle das mudanças mais significativas.

Não é necessário escolher apenas um software para cada área de conhecimento ou forma de se trabalhar, todos estes produtos ou serviços devem ser analisados no sentido de auxiliarem e agilizarem o processo de desenvolvimento, como documentar e manter a documentação atualizada, a equipe alinhada ao projeto e atualizada em relação às mudanças. O importante é que se defina apenas uma fonte de informação, ou seja, o repositório deve ser único, dessa forma, para os desenvolvedores, ele deverá conter arquivos comuns ao desenvolvimento: Banco de dados, o CSS, BootStrap, as API's, entre outros, pois serão utilizados por todos no projeto. Para se reduzir a curva de aprendizado é importante haver documentos de alto nível que explique a instalação, a configuração, os exemplos de utilização e a arquitetura implementada. Esses arquivos de alto nível também cabem para processos que venham a ter inúmeras subtarefas; nesse caso, os diagramas UML, que podem colaborar muito para a compreensão rápida do que foi feito ou se deve fazer, no caso do Banco de dados, o dicionário e a estrutura MER são fundamentais para os processos que envolvem cadastros, geração de relatórios e Buscas.

Vale lembrarmos que entre as consequências mais graves causadas pela ausência da documentação está a falta de credibilidade pelo cliente e os parâmetros para medição entre o previsto e o entregue, o que leva a se questionar diretamente a qualidade não apenas do produto, mas da empresa e dos profissionais. Mesmo ciente de que a documentação gerará gasto de recursos, é importante que se defina o momento adequado, o que documentar e como documentar. Consequências da falta ou da má documentação são desnecessárias ao suporte, já que o cliente não tem acesso à documentação das funcionalidades do sistema desenvolvido, o que consumiria tempo e a necessidade de treinamento dos novos colaboradores no time de desenvolvimento, considerando a rotatividade dos profissionais (NEWTON, 2011).

Repetir que o ambiente de desenvolvimento de projeto de sistemas é extremamente dinâmico e incerto pode ser redundante, porém é a realidade que, por muitas vezes, coloca a equipe do projeto a enfrentar situações que podem vir a fugir do controle, e uma das formas de auxiliar o enfrentamento dessa realidade com maior segurança é por meio de documentação adequada ao decorrer de todo o desenvolvimento, evitando-se riscos e deslizes desnecessários (PRESSMAN, 2016).

## DOCUMENTAÇÃO COMO ELEMENTO DE QUALIDADE

O primeiro item a ser considerado a fim de que a documentação seja relevante é a efetiva comunicação entre os *stakeholders*, área de conhecimento bastante discutida, pois é por meio de uma comunicação clara, eficiente e eficaz que se consegue manter a equipe engajada, pois não haverá dúvidas ou má interpretação, tornando o projeto alinhado com todos os *stakeholders*. Esse processo de comunicação pode ser por meio da criação de relatórios de status, contendo informações a partir de dados precisos e com frequência adequada, e a consequência do uso desses relatórios é uma expectativa realista (SOMMERVILLE, 2018).

Para que ocorra avaliações metódicas, o histórico referente ao projeto é um fator que eleva a importância da documentação. Em momentos oportunos, é possível que se resgate o que aconteceu para essa ou aquela tomada de decisão, facilitando a identificação das falhas ou até gerando as oportunidades que não foram previstas anteriormente. Um momento que pode gerar essa necessidade é quando se insere um novo membro à equipe; é por meio da documentação histórica que esse membro tem a possibilidade de se inteirar de tudo que ocorreu até o momento, ofertando uma oportunidade de contribuir com eficácia para as tarefas assumidas.

O documento de controle integrado de mudanças, como é citado no PMBOK, oportuniza a verificação de um ciclo para o controle das alterações, iniciando com a solicitação da mudança, passando para uma análise do impacto no projeto e tendo o *feedback* de aprovado ou de reprovado. No caso de aprovado, passa para o replanejamento, sendo revisto os riscos, o cronograma, os custos, a forma de se analisar se o requisito será atendido por completo (DoD) e a qualidade esperada para essa nova funcionalidade ou para a alteração. Após esse processo de atualização de todos os documentos, deve-se informar os *stakeholders* e executá-lo (NEWTON, 2011).

Por fim, voltamos a falar sobre o registro das lições aprendidas, que trará à equipe foco no desenvolvimento embasado em melhores práticas, otimizando o tempo e os recursos a cada nova atividade a ser desenvolvida, o que trará produtividade e eficácia. Cabe

ressaltarmos que a documentação das lições aprendidas inclui absolutamente tudo que deu certo e o que deu errado (PMBOK, 2017).

Levando-se em conta a metodologia ágil dentro de todo esse processo, vale considerarmos a documentação como parte integrante do processo de desenvolvimento, ou seja, a documentação pode ser visualizada como parte integrante do código, dessa forma, a documentação será entregue como parte do produto a ser desenvolvido, visando a evitar o retrabalho, facilitando a manutenção e reduzindo a inconsistências entre produto e documentação, podendo utilizar as mesmas ferramentas aplicadas ao desenvolvimento do software, até porque, essas ferramentas de amplo uso nas codificações, revisão e versionamento trazem o benefício da agilidade, do colaborativíssimo e da organização, muito estimulados pelas metodologias de gestão de projetos.

Todo esse contexto oportuniza um impacto positivo na qualidade da escrita e na publicação, agregando valor às documentações (COSTA; PEREIRA, 2019).

Em um mercado em que a agilidade no processo de desenvolvimento é valorizada tanto quanto a qualidade pelos clientes, com certeza, uma maneira de conquistar uma vantagem competitiva mediante a concorrência é a busca por meios de processos de gestão de risco e da qualidade, o domínio do conhecimento em relação às características e situações referentes à gestão de projetos de software, independentemente da forma de programação ou do método de gestão, bem como de qual ferramenta utilizar no auxílio das tarefas em todo o ciclo de vida do projeto, é fundamental a existência da documentação padronizada e compartilhada por todos os envolvidos (SOMMERVILLE, 2018).

#### **ASSIMILE**

Documentação é uma forma tangível de se descrever, de diversas formas, a representação de um sistema (Diagramas UML, códigos, requisitos funcionais e não funcionais etc.); gerar um cronograma com critérios bem claros, de forma a definir quando a atividades estará encerrada; deixar o cliente ciente de suas responsabilidades pelo levantamento de todas as suas necessidades, afinal, o projeto se encerrará e as funcionalidades deverão se manter funcionando; criar uma gestão das expectativas quanto à qualidade e ao impacto de cada uma das funcionalidades; e tornar esses itens parte de uma documentação formal, que trará limites e responsabilidades a todos os envolvidos (SOMMERVILLE, 2018; NEWTON, 2011).

Em geral, sabe-se que os princípios da metodologia ágil junto dos guias e métodos de gestão de projeto, de governança coorporativa e governança de TI levaram você, profissional da área, a ser um desenvolvedor flexível e adaptativo aos projetos e às empresas nas quais ingressar, bem como a ter um excelente relacionamento interpessoal, pois, por meio de todo o conhecimento e a experiência adquirida em sala de aula, leva para a vida profissional a organização e o planejamento, oferecendo ao seu cliente e colegas de trabalho muito mais que ética no trabalho e resiliência, mas também inovação e criatividade na forma de se conduzir o desenvolvimento, a documentação, a manutenção e, principalmente, o cumprimento do prazo, do custo e do escopo, o que resultará em produto de qualidade.

## FAÇA VALER A PENA

### Questão 1

“O custo e o prazo de um projeto são planejados em função do escopo. Portanto, é fundamental que sejam revisados ao longo do projeto para garantir que continuam coerentes entre si. SE o escopo do projeto mudar, consequentemente, o custo e o prazo também mudarão” (COSTA; PEREIRA, 2019, p. 123)

- I. O custo do projeto inclui todos os gastos necessários para planejar, executar e entregar o projeto ao cliente.
- II. O planejamento do escopo e a sua correta documentação são necessários para se saber se o projeto trará o retorno necessário.
- III. Evidentemente, não há a necessidade de se voltar para revisar o planejamento do projeto ao longo da execução, reanalizando, com cuidado, toda e qualquer alteração que ocorra.

Considerando o contexto apresentado, assinale a alternativa correta.

- a. Apenas as afirmativas I e II estão corretas.
- b. Apenas a afirmativa I está correta.
- c. Apenas a afirmativa II está correta.
- d. Apenas as afirmativas I e III estão corretas.
- e. As afirmativas I, II e III estão corretas.

### Questão 2

“O gerenciamento de risco no desenvolvimento ágil é menos formal. As mesmas atividades fundamentais ainda devem ser seguidas, e os riscos, discutidos, embora possam não estar **formalmente documentados**” (SOMMERVILLE, 2018, p. 612).

- ( ) O desenvolvimento ágil reduz alguns riscos, como os de mudanças de requisitos.
- ( ) O desenvolvimento ágil independe de pessoas, portanto, a rotatividade de desenvolvedores não trará efeitos significativos para o software a ser desenvolvido e para o projeto.
- ( ) A falta de documentos formais e a comunicação informal dificultam a continuidade e o ritmo no caso de desenvolvedores-chave deixarem o projeto.

Assinale a alternativa que apresenta a sequência CORRETA.

- a. V – V – F.
- b. F – V – V.
- c. V – F – V.
- d. F – V – F.
- e. V – F – F.

### Questão 3

“O saber adquirido guia o profissional, direcionando-o a soluções para problemas que já ocorreram, e que podem ser evitados com algum tratamento específico. Os acertos do

passado podem ser replicados no presente, e os do presente, no futuro” (COSTA; PEREIRA, 2019, p. 214)

- I. O principal objetivo de se conservar as informações de um projeto organizadas é poder usá-las como lições aprendidas.

POR TANTO

- II. Ao final do projeto, devemos criar o documento das lições aprendidas, incluindo não só as decisões de sucesso, mas as falhas nas decisões equivocadas, pois elas servirão de construção de conhecimento para projetos futuros.

Considerando o contexto apresentado, avalie as seguintes asserções e a relação proposta entre elas. A respeito dessas asserções, assinale a alternativa CORRETA.

- a. As asserções I e II são proposições verdadeiras e a II não é uma justificava da I.
- b. As asserções I e II são proposições verdadeiras e a II é uma justificava da I.
- c. A asserção I é uma proposição falsa e a II é uma proposição verdadeira.
- d. A asserção I é uma proposição verdadeira e a II é uma proposição falsa.
- e. As asserções I e II são proposição falsas.

**REFERÊNCIAS**

CARVALHO, F. C. A de. **Gestão de projetos**. 2. ed. Pearson Education do Brasil, 2018.

COSTA, A. B. da; PEREIRA, F. da S. **Fundamentos de gestão de projetos**: da teoria à prática – como gerenciar projetos de sucesso. Curtitiba: Intersabers, 2019.

MONTES, E. **Introdução ao Gerenciamento de Projetos**, 1. ed. São Paulo: Createspace Independent Publishing Platform, 2017.

NEWTON, R. **O gestor de projetos**. 2. ed. São Paulo: Pearson Prentice Hall, 2011.

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

PRESSMAN, R. S. **Engenharia de software**. 6. ed. Rio de Janeiro: McGraw-Hill, 2006. 720p.

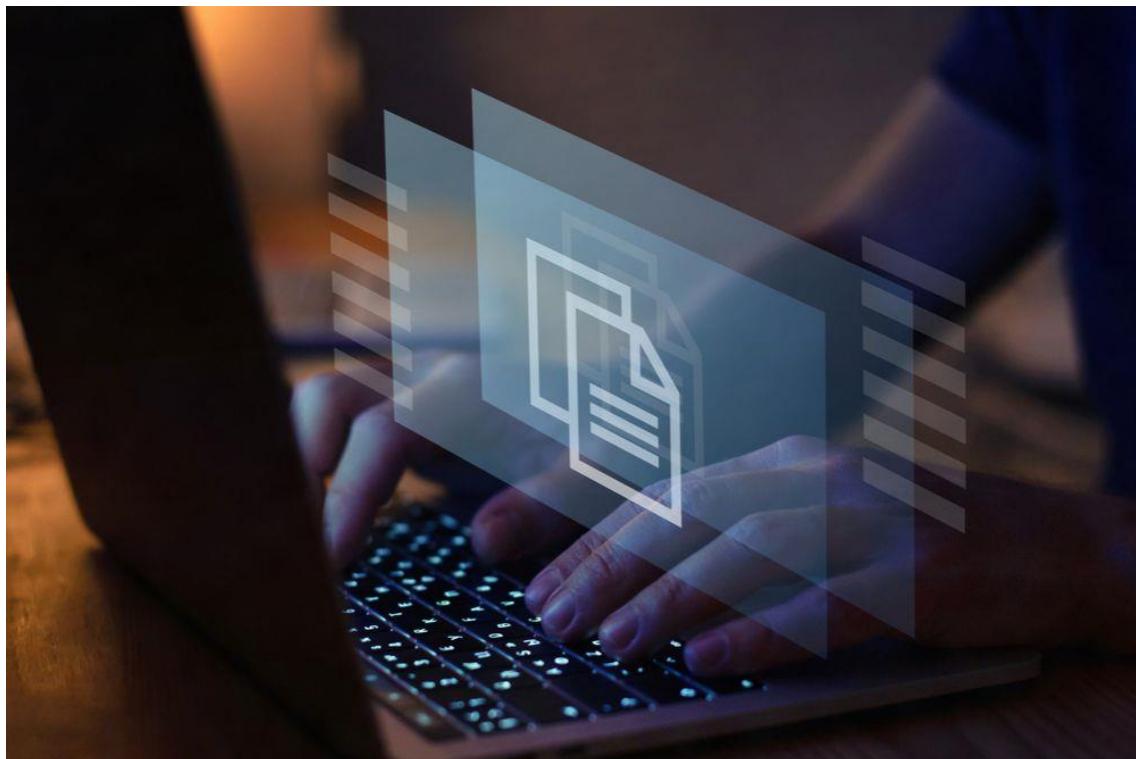
SHIMOKAWA, K; FUJIMOTO, T. **The birth of learn**. Cambridge: Lean Enterprise Institute, 2009.

SOMMERVILLE, I. **Engenharia de software**. 10. ed. São Paulo: Pearson Prentice Hall, 2018.

WAZLAWICK, R. S. **Engenharia de software**: conceitos e práticas. Rio de Janeiro: Elsevier Editora Ltda., 2013.

## ***FOCO NO MERCADO DE TRABALHO***

### *DOCUMENTAÇÃO DO PROJETO*



Fonte: Shutterstock.

**Deseja ouvir este material?**

#### ***SEM MEDO DE ERRAR***

Para que a sua equipe possa propor um sistema de visualização dos exames do paciente durante o procedimento cirúrgico, utilizando comando de voz e/ou gestos, evitando-se, assim, a perda de tempo no processo de higienização das mãos após manipular o computador ou os exames físicos. É de grande relevância que se tenha o TAP, um dos principais documentos que foi trabalhado na seção, visando sua importância do início ao fim do desenvolvimento do projeto. Já que este é o documento que formaliza o comprometimento entre as empresas para que ocorra o início do projeto.

Utilizando-se das informações já trabalhadas nas seções anteriores, fica a sugestão de uma possível solução, mesmo que ainda incompleta. Para o desenvolvimento desse projeto, considerando o mundo acadêmico, pode se utilizar a criação de uma equipe por meio da ferramenta Microsoft Teams, usando o kanbanFlow para a gestão das

atividades, aproveitando os recursos da ferramenta Microsoft Teams, utilizar o pacote office para a editoração de arquivos de alto nível e planilhas para o controle de alterações, a partir do documento de solicitação de controle de mudanças, após todo o processo de análise, afim de observar a quantidade de alterações que estão sendo utilizadas, o uso do sistema astah na criação dos diagramas UML, para a codificação do sistema o Visual Studio Code são sugestões de ferramentas integradas afim de conseguir bons resultados no processo de desenvolvimento deste projeto.

Controle de Versões			
Versão	Data	Autor	Notas da Revisão
1.0	24/08/2020	Provedor do Hospital	Daniela Uebele

#### ✓ Objetivos do documento

Oficializar o início do projeto, definir os responsáveis, as entregas essenciais, documentar requisitos iniciais, premissas, restrições, cronograma e custos para o projeto de criação de um sistema de análise de exame sem o contato dos profissionais com materiais físicos.

#### ✓ Cenário atual e a justificativa do projeto

Os médicos que necessitam manipular exames durante o procedimento cirúrgico têm o inconveniente de ter de remover luvas para essa atividade, sendo necessária a repetição de todo o processo de higienização para retomar o procedimento.

O objetivo do projeto é minimizar o tempo do procedimento cirúrgico, trazendo benefício aos profissionais envolvidos e principalmente aos pacientes, devido a um menor tempo anestesiado, como aos pacientes que estão à espera da liberação do centro cirúrgico, maximizando a utilização desse recurso.

#### ✓ Objetivos SMART e critérios de sucesso do projeto

O sucesso do projeto será alcançado ao atender a todos os critérios de aceitação das entregas, bem como às restrições e ao cronograma.

- Sistema com tempo de resposta ao gesto ou voz <= 5 segundos.
- Leitura de arquivos médicos de formados preestabelecidos.
- Permite utilização de forma tradicional, caso haja algum contratempo com a nova tecnologia.
- Sistema de segurança a falhas e privacidade do paciente adequada.
- Sistema concluído em 90 dias úteis, mediante documento de definição de pronto.

#### ✓ Estrutura analítica do projeto – Fases e principais entregas (principais entregas, versão inicial.)

- 1 Análise
  - 1.1 Prototipagem – esboço das telas com a definição das funcionalidades e forma de manipulação.
  - 1.2 Documentação – TAP; cronograma; análise de risco; planejamento de comunicação; análise do sistema UML.
- 2 Desenvolvimento
  - 2.1 Layout – criação dos layouts de tela com configuração visual adequada ao profissional.
  - 2.2 Leitura de arquivos médicos – definir quais são os formatos de arquivos médicos.
  - 2.3 Sistema de segurança – controle de log dos usuários, acesso apenas às informações pertinentes do paciente em procedimento.
- 3 Gestão de recursos
  - 3.1 Desenvolvedores – definir e controlar os recursos necessários para o desenvolvimento do sistema.
  - 3.2 Acessórios – aquisição dos recursos necessários para a implantação da tecnologia.
  - 3.3 Treinamento – treinamento da equipe desenvolvedora e dos usuários para a utilização do novo recurso.

Perceba que esse documento propõe conter o maior número de informações iniciais do projeto, podendo, sem dúvida alguma, sofrer alterações, no entanto, você e sua equipe

estão preparados para atender à nova análise e ao novo planejamento caso venha a ser necessário, e o domínio do conhecimento referente aos riscos que o projeto traz com ele será facilmente superado pela sua criatividade, seu planejamento e sua organização, além do processo criativo e inovador da utilização das várias tecnologias existentes no mercado.

Observe que compreender a necessidade de oferecer a qualidade em projetos de software é fundamental para que você e sua equipe sejam reconhecidos pela empatia de um bom relacionamento interpessoal acompanhado de ética profissional, flexibilidade e adaptação às inúmeras situações que vivemos. Parabéns pelo bom desempenho nesse processo de aprendizado, em que os desafios tendem a ser sempre maiores, mas o que vale é a persistência e o constante aprendizado.

## Unidade – 4 seção -1

# **NÃO PODE FALTAR**

## *DIFERENÇAS E SEMELHANÇAS NA CONCEPÇÃO E NO DESENVOLVIMENTO DO PROJETO*

Izabelly Soares de Moraes



Fonte: Shutterstock.

**Deseja ouvir este material?**

### ***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.

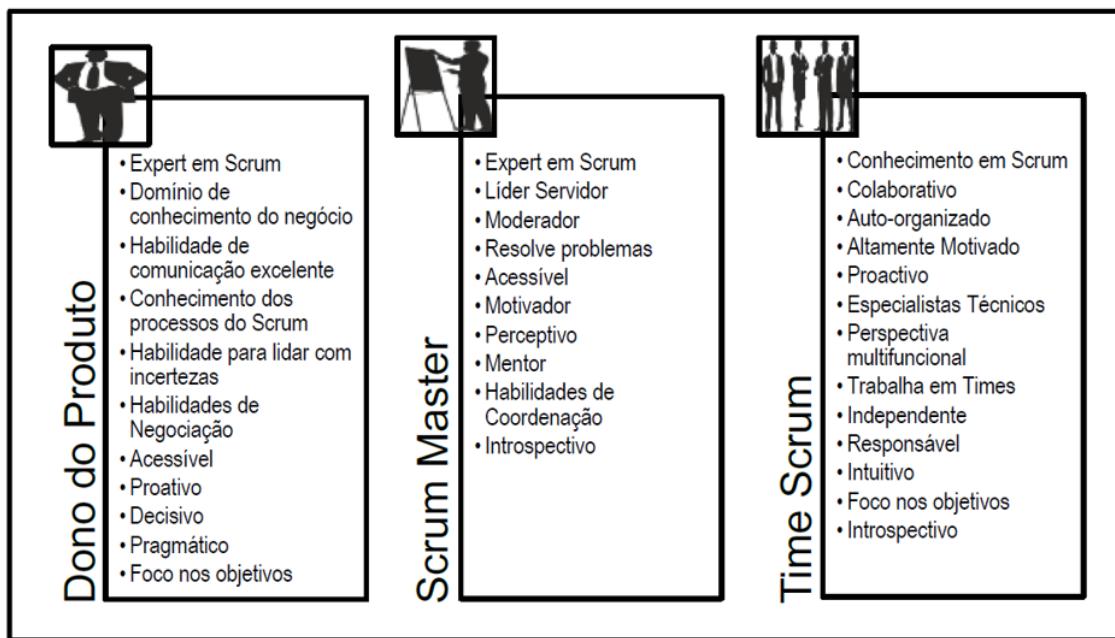
A organização de um time pode variar bastante, a filosofia ágil defende alguns princípios que acabam interferindo também nesse momento. Entre eles, estão: indivíduos e interações são vistos como mais importantes do que processos e ferramentas, colaboração com o cliente mais que negociação de contratos, 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 “enfatiza 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*.

Figura 4.1 | Papeis do *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 *storys*, 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 Scrum.	<ul style="list-style-type: none"> <li>• Fornecer <i>inputs</i> para a criação do plano de colaboração e plano</li> </ul>
Desenvolver os épico(s).	<ul style="list-style-type: none"> <li>• Garantir uma compreensão clara sobre os épico(s) e as persona</li> </ul>
Backlog priorizado do produto.	<ul style="list-style-type: none"> <li>• Compreender as estórias de usuário no <i>backlog</i> priorizado do p</li> <li>• Concordar com outros membros do Time central do Scrum sobr</li> <li>• Buscar esclarecer novos produtos ou mudanças nos produtos já</li> <li>do produto refinado.</li> </ul>
Conduzir o planejamento da <i>release</i> .	<ul style="list-style-type: none"> <li>• Concordar com outros membros do Time central do Scrum sobr</li> <li>• Buscar esclarecer novos produtos ou mudanças nos produtos já</li> <li>do produto refinado.</li> </ul>
Criar as estórias de usuário.	<ul style="list-style-type: none"> <li>• Fornecer inputs para o dono do produto na criação das estórias</li> </ul>
Aprovar, estimar e comprometer as estórias de usuário.	<ul style="list-style-type: none"> <li>• Estimar as estórias de usuário aprovadas pelo dono do produ</li> <li>• Comprometer as estórias de usuário a serem concluídas na spr</li> </ul>

<b>Processos</b>	<b>Responsabilidades do Time Scrum</b>
Criar as tarefas.	<ul style="list-style-type: none"> <li>Desenvolver a lista de tarefas com base em estórias de usuário.</li> </ul>
Estimar as tarefas.	<ul style="list-style-type: none"> <li>Estimar os esforços para as tarefas identificadas e, se necessário, ajustá-las.</li> </ul>
Criar o <i>backlog da Sprint</i> .	<ul style="list-style-type: none"> <li>Desenvolver o <i>backlog da sprint</i> e o gráfico <i>burndown</i> da <i>sprint</i>.</li> </ul>
Criar os entregáveis.	<ul style="list-style-type: none"> <li>Criar os entregáveis.</li> <li>Identificar riscos e implementar ações de mitigação de risco, se necessário.</li> <li>Atualizar o Registro de impedimento e dependências.</li> </ul>
Conduzir a reunião diária.	<ul style="list-style-type: none"> <li>Atualizar o gráfico <i>burndown</i>, <i>scrumboard</i> e o registro de impedimento.</li> <li>Discutir problemas enfrentados por membros individuais e buscas de soluções.</li> <li>Identificar riscos, se houver.</li> <li>Submeter solicitações de mudança, se necessário.</li> </ul>
Refinamento do <i>backlog</i> priorizado do produto.	<ul style="list-style-type: none"> <li>Participar de reuniões de revisão do <i>backlog</i> priorizado do produto.</li> </ul>
Convocar o <i>Scrum de Scrums</i> .	<ul style="list-style-type: none"> <li>Fornecer inputs ao <i>Scrum Master</i> para Reuniões do <i>Scrum de Scrums</i>.</li> </ul>
Demonstrar e validar a <i>Sprint</i> .	<ul style="list-style-type: none"> <li>Demonstrar ao dono do produto as entregas concluídas, que reforçam o valor do projeto.</li> </ul>
Retrospectiva da <i>Sprint</i> .	<ul style="list-style-type: none"> <li>Identificar oportunidades de melhorias na <i>Sprint</i> atual e concordar com elas para a próxima <i>Sprint</i>.</li> </ul>
Retrospectiva do projeto.	<ul style="list-style-type: none"> <li>Participar da reunião de retrospectiva do projeto.</li> </ul>

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*

<b>Processos</b>	<b>Responsabilidades do <i>Scrum Master</i></b>
Identificar o <i>Scrum Master</i> e o(s) <i>Stakeholder(s)</i> .	<ul style="list-style-type: none"> <li>• Ajudar a identificar o(s) <i>Stakeholder(s)</i> para o projeto.</li> </ul>
Formar o Time <i>Scrum</i> .	<ul style="list-style-type: none"> <li>• Facilitar a seleção do Time <i>Scrum</i>.</li> <li>• Facilitar a criação do plano de colaboração e do plano de trabalho.</li> <li>• Garantir a disponibilidade de backup de recursos para o backlog.</li> </ul>
Desenvolver o(s) épico(s).	<ul style="list-style-type: none"> <li>• Facilitar a criação de épico(s) e de personas.</li> </ul>
Criar o <i>backlog</i> priorizado do produto.	<ul style="list-style-type: none"> <li>• Ajudar o dono do produto na criação do <i>backlog</i> priorizado e pronto.</li> </ul>
Conduzir o planejamento da <i>release</i> .	<ul style="list-style-type: none"> <li>• Coordenar a criação do cronograma de planejamento da <i>release</i>.</li> <li>• Determinar a duração da <i>Sprint</i>.</li> </ul>
Criar as estórias de usuário.	<ul style="list-style-type: none"> <li>• Auxiliar o Time <i>Scrum</i> na criação das estórias de usuário e suas tarefas.</li> </ul>
Aprovar, estimar e comprometer as estórias de usuário.	<ul style="list-style-type: none"> <li>• Facilitar as reuniões do Time <i>Scrum</i> para estimar e criar as estórias de usuário.</li> </ul>
Criar as tarefas.	<ul style="list-style-type: none"> <li>• Facilitar ao Time <i>Scrum</i> a criação da lista de tarefas para a <i>Sprint</i>.</li> </ul>
Estimar as tarefas.	<ul style="list-style-type: none"> <li>• Auxiliar o Time <i>Scrum</i> a estimar os esforços.</li> </ul>
Criar o <i>backlog</i> da <i>Sprint</i> .	<ul style="list-style-type: none"> <li>• Auxiliar o Time <i>Scrum</i> no desenvolvimento do <i>backlog</i> da <i>Sprint</i>.</li> </ul>
Criar os entregáveis.	<ul style="list-style-type: none"> <li>• Suportar o Time <i>Scrum</i> na criação das entregas acordadas.</li> <li>• Ajudar a atualizar o <i>scrumboard</i> e o registro de impedimentos.</li> </ul>
Conduzir a reunião diária.	<ul style="list-style-type: none"> <li>• Garantir que o <i>scrumboard</i> e o registro de impedimentos estejam sempre atualizados.</li> </ul>
Refinamento do <i>backlog</i> priorizado do produto.	<ul style="list-style-type: none"> <li>• Facilitar as reuniões de revisão do <i>backlog</i> priorizado do produto.</li> </ul>
Convocar o <i>Scrum of Scrums</i> .	<ul style="list-style-type: none"> <li>• Garantir que os problemas que afetam o Time <i>Scrum</i> sejam discutidos e resolvidos.</li> </ul>
Demonstrar e validar a <i>Sprint</i> .	<ul style="list-style-type: none"> <li>• Facilitar a apresentação de entregas concluídas pelo Time <i>Scrum</i>.</li> </ul>
Retrospectiva da <i>Sprint</i> .	<ul style="list-style-type: none"> <li>• Garantir a existência de um ambiente ideal para o projeto, para as <i>Sprints</i> seguintes.</li> </ul>
Retrospectiva do projeto.	<ul style="list-style-type: none"> <li>• Representar o Time Central do <i>Scrum</i>, fornecendo lições aprendidas e melhorias para o projeto.</li> </ul>

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, consequentemente, 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*?

## TIME ÁGIL

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.
- b. V – V – F – F.
- c. F – V – V – F.
- d. V – F – F – F.
- e. F – V – 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.

# **FOCO NO MERCADO DE TRABALHO**

## *DIFERENÇAS E SEMELHANÇAS NA CONCEPÇÃO E NO DESENVOLVIMENTO DO PROJETO*



Fonte: Shutterstock.

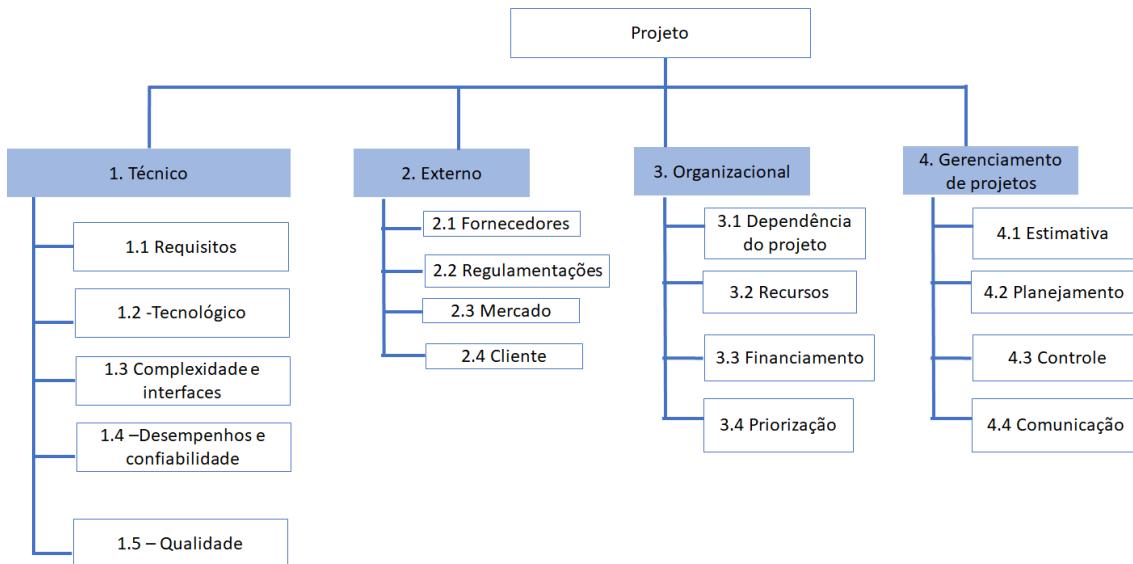
**Deseja ouvir este material?**

### **SEM MEDO DE ERRAR**

Como **gerente do projeto**, você deverá, a princípio, compreender bem o negócio do cliente, e se o projeto adotar o desenvolvimento ágil com *Scrum*, você terá a colaboração do *Scrum Master*, que vai garantir que os conceitos do *Scrum* realmente sejam aplicados, e do *Product Owner*, que assumirá o papel de “dono do produto”, apresentando a todos o que vai agregar valor ao negócio do cliente, nesse caso, da empresa do ramo alimentício.

Você, como gerente de projetos, tem um olhar macro diante dos passos que serão executados durante o desenvolvimento, então, deve elaborar uma Estrutura Analítica do Projeto, conhecida como EAP. Você pode usar vários programas para desenvolver o diagrama. Uma delas é a XMind.

Figura 4.2 | EAP - Estrutura Analítica do Projeto



Fonte: elaborada pela autora.

Você dividiu seu projeto em 4 etapas gerais, uma com perspectiva técnica, em que haverá obtenção dos requisitos com o setor de contabilidade e tecnológico para a definição de quais recursos serão essenciais ao sistema, das interfaces, do desempenho e da confiabilidade, já que será um sistema que utilizará dados sensíveis e precisos, e da qualidade tanto do sistema quanto de suas funcionalidades.

Posteriormente, você poderá definir quais fatores externos serão necessários ao seu projeto, como quem serão os fornecedores, que, no caso, podem ser os funcionários do setor de contabilidade, que podem ser internos da empresa ou até mesmo de empresas especializadas apenas em contabilidade. Outro ponto muito importante é o de regulamentações com leis locais e internas da empresa. O estudo do mercado, de definição e conhecimento dos clientes também é importante.

Outro ponto importante que você pode destacar é o organizacional, para ver uma perspectiva de dependência do projeto, quais recursos você vai precisar, podendo até mesmo ser uma licença de software, e pode abranger recursos humanos também. Tem as questões financeiras e de priorização. Além disso, tem o gerenciamento de projetos, como estimativa de planejamento, controle e comunicação, ou seja, como todos esses passos serão executados.

## NÃO PODE FALTAR

### FERRAMENTAS PARA GESTÃO DE PROJETOS

Izabelly Soares de Moraes



Fonte: Shutterstock.

Deseja ouvir este material?

### PRATICAR PARA APRENDER

Prezado aluno, ao olharmos ao nosso redor, deparamo-nos com vários conceitos que caracterizam a nossa vida em sociedade. Entre eles, podemos destacar os processos de comunicação, que estão sendo mencionados dessa forma porque, ao longo dos anos, o ser humano desenvolveu diversas habilidades para praticar a comunicação. Então, percorrendo uma trajetória histórica, houve uma evolução das pinturas rupestres das cavernas ao uso de recursos digitais, que podem promover a comunicação por meio da internet e demais ferramentas. Você já percebeu que estamos cada vez mais exigentes nessas práticas? Evoluímos, mas também mudamos não só a forma de nos comunicarmos, mas também de criarmos ferramentas, linguagens e outros meios de prover essa interação. Esses acontecimentos fazem parte do ciclo de vida de um software, em que mudanças passaram a ser constantes e, consequentemente, exigem que sejam dialogadas e documentadas.

Então, a relevância da comunicação em um projeto é perceptível, pois, a todo instante, algo está ocorrendo, precisa ser comunicado e documentado. E para suprir essas

demandas, surgiram várias ferramentas que abrangem não só alguns tipos de comunicação, mas também os perfis de pessoas, dentro de uma organização, que precisam ter acesso às informações. E o mais interessante é que existem formas que abrangem tanto metodologias sequenciais, vistas também como tradicionais, quanto metodologias ágeis. Então, quando você envia um e-mail, por exemplo, você está, de certa forma, realizando a comunicação com as pessoas.

Vendo uma perspectiva de projeto de software, esses recursos devem atender a demandas simples e complexas, então, exemplos como Trello, Azure DevOps, Asana, podem ser utilizados nesse cenário.

A indústria de jogos ficou em alta nos últimos anos, pois, além dos jogos eletrônicos, os jogos mais tradicionais, como os de tabuleiro, estão retornando ao mercado, porém assumindo, na maioria das vezes, um cenário diferente do que foi utilizado em sua criação, pois, hoje, tudo está automatizado, logo, esses jogos, estão sendo apresentados aos jogadores por meio dos aplicativos.

Devido às suas habilidades como gestor de projetos de tecnologia, você trabalha em uma empresa de jogos online e foi alocado para um projeto de um novo jogo; para isso, a empresa sugere que algumas funcionalidades de sistemas já existentes também sejam utilizadas, com o objetivo de trazer redução de custos ao ciclo de desenvolvimento do novo produto.

Conforme as atribuições de seu cargo de gestor de projetos, você deverá apresentar para a equipe uma perspectiva geral de algumas atividades principais que terão de ser desenvolvidas durante esse projeto, tais como: avaliar a possibilidade de haver ou não o reuso do software, qual modelo de desenvolvimento você e sua equipe utilizará, quem é o cliente de seu projeto, quem será o financiador, quem serão os usuários, qual problema o produto final deverá solucionar, entre outros. Para isso, você utilizará a ferramenta Trello, que já é utilizada por sua equipe em outras situações.

Perceba que a comunicação é muito importante, pois, além de manter todos informados, evita duplicidade nas informações, amenizando as chances de algo sair do planejado durante o desenvolvimento do software. Então, aproveite os conhecimentos adquiridos e os coloque em prática da melhor maneira possível!

## **CONCEITO-CHAVE**

### **INFORMAÇÕES ESPECÍFICAS DA COMUNICAÇÃO**

Ao longo dos anos, a sociedade foi criando formas de sobreviver às adversidades da vida humana. Dessa forma, desde seus primórdios, a comunicação se tornou a forma mais eficaz de interação, independentemente da forma que aconteça, pois, conforme os resquícios históricos, vários recursos podem ser utilizados para proporcionar essa ação.

Atualmente, devido à globalização, os avanços tecnológicos proporcionaram uma disseminação multicultural em diversos sentidos, possibilitando o envolvimento diverso de pessoas, processos, ferramentas e metodologias. Trazendo essa perspectiva ao

desenvolvimento de software, é possível termos conhecimento de tudo que está acontecendo em todo o mundo, graças à internet e ao interesse da comunidade de disseminar suas experiências, relatando pontos positivos e negativos. E o que isso acrescenta às nossas práticas? Tudo que temos de registros na atualidade é oriundo dessas vivências.

Antes de você conhecer informações específicas da **comunicação**, é preciso contextualizar um ambiente em que um software pode ser desenvolvido, e aqui, isso será feito de forma genérica, pois o software pode ser aplicado e originado em diversas possibilidades, porém com pontos em comum. Então, um dos primeiros pontos é que existirão pessoas envolvidas nisso tudo, além disso, a cada dia que passa, ferramentas também estão sendo inseridas nesse universo, proporcionando a automatização de diversas atividades.

### **PRINCÍPIOS FUNDAMENTAIS DE UM PROJETO DE SOFTWARE**

Alguns princípios são fundamentais para um projeto de software, entre eles, podemos destacar a necessidade da criação de valor ao negócio, produto ou serviço; além disso, ele deve atender às necessidades dos usuários, e isso exige certa execução de passos que norteiam alguma metodologia de desenvolvimento (SOMMERVILLE, 2011). Frente a isso, caro aluno, é importante que você saiba que existe certa complexidade, porém associada à quantidade de processos sendo executados ao mesmo tempo, envolvendo ações para a concepção do software, mas também com a existência de pessoas com habilidades e técnicas diferentes, porém presentes no mesmo projeto.

Por mais que pareça óbvio, a comunicação é um ponto de grande relevância dentro desse contexto, pois não envolve apenas o progresso do ciclo de vida do sistema, mas também a sua compreensão, que deve ser traduzida em atividades simples de serem compreendidas e desenvolvidas.

Aplicando o que foi comentado até então, de forma prática, saiba que a **comunicação** começa na interação entre a equipe do projeto e o cliente, ou seja, não se reserva apenas de forma interna, e sim com todos os envolvidos. Então, identificar os requisitos é um processo que deve ser executado de forma muito clara, pois, imagine documentar um requisito de forma incorreta seguindo uma metodologia tradicional; você, talvez, só descubra o engano no final do projeto, no momento em que pensava que tudo estava finalizado.

O Quadro 4.1 traz princípios relatados por Pressman *et al.* (2016).

Quadro 4.3 | Princípios da comunicação

Princípios	Descrição
Princípio 1. Ouça.	Foque em ouvir, tenha paciência e saiba definir suas necessidades. Deixe que os outros fiquem mais à vontade para relatar suas preocupações.
Princípio 2. Prepare-se antes de se comunicar.	É importante haver uma reflexão em relação ao que se deseja comunicar. Dedique um tempo para se preparar e saber o que se quer dizer. Isso inclui enquadração. Como quais termos são utilizados.

Princípios	Descrição
Princípio 3. Alguém deve facilitar a atividade.	Defina quem norteará a reunião, tendo como predefinidos com antecedência.
Princípio 4. Comunicar-se pessoalmente é melhor.	Procure trazer elementos que enriqueçam o meio de gráficos, protótipos e outros recursos mais produtivo, e todos poderão compreender.
Princípio 5. Anote e documente as decisões.	Utilize recursos tecnológicos para facilitar a criação de arquivos que armazenem informações que podem ser utilizadas posteriormente.
Princípio 6. Esforce-se para conseguir colaboração.	Ao utilizar as habilidades de sua equipe de forma eficiente, os membros envolvidos conforme as demandas que o projeto apresenta, e suas características do produto ou serviço. Além disso, é importante que todos estejam dispostos à exposição das informações.
Princípio 7. Mantenha o foco; crie módulos para a discussão.	Estabeleça e deixe claro os pontos principais que devem ser discutidos e resolvido naquele momento.
Princípio 8. Faltando clareza, desenhe.	Utilize recursos visuais, eles enriquecem sua comunicação, como esboços, protótipos, entre outros.
Princípio 9. (a) Uma vez de acordo, siga em frente. (b) Se não chegar a um acordo, siga em frente. (c) Se uma característica ou uma função não estiver clara e não puder ser elucidada no momento, siga em frente.	Caso haja algum empecilho diante de um acordado, é importante que os membros busquem novas soluções para serem apresentadas e, posteriormente, discutidas. Por isso, todos devem estar cientes dos prazos estabelecidos.
Princípio 10. Negociação não é uma competição nem um jogo; funciona melhor quando as duas partes saem ganhando.	

Fonte: adaptado de Pressman *et al.* (2016).

O diálogo acontece em vários níveis para atender a mais variada demanda, até porque está presente desde o primeiro contato entre todos, logo, acaba percorrendo pontos muito importantes em um projeto de software.

## ■ GUIA PMBOK E A COMUNICAÇÃO

De acordo com o Guia PMBOK (PMI, 2017, p. 360), a “comunicação é a troca de informações, intencional ou involuntária. As informações trocadas podem estar em forma de ideias, instruções ou emoções”. O guia apresenta alguns mecanismos por meio dos quais as informações são trocadas:

- **Em forma escrita:** podendo ser utilizados meios físicos (como cartas, notas) ou eletrônicos (e-mails, publicações).
- **Em forma falada:** podendo ser presencial (reuniões) ou remoto (reuniões remotas, videoconferências).
- **Formais ou informais:** por meio de documentos formais ou em mídia social.
- **Por meio de gestos:** tom de voz e expressões faciais.
- **Por meio de mídias:** imagens, ações ou mesmo pela escolha de palavras.

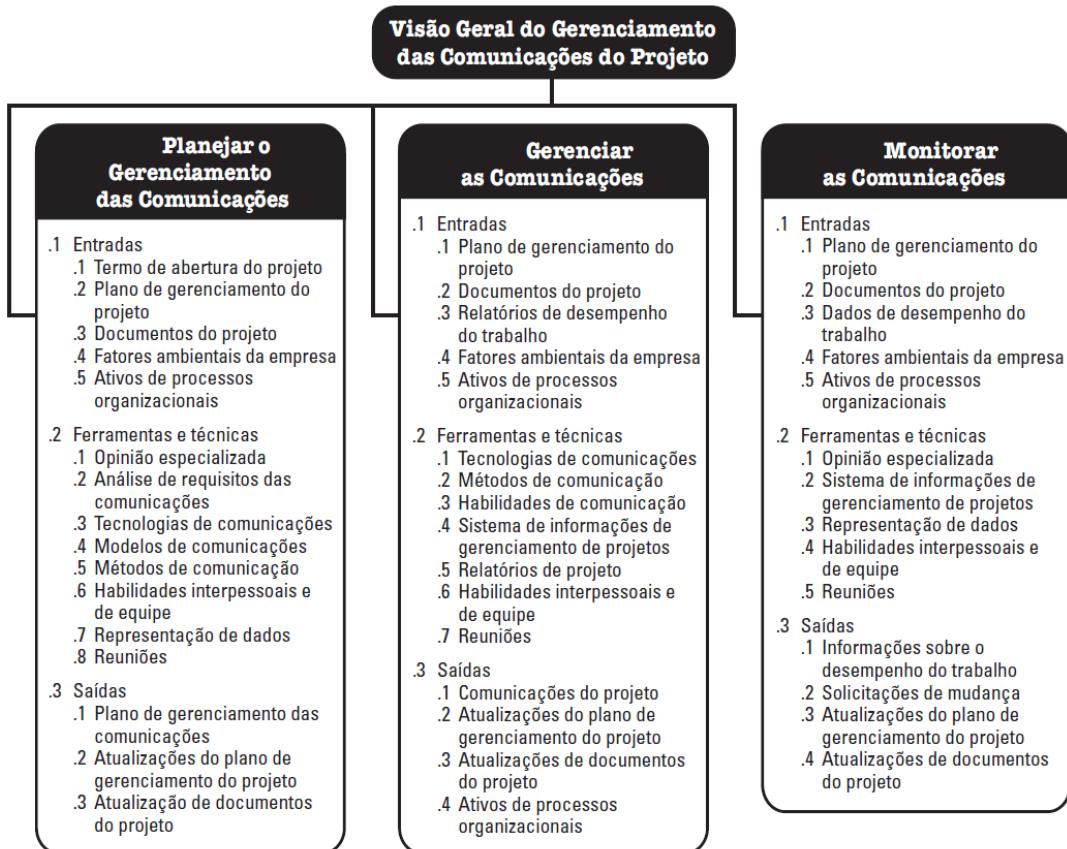
- **Escolha de palavras:** aquelas que melhor se adequem à ocasião, como o uso de termos técnicos em reuniões em que nem todos os presentes conhecem o contexto.

Além disso, o guia ainda destaca que a comunicação e suas respectivas ações podem assumir **dimensões** distintas, não sendo apenas:

- **Internas:** selecionando a quem deve ser direcionada determinada informação, ou seja, algumas informações fazem parte apenas da comunicação interna da equipe. Por exemplo: os detalhes e acontecimentos durante uma *sprint* não precisam ser externados ao cliente, tendo em vista que ele, provavelmente, não é especialista em assuntos técnicos voltados ao desenvolvimento.
- **Externas:** essas podem ser alinhadas para que o cliente possa compreender os acontecimentos do projeto. É importante mencionar que não é só o cliente que faz parte das partes externas, pois temos os fornecedores, outras empresas, setores governamentais, entre outros.
- **Foco hierárquico:** a comunicação pode ser adaptada conforme o público-alvo, então, em uma empresa, por exemplo, pode haver a necessidade da comunicação de um aviso que deva atender a todos os funcionários, então, pode ser realizada por meio de um e-mail, por exemplo. Porém, outros conceitos desse mesmo assunto também devem ser repassados aos acionistas em forma de gráficos, *dashboards*, que são quadros informativos, com dados, cálculos. Então, talvez nessa situação, a comunicação pode ser feita por meio de uma reunião ou videoconferência. E falando no foco hierárquico, o Guia PMBOK (PMI, 2017) destaca:
  - *Upward:* partes interessadas da alta administração.
  - *Downward:* a equipe e outras pessoas que contribuirão para o trabalho do projeto.
  - *Horizontal:* pares do gerente do projeto ou da equipe.

A Figura 4.3 apresenta uma visão geral do Gerenciamento das comunicações de um projeto.

Figura 4.3 | Visão geral do gerenciamento das comunicações do projeto



Fonte: SCRUMstudy (2016, p. 360).

Note que, por haver grande diversidade envolta da comunicação, é importante organizarmos etapas para gerenciá-la. Então, primeiro, há um momento para planejamento de como essa gestão será realizada durante um projeto de software; depois, esses planos serão colocados em prática por meio do próprio gerenciamento em si; em seguida, serão monitorados para tentarmos comprovar se nossas escolhas deram ou não resultados positivos. Em caso negativo, há sempre a lição aprendida, e nos próximos planejamentos, você poderá tentar fazer de outra forma, afinal de contas, é impossível saber se aquelas práticas realmente darão certo.

Além disso, perceba que, ao se tratar de projetos, com muitos processos, é importante documentar as informações, até porque nem sempre todos os envolvidos e interessados conseguem estar presentes em reuniões em que as decisões são tomadas.

Entre os diversos documentos e acordos que podem surgir em um projeto, conforme o Guia PMBOK (2017), é possível destacar alguns, tais como:

- **Termo de abertura do projeto:** é um documento que apresenta, na maioria das vezes, todas as informações sobre o projeto, abrangendo desde objetivos, a premissas, situação atual, justificativa do projeto e critérios de sucesso, em que metas podem ser apresentadas. Uma Estrutura Analítica do Projeto (EAP) também já pode ser inserida. Por meio da representação gráfica, é possível mostrar às pessoas como o seu gerenciamento de projeto acontecerá.

## EXEMPLIFICANDO

A Estrutura Analítica do Projeto (EAP), na verdade, também pode ser utilizada em outros contextos, já que apresenta uma subdivisão das atividades que vão ser desenvolvidas em um projeto. Outro ponto que vale destaque é que a EAP não precisa ser apresentada por meio de gráficos, ela também pode conter informações organizadas, semelhantes a um sumário. Por exemplo: você tem um projeto de comprar uma casa, logo, sua EAP pode ser organizada da seguinte maneira:

1. Comprar uma casa
  - 1.1 Gestão financeira
    - 1.1.1 - quitar dívidas
    - 1.1.2 - criar uma poupança
  - 1.2 Pesquisar casas à venda
    - 1.2.1 - definir preferências
    - 1.2.2 - escolher sites para busca

Não existe regras para quantidade ou a forma como será organizada. É indicado o seu uso apenas por facilitar a organização das informações mais importantes de um projeto (PMI, 2017).

- **Plano de Gerenciamento do Projeto:** geralmente, esse documento apresenta outros componentes do projeto, como escopo, qualidade, recursos. Então, ele apresenta uma lista de marcos, ou seja, etapas mais importantes do projeto, por exemplo: a aprovação do próprio Termo de Abertura pode ser vista como um marco, pois é um momento importante, assim como alguma entrega também pode ser considerada um marco. Na realidade, acaba tudo sendo muito subjetivo, já que depende das prioridades do projeto.

Dentro dos planos citados acima, estão presentes, também, o uso de **ferramentas** que auxiliam nos processos de comunicação, tanto que os próprios documentos citados também são ferramentas de comunicação. Elas estavam mais presentes e eram mais vistas dessa forma quando os projetos utilizavam com mais frequência as **metodologias sequenciais** ou tradicionais, como também são conhecidas.

## ASSIMILE

A metodologia sequencial, também conhecida como tradicional, surgiu em meados dos anos 1970, a partir das primeiras percepções a respeito da importância das metodologias durante o desenvolvimento de software. Posteriormente, surgiu o *Manifesto ágil*, trazendo novos valores aos que já existiam, visando a apresentar mais agilidade aos processos que antes eram pautados por seguir sequências e apresentar a necessidade de documentações mais extensas (PRESSMAN, 2016).

## METODOLOGIAS SEQUENCIAIS

As metodologias garantem sua comunicação por meio dos documentos, dos registros de tudo o que acontece em suas etapas, que são seguidas rigorosamente. Conforme Pressman (2016), a modernidade que os softwares tendem a seguir pode apresentar muitos desafios ao processo comunicativo, tendo em vista que as mudanças ocorrem e são aceitas com mais frequência, justamente devido à mudança das próprias

metodologias, até chegarmos aos princípios da **metodologia ágil**. Então, independentemente da metodologia, é importante definir mecanismos para a comunicação tendo uma vertente formal, porque tudo precisa ser documentado, e informal, para debates mais simples e que, posteriormente, serão formalizados.

Falando em documentação, não pense que a metodologia ágil não possui documentação, pelo contrário. A diferença de uma metodologia tradicional é que a documentação gerada é apenas essencial ao projeto, mas não podemos garantir que isso ocorrerá em uma metodologia tradicional. Com o passar do tempo, ferramentas e conceitos adicionais foram sendo desenvolvidos para apoiá-las durante o processo de comunicação, então, não precisamos separá-las. Na realidade, o gerente de projetos pode escolher a que melhor se adapta a sua necessidade, e alguns exemplos podem ser mencionados.

Mas, antes, contextualize um ambiente de desenvolvimento de software, imagine uma equipe com, em média, 10 integrantes divididos entre desenvolvedores, especificadores, designers, entre outros. Agora, imagine a quantidade de atividades que cada um terá para desenvolver ao longo do dia. O grande fluxo de acontecimentos e informações é evidente, então, o uso de uma ferramenta para auxiliar o gerente de projeto no acompanhamento do progresso de todos é essencial. Pontos a serem destacados são: as tecnologias utilizadas para prover a comunicação. Pense que, qualquer recurso que for utilizado deverá garantir também a segurança da informação. Alguns exemplos podem ser citados: uso de **e-mails**, **repositórios de compartilhamento** e armazenamento na nuvem (OneDrive, Dropbox, Drive), **videoconferências**, entre outros.

## MÉTODOS DE COMUNICAÇÃO

Existem, também, **métodos de comunicação** que podem interferir na **abrangência** de como as ferramentas atuam de acordo com o Guia PMBOK (PMI, 2017):

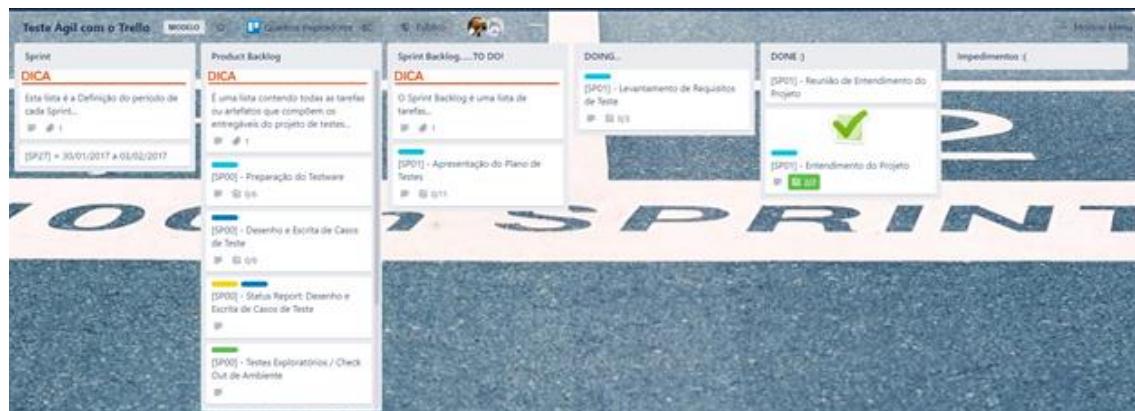
- **Comunicação interativa:** pode ser caracterizada pela comunicação entre dois ou mais interessados no compartilhamento das informações. As ferramentas mais utilizadas nesse cenário são reuniões, ligações telefônicas, videoconferências, entre outras.
- **Comunicação ativa:** ocorre de forma direcionada àqueles que precisam estar cientes de tais informações. Ocorre por meio do uso de alguns recursos, como relatórios, e-mails, publicações.
- **Comunicação passiva:** adequa-se mais a situações em que as informações são mais densas e detalhadas, ou seja, complexas, demonstrando, assim, necessidade de certos cuidados de segurança da informação, então, pode ocorrer por meio da intranet, do banco de dados, dos repositórios de conhecimentos, entre outros.

Após a metodologia ágil, surgiram muitas outras voltadas a melhorar a comunicação entre todos os envolvidos em determinada atividade. Então, algumas ferramentas foram surgindo ao longo dos anos e foram sendo aperfeiçoadas, algumas com ideias totalmente novas, mas outras automatizaram essas ações que mencionamos anteriormente, quando falamos das comunicações interativa, ativa e passiva. Entre as existentes, podemos destacar:

- **Trello:** é uma ferramenta que foi desenvolvida com o objetivo de oferecer funcionalidades referentes ao gerenciamento de projetos. Ele disponibiliza ao usuário vários *templates* voltados ao tipo de time que ele escolheu, e essas opções **são divididas conforme o contexto**, sendo educacional, empresarial, de tecnologia, entre outros. Por exemplo: caso escolha “Populares”, terá as opções de modelo para "Gestão de projetos", "Template Kanban", "Central de Organização da Equipe" e "Sistema de Produtividade Pessoal"; para a opção Design, são disponibilizados os modelos "Sprint de Design", "Equipe de Design de Produto", "modelo de requisição de design" e "Gestão de projetos"; já para a opção Engenharia/TI, ele apresentará as seguintes opções: “template Kanban”, “Retrospectiva da Sprint”, “Teste ágil com o Trello” e “Relatório de bugs”, e as opções variam conforme o contexto. Caso escolha "Pequenas Empresas", as opções serão “Visão geral da empresa”, “central de organização da equipe”, “OKRs”, “Análise SWOT”, e assim por diante. Além dos tipos citados, temo: Educação, Marketing, Recursos Humanos, Operações e CRM de vendas, todos possibilitam a você a criação de um quadro personalizado com suas necessidades. Como mostra a Figura 4.2, em que a escolha realizada foi “Teste Ágil com o Trello”.

Por exemplo, você pode escolher utilizar o Trello para ilustrar atividades que devem ser realizadas em seu projeto, e já que a ferramenta disponibiliza recursos visuais, você pode aproveitar e deixar visível a evolução do seu projeto, bem como criar espaços para que todos possam ver além do que deve ser feito, mas também o que está sendo realizado e o que já foi concluído.

Figura 4.4 | Exemplo de funcionalidade da ferramenta Trello

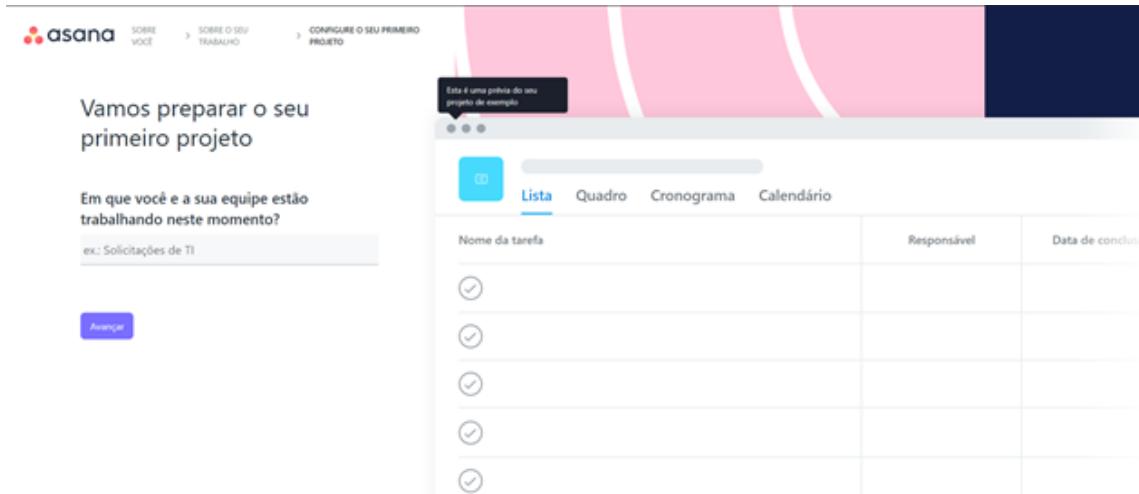


Fonte: captura de tela do Trello elaborada pela autora.

Outra ferramenta que também apresenta uma estrutura visual semelhante é a Asana, e os demais detalhes são apresentados abaixo:

- **Asana**

Figura 4.5 | Exemplo de funcionalidade da ferramenta Asana



Fonte: captura de tela do Asana elaborada pela autora.

Você vai notar que as ferramentas possuem opções de uso semelhantes. No caso do Asana, você tem a opção de **dividir por equipe**, em que a ferramenta apresenta algumas opções, como marketing, operações, vendas, produtos, entre outros. Já pelo **fluxo de trabalho**, pode ser Gestão de Projetos, Gestão de fluxos de trabalho, Calendário, Quadros Kanban, Equipes remotas, Produtividade, *Agile* e *Scrum*, entre outros. O que as diferencia é apenas a forma de operá-las, ficando a seu critério a escolha da que melhor se adapta.

Então, da mesma forma que o Trello, você pode criar quadros para expor as funcionalidades de seu sistema e direcioná-las aos componentes de sua equipe, assim, trará transparência em relação aos responsáveis por suas respectivas atividades.

- ***IceScrum***: a ferramenta possibilita a criação de *backlog*, em que o usuário pode representar as principais características do seu produto e organizar os requisitos com uma representação visual.

Figura 4.6 | Exemplo de funcionalidade da ferramenta *IceScrum*

The screenshot shows the IceScrum interface. At the top, there's a navigation bar with links like 'Peetic', 'Dashboard', 'Backlogs' (which is highlighted), 'Planning', 'Task board', 'Features', and 'Roadmap'. Below the navigation is a search bar and a 'New story' button. The main area is a 'Backlog' board with several cards. Each card has a title, a color-coded background, and some text below it. To the right of the board is a detailed view of a specific story card titled 'Resizing the pet profile breaks the styling'. The card includes sections for 'Details', 'Comments', and 'Tests'. It shows fields for 'Name' (Resizing the pet profile breaks the styling), 'Type' (Bug), 'Description' (In IE and Firefox, resizing the window width causes an overlap of fields), 'Feature' (Pet profile), 'Depends on' (No dependencies), 'Tags' (No tags), 'Effort' (5), 'Sprint' (Sprint 03), 'Value' (1), and 'Creator' (Izabella Morais). A 'Cancel' button is at the bottom right of the card view.

Fonte: captura de tela do IceScrum elaborada pela autora.

Então, por exemplo, você pode listar todas as funcionalidades do seu sistema e deixar as informações aparentes a todos. A vantagem de utilizar essas ferramentas aqui citadas é que são totalmente genéricas, e você pode adaptá-las conforme a necessidade de seu projeto, ou seja, ele não precisa, necessariamente, ser totalmente voltado à tecnologia ou ao desenvolvimento de um sistema, você pode, inclusive, utilizá-las para organizar suas atividades diárias.

- **Taiga:** a ferramenta disponibiliza opções de customização do espaço de trabalho, da forma visual, podendo, o usuário, escolher a opção do quadro *Kanban*, *Scrum*, Rastreamento de *bugs* (ou seja, erros), *Dashboards* (com opções de relatórios, painéis de desempenho, entre outros).

Figura 4.7 | Exemplo de funcionalidade da ferramenta Taiga

The screenshot shows the Taiga application. At the top, there's a navigation bar with icons for 'Help', 'Search', 'Dashboard', 'Backlog', 'Planning', 'Task board', 'Features', and 'Roadmap'. The main area is titled 'BACKLOG EXEMPLO DE PROJETO'. It features a progress bar showing '0%' completion with '0 pontos definidos' and '0 pontos fechados'. Below the progress bar is a section titled 'PERSONALIZE SEU GRÁFICO DE BACKLOG' with instructions on how to configure points and sprints. There are buttons for 'MOSTRAR FILTROS' and '+ ADICIONAR UMA NOVA HISTÓRIA DE USUÁRIO'. On the right side, there's a sidebar titled 'CORRIDA' featuring a cartoon character running and a message 'AINDA NÃO HÁ SPRINTS'. The bottom of the screen displays a decorative footer with cactus icons and the text 'O BACKLOG ESTÁ VAZIO! Você pode querer criar uma nova história de usuário'.

Fonte: captura de tela do Taiga elaborada pela autora.

Conforme vem sendo mencionado, apresenta, basicamente, um contexto totalmente genérico. O objetivo da apresentação das ferramentas é apresentar a você, caro aluno, a possibilidade de escolha, pois, como as ferramentas possuem funcionalidades comuns a diversos projetos, cabe a você selecionar a que melhor supre suas necessidades. Muitas vezes, a própria interface da ferramenta auxilia nessa escolha.

#### REFLITA

Perceba que os projetos de software podem atender a situações muito diversas, ou seja, desde sistemas hospitalares a sistemas de monitoramento terrestre, cada um trazendo sua complexidade e suas particularidades. Então, imagine os desafios que são impostos aos integrantes dessas equipes e os tópicos que devem estar presentes em seu gerenciamento. Como você planejaria o gerenciamento da comunicação nessas situações mencionadas? Quais documentos você selecionaria para serem utilizados?

O uso de ferramentas traz contribuições positivas ao projeto de software, tendo em vista que as suas funcionalidades evoluem conforme as demandas para o desenvolvimento de um sistema computacional. Além disso, com o passar do tempo, além de se aperfeiçoarem, dão a oportunidade da aplicação de padronizações e adaptações que agregam valor aos projetos.

#### FAÇA A VALER A PENA

##### Questão 1

O desenvolvimento de software envolve o uso de ferramentas e metodologias que visam a automatizar as atividades mais complexas desse processo. Além disso, a participação e a organização do time é essencial, e a comunicação é uma das práticas mais importantes que está presente desde o primeiro ao último passo.

Sobre a comunicação durante o desenvolvimento de um projeto de software, assinale a alternativa correta:

- a. Ouvir é um ato importante no processo da comunicação, por meio dessa ação, é possível compreender os relatos dos clientes.
- b. No processo de comunicação, é preciso dar razão ao cliente, e a equipe deve seguir todos os seus pedidos.
- c. O cliente pode nortear as reuniões, provendo os questionamentos que devem ser realizados.
- d. Os principais pontos de uma reunião poderão ser definidos no momento da reunião, por meio da documentação.
- e. A documentação impressa se torna um documento único gerado em reuniões, sendo também utilizado como meio de comunicação entre os envolvidos.

#### Questão 2

A comunicação está presente em todas as etapas do ciclo de vida de um software, dessa forma, existem princípios que a norteiam e ferramentas que auxiliam nesse processo. De certa forma, são de extrema relevância e deve ocorrer da melhor maneira possível. Por isso, existem, também, métodos de comunicação que podem interferir na abrangência de como as ferramentas podem ser utilizadas.

De acordo com as informações apresentadas no quadro a seguir, faça a associação dos feitos contidos na Coluna A com seus respectivos autores na Coluna B.

COLUNA A	COLUNA B
I. Ocorre de forma direcionada àqueles que precisam estar ciente de tais informações. Ocorre por meio do uso de alguns recursos como relatórios, e-mails, publicações.	• A comunicação é direcionada para aqueles que precisam estar ciente de tais informações. Pode ocorrer por meio de relatórios, e-mails, publicações, entre outros.
II. Adequa-se mais a situações em que as informações são mais densas e detalhadas, ou seja, complexas, demonstrando, assim necessidade de certos cuidados de segurança da informação podendo ocorrer por meio de internet, banco de dados, repositórios de conhecimentos, entre outros.	• A comunicação é adequada para situações em que as informações são mais densas e detalhadas, ou seja, complexas. Pode ocorrer por meio de internet, banco de dados, repositórios de conhecimentos, entre outros.
III. Pode ser caracterizada pela comunicação entre dois ou mais interessados no compartilhamento das informações. As ferramentas mais utilizadas nesse cenário são reuniões, ligações telefônicas, videoconferências, entre outras.	• A comunicação é entre dois ou mais interessados no compartilhamento das informações. As ferramentas mais utilizadas nesse cenário são reuniões, ligações telefônicas, videoconferências, entre outras.

Assinale a alternativa que apresenta a associação CORRETA entre as colunas.

- a. I-1; II-3; III-2.
- b. I-2; II-3; III-1.
- c. I-1; II-2; III-3.
- d. I-3; II-1; III-2.
- e. I-2; II-3; III-1.

**Questão 3**

O Gerenciamento das Comunicações do Projeto inclui os processos necessários para garantir que as necessidades de informações do projeto e de suas partes interessadas sejam satisfeitas, com o desenvolvimento de artefatos e a implementação de atividades projetadas para realizar a troca eficaz de informações.

(PMI, 2017, p. 359)

Considerando o contexto, avalie as afirmativas a seguir:

- I. Entre os princípios da comunicação, é possível citar a ação de ouvir e o preparo antes de uma reunião.
- II. Reuniões e ligações telefônicas caracterizam a comunicação passiva, ocorrendo, assim, a comunicação entre as partes interessadas.
- III. A comunicação ativa faz uso de recursos com abrangência maior, como publicações em revistas, jornais, e-mails, entre outros.
- IV. O uso de videoconferência caracteriza uma comunicação interativa, proporcionando a troca de informações entre duas ou mais partes.

Considerando o contexto apresentado, assinale a alternativa correta.

- a. Apenas as afirmativas I e II estão corretas.
- b. Apenas as afirmativas I, III e IV estão corretas.
- c. Apenas as afirmativas III e IV estão corretas.
- d. Apenas as afirmativas I e IV estão corretas.
- e. Apenas as afirmativas I, II estão corretas.

**REFERÊNCIAS**

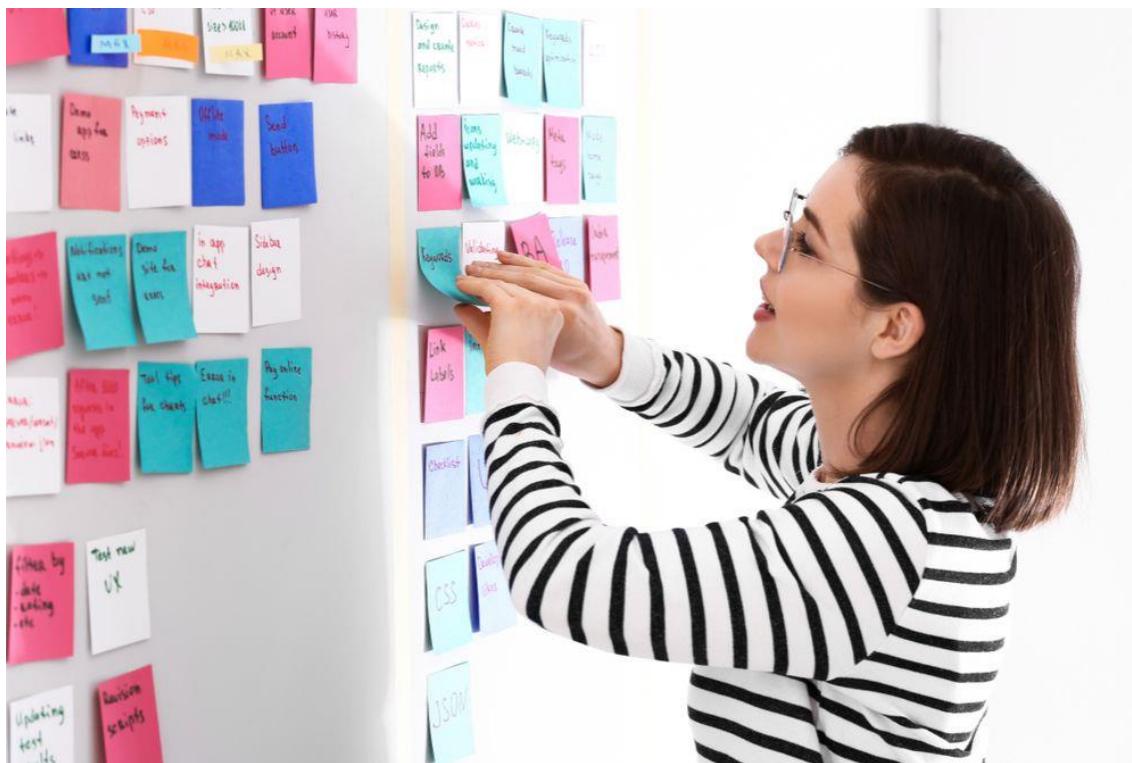
PMI — Project Management Institut. **Guia do conhecimento em gerenciamento de projetos**: guia PMBOK. 6. ed. [S.l.]: PMI, 2017.

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.

## **FOCO NO MERCADO DE TRABALHO**

### **FERRAMENTAS PARA GESTÃO DE PROJETOS**



Fonte: Shutterstock.

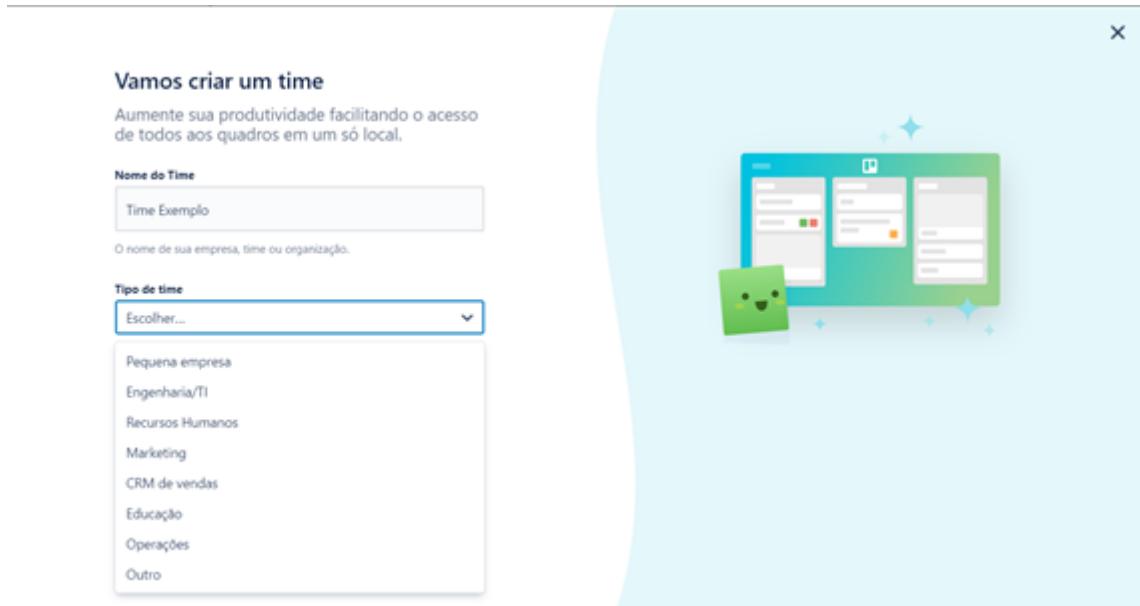
**Deseja ouvir este material?**

#### **SEM MEDO DE ERRAR**

Você fará uso da ferramenta Trello em um projeto para o novo produto da empresa de jogos online em que você trabalha. Seu desafio é alinhar com a equipe as principais atividades do projeto.

O primeiro passo é criar um espaço na ferramenta para o seu projeto. Para isso, comece inserindo o nome do Time que representará o projeto na ferramenta. Além disso, você pode escolher o tipo de time, ou seja, qual área está relacionado aquele espaço. Nessa tela, você ainda pode adicionar descrição de seu time para que todos que tenham acesso e saibam do que se trata o projeto.

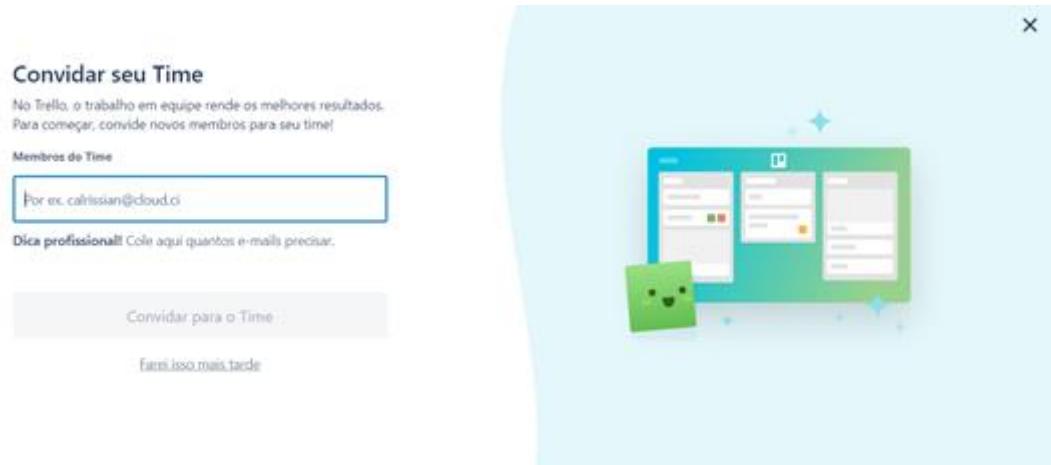
Figura 4.8 | Criação de um time de projeto



Fonte: elaborada pela autora.

Posteriormente, adicione as pessoas que fazem parte de seu time para terem acesso ao quadro.

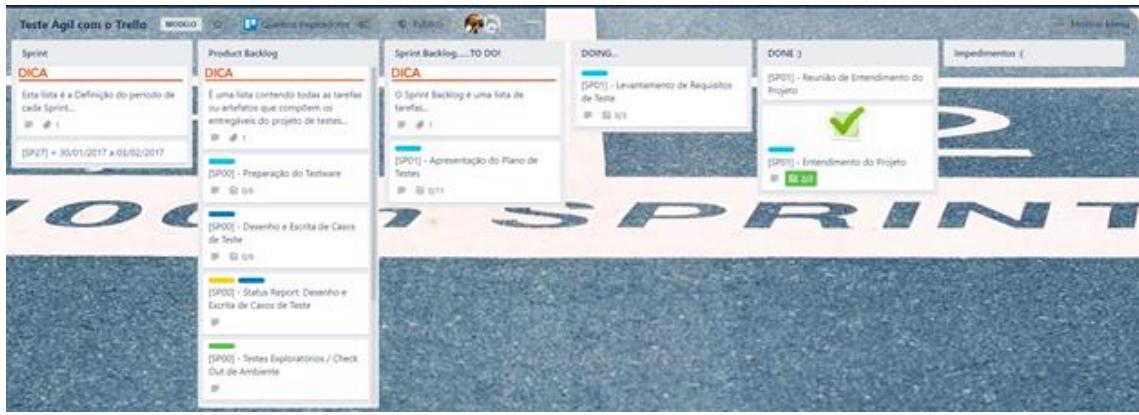
Figura 4.9 | Convidar o time



Fonte: elaborada pela autora.

Você terá várias opções de configuração visual para o seu projeto, e elas podem variar conforme o tipo de time que você escolher. Caso escolha a opção de Engenharia/TI, ela apresentará algumas opções, como o template Kanban, Retrospectiva do Sprint, Teste ágil com o Trello e Relatório de bugs. Uma sugestão é utilizar o “Teste Ágil com o Trello”. Observe, na figura abaixo, que ele apresenta as principais divisões de uma *sprint* do *Scrum*, com Sprint, *Product Backlog*, e assim por diante.

Figura 4.4 | Exemplo de funcionalidade da ferramenta Trello



Fonte: captura de tela do Trello adaptada pela autora.

Preencha com as principais atividades do seu projeto. Como, por exemplo, avaliar a possibilidade de haver ou não o reuso do software, já que foi indicado que haja, para haver redução dos custos, qual modelo de desenvolvimento você e sua equipe irá utilizar, como por exemplo, metodologia ágil *Scrum*, quem é o cliente de seu projeto, quem será o financiador, quem serão os usuários , nesse caso, esses três pontos serão importantes para que cada um assuma seu papel dentro da metodologia *Scrum*, então, o cliente, o financiador e os usuários serão denominados como *stakeholders*, lembrando que nem sempre o cliente é o principal e único financiador do projeto; além disso, o usuário pode assumir vários perfis, então, você pode criar as definições dos perfis e inserir no seu quadro trello, a fim de que todos sibam de todos os detalhes do projeto. E apesar de ter a possibilidade do projeto mudar bastante durante seu desenvolvimento, você pode listar, em seu quadro no trello, qual problema o produto final deverá solucionar. Mas lembre-se de que você, como Gestor do projeto, deve avaliar a questão do custo benefício de todas as suas ações. Além disso, o uso do Trello ajudará você na organização das informações norteadoras para sua equipe, trazendo informações da Sprint, da sua *Backlog* da Sprint, e assim por diante.

## ***NÃO PODE FALTAR***

***CICLO DE VIDA NAS DIFERENTES ABORDAGENS***

*Izabelly Soares de Moraes*



Fonte: Shutterstock.

**Deseja ouvir este material?**

### ***PRATICAR PARA APRENDER***

Prezado aluno, estamos susceptíveis a tomar diversas decisões em nosso cotidiano que podem provocar mudanças em nossos planos, e é sabido que algumas podem ter pontos positivos, mas não podemos garantir que sempre será assim, então, acaba sendo uma aposta. No processo de desenvolvimento de software, ocorrem situações bem semelhantes, pois não é considerado algo estável, no sentido de que deve permanecer da mesma maneira que foi implementado.

Esse fator ocorre devido às nossas necessidades de mudança, fazendo, nesse cenário, com que o software tenha que se adequar de alguma forma para suprir essas necessidades. Essas ações devem ser geridas com responsabilidade e com fundamentações que garantam sua qualidade, independentemente do que for realizado, e são postas em ação por meio da manutenção do software, que pode ocorrer tanto em paralelo ao desenvolvimento quanto durante sua implantação no ambiente do cliente.

A fase de implantação, além de demonstrar a necessidade de manutenção, também apresenta algumas atividades essenciais, como prestar o suporte ao usuário, realizar testes no sistema que foi disponibilizado ao usuário e criar uma *release* do software, que se caracteriza pela entrega das funcionalidades do sistema. A depender da metodologia, pode ser em partes ou do sistema como um todo, ou seja, é uma etapa responsável por definir quais são os artefatos que devem ser gerados, tendo sempre como base a geração de valor ao cliente. E essa escolha pode até ser com base em alguma mudança que precisou ser realizada. Além disso, também é uma ação que pode ocorrer durante o ciclo ou ao seu final.

Falando nisso, é importante compreender como ocorre o ciclo do software e o que acontece após ele ser entregue, já que as etapas de sua concepção podem ser resumidas, de forma genérica, em comunicação, planejamento, modelagem, construção e entrega, podendo apresentar algumas subatividades.

As atividades bancárias se fazem presentes no cotidiano das pessoas durante anos. Portanto, fazem parte de um setor que precisa se atualizar diante do cenário global, tendo em vista que seus clientes possuem perfis diversos. Além disso, no decorrer dos anos, a concorrência também foi um fator determinante na corrida contra o tempo para deixar o sistema sempre atualizado.

Você trabalha no setor de tecnologia de um banco, faz parte do setor de desenvolvimento e uma atividade de manutenção do sistema foi direcionada a você. Nela, você terá de desenvolver diagramas de casos de uso, trazendo melhorias e apresentando à sua equipe os detalhes de como algumas funcionalidades são executadas na atualidade, para que, com isso, consigam desenvolver um planejamento para a manutenção delas, onde poderá ou não haver alguma melhoria no sistema.

Um caso de uso representa como um usuário interage com o sistema, para isso, é possível elaborar o diagrama de caso de uso, em que as interações podem ser representadas utilizando-se o padrão UML. Conforme Pressman *et al.* (2016, p. 875), "um diagrama UML de caso de uso é uma visão geral de todos os casos de uso e de como eles estão relacionados. Fornece uma visão geral da funcionalidade do sistema."

A funcionalidade que deve ser desenvolvida é a de sacar dinheiro da conta corrente. Uma das sugestões é que você utilize a ferramenta "*lucidchart*", mas saiba que a escolha da melhor ferramenta fica a seu critério, tendo em vista que o caso de uso pode ser desenvolvido facilmente em outros tipos de ferramentas.

O *lucidchart* é um recurso web que disponibiliza diversos recursos para elaboração de artefatos, como diagramas. Além disso, eles proporcionam o compartilhamento e a construção coletiva desses materiais.

Para nortear sua resposta, realize a leitura do Apêndice 1, do livro de Roger Pressman e Bruce R., conforme indicação nas referências.

Conhecer as particularidades de um software é fantástico! Aproveite esses conhecimentos adquiridos para aprofundá-los e aplicá-los em situações práticas de seu cotidiano!

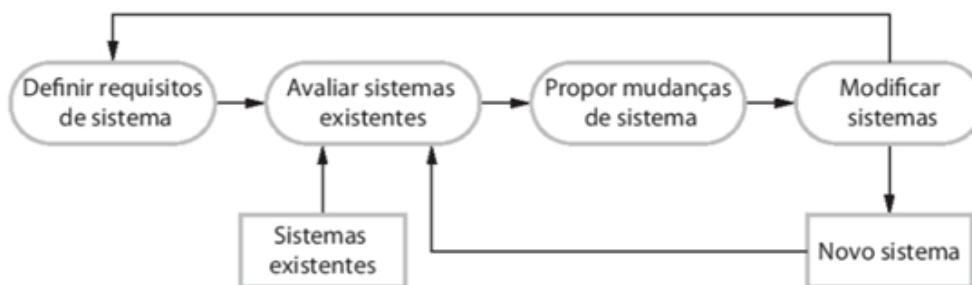
## CONCEITO-CHAVE

### Evolução do software

No decorrer do tempo, algumas percepções a questões que envolviam o software se tornaram mais evidentes; os sistemas que foram desenvolvidos inicialmente estavam ficando obsoletos, alguns por desempenharem automatizações que haviam se tornado desnecessárias para o cenário atual e outras que precisavam ser atualizadas para continuar suprindo as necessidades das pessoas, como uma simples calculadora, em que, na atualidade, é possível ter acesso a aplicações que apresentam funcionalidades de calculadora científica, muito utilizada em algumas áreas como engenharia. Outro exemplo é a planilha, que foi criada para automatizar atividades que antes eram desempenhadas no papel, porém, hoje, é possível até realizar importação diretamente de banco de dados e virtualizar as informações, além de ações que envolvem a computação em nuvem.

Note, caro aluno, que tudo acaba passando por um processo evolutivo. E com isso, novos ciclos podem ser executados diante de qualquer tipo de software. E para essa ação, na Figura 4.10, Sommerville (2011) destaca alguns estágios primordiais.

Figura 4.10 | Evolução do sistema



Fonte: Sommerville (2011, p. 29).

Ainda sob o ponto de vista de Sommerville (2011), ao mencionar o ciclo de evolução do sistema, é possível perceber que o desenvolvimento e a manutenção, na realidade, são duas atividades que percorrem, juntas, todo o ciclo do software. Então, por mais que tenha um momento para **definir os requisitos do sistema**, ele não deve, necessariamente, ser executado apenas uma vez; além disso, durante esse processo evolutivo, após essa ação, existe, ainda, a necessidade de se realizar a **busca por sistemas existentes**, em que pode haver, também, uma **avaliação**. Essas etapas são importantes, pois é uma oportunidade de se verificar o que já existe no mercado e, simultaneamente, descobrir novidades tecnológicas ou novas formas de solucionar certas questões. Após algumas análises, **mudanças são propostas**, sejam de funcionalidades já existentes ou para a indicação da necessidade de criação de novas, e após a **modificação do sistema**. Além disso, a depender da mudança, o sistema pode assumir um objetivo totalmente diferente de quando foi criado inicialmente.

## MANUTENÇÃO DO SOFTWARE

A manutenção também pode ser vista como uma etapa que ocorre após a entrega do sistema ao cliente, ou seja, após a implantação. Conforme Sommerville (2011), existem três diferentes tipos de manutenção de software:

- **Correção de defeitos:** nesse sentido, o autor apresenta duas perspectivas, em que menciona que ações de correção na codificação são menos onerosas do que erros de projeto, ou seja, a fase de planejamento é extremamente importante, pois pode ocasionar necessidades de mudanças após a finalização do sistema, o que sai muito mais caro, pois imagine que você terá de refazer todas as etapas.
- **Adaptação ambiental:** ocorre quando as funcionalidades precisam se adaptar às mudanças presentes no atual contexto em que o software precisa ser utilizado. Nesse cenário, mudanças não funcionais também podem acarretar esse tipo de manutenção, como alteração nos componentes físicos do sistema, mudanças de sistema operacional, entre outros.
- **Adição de funcionalidade:** nesse tipo de manutenção, os requisitos do sistema sofrem alterações e devem atender às novas solicitações da organização. Geralmente, ocorre com mais frequência.

A manutenção também pode receber termos adicionais, como "**corretiva**" ou "**adaptativa**", em que, respectivamente, um é utilizado para situações em que o sistema precisa da correção de algum problema e o outro é para adaptação, que pode se enquadrar a um novo ambiente ou a requisitos. Por ser uma ação que, com o passar dos anos, tornou-se frequente em outras etapas, diferentes apenas da implantação, existem algumas estimativas sobre a distribuição dos esforços e custos durante o desenvolvimento de um sistema. Além disso, mencionamos que o desenvolvimento tem como companhia, durante sua execução, a manutenção, porque também pode haver a chamada "**preventiva**", que ocorre durante o processo, trazendo melhorias ao sistema e garantindo que ele esteja atendendo às necessidades do cliente.

## RESPONSABILIDADES DA EQUIPE

Independentemente do tipo de manutenção, existem algumas **responsabilidades** imposta aos membros da equipe diante das possíveis ações que podem ser tomadas para realização de tal ação. Entre elas, deve haver uma estabilidade da equipe, levando-se em consideração que os envolvidos naquele processo acabam se tornando mais capacitados do que novos membros ou outras equipes, então, é indicado que a manutenção seja realizada pela mesma equipe que concebeu o sistema. Para tal feito, também se faz necessário trazer melhorias para a equipe, no sentido de promover qualificações a respeito das habilidades já existentes e às que podem se tornar necessárias no futuro, pois o sistema está em constante evolução.

Além de responsabilidades impostas às pessoas, existem também as que englobam o âmbito legal; ou seja, toda manutenção, assim como ações referentes ao sistema em si, deve ser documentada. No caso das manutenções, o indicado é que haja um contrato, selando os acordos realizados para as mudanças que serão feitas.

Conforme Paula Filho (2019, p. 286), vários papéis da organização se destacam nesses processos de mudança em um software, cada um trazendo uma **responsabilidade** compatível com suas habilidades:

- "Gerentes de produto, responsáveis pelas atividades referentes a um produto ou a um grupo de produtos;
- Representantes dos usuários, que acionam as solicitações de manutenção;
- Proprietários dos itens, que realizam as modificações necessárias nos itens afetados;
- Proprietários dos produtos, que tomam decisões eventualmente necessárias sobre cada produto;
- Comissões de controle de configurações, que tomam as decisões importantes relativas à gestão de configurações, em nível de produto ou de grupo de produtos correlatos;
- Administrador de configurações, que centraliza o funcionamento dos mecanismos dessa disciplina, também em relação às atividades de manutenção."

### **IDENTIFICAÇÃO DAS VERSÕES DO PRODUTO**

Como mencionamos anteriormente, as solicitações devem ser documentadas, e o processo que será executado deve ser muito bem definido, evitando que ações fora do escopo sejam executadas. Além disso, é importante que haja a identificação das versões do produto. Um exemplo que pode ser mencionado é o sistema operacional Android, desenvolvido para smartphones. Caso você realize uma busca no site, é possível acompanhar todas as mudanças ocorridas em suas versões. O histórico das mudanças que foram implementadas também apresenta informações importantes.

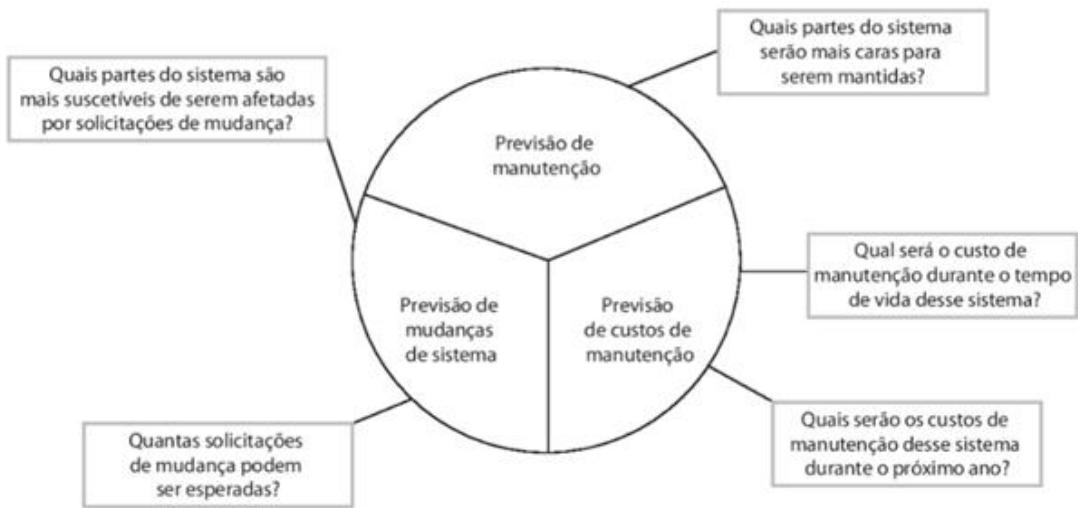
O que acontece é que, após o sistema ser colocado em uso, é possível obter algumas informações importantes para ratificar-se de que o processo utilizado para seu desenvolvimento foi bem executado. Por meio dos primeiros relatos, algumas necessidades de mudanças podem ser documentadas.

Para isso, existe o que é chamado de solicitação de modificação, e esse registro deve conter uma descrição do problema que fez com que a necessidade da mudança fosse retratada, dados para que a situação possa ser replicada e avaliada e estimativas de esforços para que a mudança seja realizada.

### **MELHORIAS REALIZADAS PELA MANUTENÇÃO**

Trazendo uma percepção de que a manutenção pode agregar mais valor ao sistema, ela pode também sinalizar as melhorias que podem ser realizadas. Então, é possível mensurar essas ações, como manutenção e seus respectivos tipos (preventiva, corretiva, adaptativa) e melhorias de pequeno ou grande porte. Abaixo, a Figura 4.11 traz alguns aspectos sobre a previsão de manutenção.

Figura 4.11 | Previsão de manutenção



Fonte: Sommerville (2011, p.173).

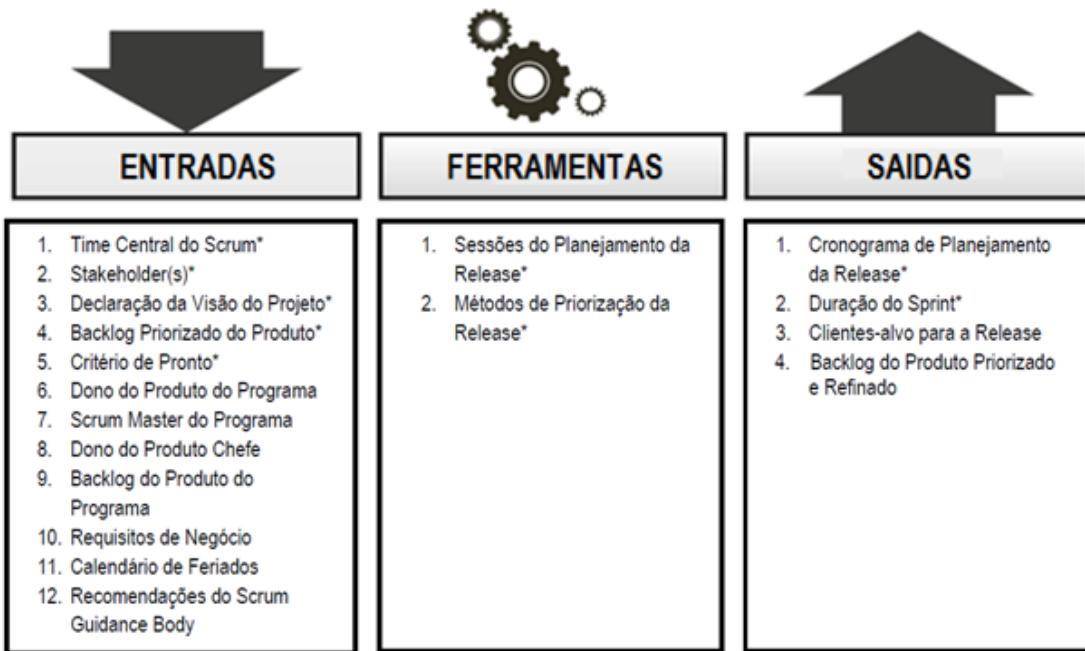
A Figura 4.11 traz aspectos relacionados à previsão de manutenção, então, alguns questionamentos podem ser realizados, e conforme as respostas são fornecidas, essa necessidade pode ser apresentada. Então, os questionamentos a serem realizados podem abranger os seguintes pontos:

1. **Custo de manutenção** diferente para cada parte do sistema, ou seja, é mais barato ir realizando testes durante o desenvolvimento, para se verificar as necessidades de mudanças, ou é melhor esperar a finalização do sistema?
2. Durante o tempo de vida do sistema, quais **mudanças** deverão ocorrer e qual custo de manutenção será obtido?
3. Existe algum planejamento para **gestão das mudanças**? Elas serão solicitadas conforme a necessidade do usuário ou a equipe fará testes constantes para verificar essas necessidades? Existem partes mais sensíveis à necessidade de mudanças? Por exemplo: um sistema hospitalar, devido ao seu alto grau de complexidade, por ser, na maioria das vezes, sistemas de tempo real, demandam testes e mudanças a todo momento, então, devem ser estimados

## ■ PLANEJAMENTO DE RELEASES

Aproveitando o exemplo do sistema hospitalar, é possível associá-lo à importância do planejamento das *releases*, ou seja, quais são as funcionalidades que serão entregues? Serão entregues a cada iteração ou apenas no final do ciclo? A depender da metodologia, as duas situações podem ocorrer. Conforme o PMI (2017), existem algumas métricas para conduzir o planejamento da *release*, que envolve entradas, ferramenta e saídas, como descrito na Figura 4.12.

Figura 4.12 | Planejamento da *release*



Fonte: PMI (2017, p. 173).

A Figura 4.12 demonstra o que pode compor cada particularidade do **planejamento de uma release**, então, como entrada, é possível notar uma lista de ações voltadas, nesse caso, ao uso da metodologia ágil *Scrum*, que aplica os principais conceitos do manifesto ágil, que são:

- Indivíduos e interações mais 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.

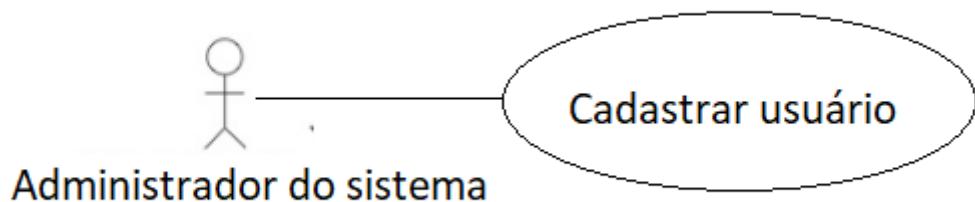
Conforme o PMI (2017), entre as ferramentas utilizadas para o planejamento da release, as Sessões de Planejamento da Release tem como objetivo, no *Scrum*, desenvolver um cronograma de plano da release, trazendo transparência para que o Time *Scrum*, que é formado pelos desenvolvedores, designers, especificadores, entre outros, possa ter uma visão geral de todo o projeto e passem a priorizar as principais entregas, ou seja os requisitos mais importantes. E como artefatos de saída, espera-se que o cronograma de planejamento tenha sido concluído e que traga informações, por exemplo, da duração da *sprint*, ou seja, da iteração. Além disso, é possível visualizar na figura que os clientes interessados nas atividades da *release* e o *backlog* priorizado e refinado também são tidos como resultados finais. É importante salientar que esses cronogramas e o planejamento da release podem ser atualizados, ou seja, os passos e as definições podem passar por mudanças.

#### EXEMPLIFICANDO

Conforme Pressman (2016), existem os modelos de requisitos que têm como base demonstrações dos **requisitos dos clientes**, então, podem apresentar domínio da informação, funcional e comportamental do sistema. Alguns artefatos podem ser gerados da manutenção do sistema, como o caso de uso, que é retratado por meio da UML (*Unified Modeling Language*), que apresenta uma linguagem padrão para modelagem de sistemas e disponibiliza muitos elementos e diversos tipos de diagramas;

entre eles, podemos citar o **diagrama de caso de uso**, como mostra a Figura 4.13, em que uma funcionalidade do sistema, que, no caso, é “cadastrar usuário”, é visualmente retratada pelo ator e pelo caso de uso. Nesse caso, o ator é a representação de quem vai executar tal ação, podendo, até mesmo, ser o próprio sistema; já o caso de uso é a ação, que é representada, conforme os padrões UML, pela figura geométrica elipse.

Figura 4.13 | Exemplo de Caso de Uso.



Fonte: elaborada pela autora.

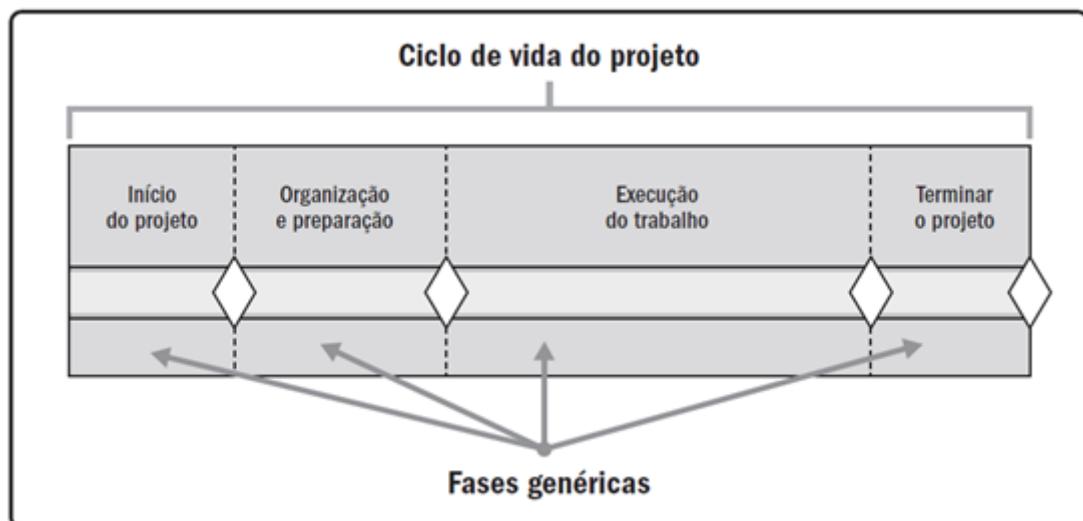
Uma ferramenta que pode ser utilizada para o desenvolvimento de caso de uso é a *lucidchart*, que possui algumas funcionalidades gratuitas e é online.

Além disso, há os **modelos de análise e projetos**, que trazem informações técnicas referentes à arquitetura, interface, entre outros. A UML também abrange esse tipo de modelagem.

#### CICLO DE VIDA DE UM PROJETO

O **ciclo de vida de um projeto** pode ser originado a partir de alguns passos genéricos, entre eles, o início do projeto, a organização, a preparação, a execução do trabalho e o término do projeto, conforme a Figura 4.14.

Figura 4.14 | Ciclo de vida do projeto



Fonte: PMI (2017, p. 548).

**O início do projeto** envolve a etapa de comunicação, que é primordial para dar início ao projeto de software. Ela é responsável por amenizar as complexidades apresentadas pela demanda do cliente, ou seja, por meio da comunicação, é possível deixar as informações mais relevantes acessíveis a todos os interessados no projeto, e é durante esse processo que, geralmente, os requisitos do software são obtidos, pois acabam definindo as metas e os objetivos do sistema de forma geral.

**A organização e a preparação** estão relacionadas ao planejamento, que é um momento dedicado à análise do que foi obtido durante a comunicação. Então, são definidas estratégias para alcançar os objetivos antes traçados, já que todo o escopo do projeto deve ser compreendido. Além disso, todos os envolvidos devem estar cientes das prioridades do projeto e da modelagem, que proporciona a apresentação das particularidades do sistema em diferentes níveis de abstração, podendo apresentar informações com linguagens simples e comuns ao nosso cotidiano, mas também podem ser utilizados termos mais técnicos.

**A execução do trabalho** é o momento que envolve a implementação, uma vez que as etapas anteriores já aprimoraram as ideias do que deve ser desenvolvido. Dessa forma, além da codificação na linguagem de programação escolhida, existem, também, as atividades de testes.

#### ASSIMILE

O diferencial de uma metodologia ágil e uma metodologia tradicional ou sequencial, como também é conhecida, é que a metodologia ágil traz ideias do manifesto ágil, mudando um pouco algumas abordagens levantadas ao longo dos anos pelas metodologias mais tradicionais. Como exemplo, é possível mencionar a execução de uma etapa associada à finalização da etapa anterior, o que não necessariamente ocorre em uma metodologia ágil. Como exemplo, podemos mencionar o *Scrum*, *Xp*, e como metodologia tradicional, temos a modelo cascata, evolucionário, iterativo e incremental, entre outros.

Então, é possível observar que existe um ciclo considerado genérico, mas que nada impede de que haja adaptações de suas etapas para suprir as necessidades dos envolvidos no projeto. Inclusive, **após a entrega** do sistema, existem etapas que ainda fazem parte do ciclo de vida do projeto, tais como a manutenção, o treinamento da equipe que utilizará o sistema (cliente e usuários), as melhorias no sistema que foi entregue, entre outros.

#### REFLITA

Na metodologia ágil *Scrum*, por exemplo, é possível visualizar a aplicação da mudança a partir do momento em que os artefatos gerados nas *sprints*, ou seja, nas iterações ou ciclos, são ou não aprovados conforme os critérios de avaliação do produto do cliente. Então, caso não seja, haverá mudança, a fim de que, ao passar novamente pelo ciclo, seja aprovado.

Porém, apesar de trazer um aspecto de liberdade e de ser totalmente aberto às mudanças, todas devem passar pelo gerenciamento, tendo em vista que se faz necessária, para um bom fluxo de desenvolvimento, a avaliação do que realmente faz ou não sentido para o projeto.

Então, o desafio é: como lidar e como definir o que deve ou não ser inserido nesse processo? Será que todas as funcionalidades realmente fazem sentido para o projeto da forma como foram definidas durante o processo de planejamento inicial?

O software é um artefato inconstante, que necessita de etapas bem definidas para ser desenvolvido, mas que, em relação ao seu uso, precisa fazer parte de um sistema de iterações e interações diversas, por isso, é importante conhecer um pouco sua manutenção, as responsabilidades envolta dessa etapa, o planejamento de releases e o ciclo de vida do projeto, tanto na perspectiva inicial, quanto após a entrega do sistema pronto.

### FAÇA A VALER A PENA

#### Questão 1

No ciclo de vida de um software são executadas diversas etapas, então, inicialmente, deve haver um planejamento, fundamentando as demais ações e os passos que deverão ser executados posteriormente. Os custos de investimento durante esse ciclo variam conforme a etapa; dessa forma, caso seja detectado algum erro no sistema e uma mudança tenha quer ser realizada, os custos serão variados.

A respeito da etapa mais onerosa para detecção de erros, assinale a alternativa correta.

- a. A etapa mais onerosa é a de manutenção.
- b. A etapa mais onerosa é a de planejamento.
- c. A etapa mais onerosa é a de implementação.
- d. A etapa mais onerosa é a de modelagem.
- e. A etapa mais onerosa é a de comunicação.

#### Questão 2

O desenvolvimento de um software se torna complexo, porém, não é uma complexidade pautada pela dificuldade e sim pela quantidade de funcionalidades, tecnologias, dispositivos que ele deve atender, até porque o seu uso só faz sentido se ele conseguir desempenhar sua função da maneira correta. Dessa forma, a existência de um ciclo de vida do projeto é muito importante para nortear todas essas atividades.

Considerando as informações apresentadas, analise as afirmativas a seguir:

- I. A comunicação é uma etapa executada no pós-entrega, quando o cliente recebe o sistema em seu ambiente.
- II. A organização e a preparação estão relacionadas ao planejamento e à modelagem do sistema que precisa ser desenvolvido.
- III. A implementação envolve não só a codificação do sistema, tendo como base as ideias aprimoradas anteriormente, mas também os testes.
- IV. O treinamento dos usuários é uma ação que caracteriza o início do ciclo de vida do projeto de software.

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

- a. I e II, apenas.
- b. II e III, apenas.
- c. III e IV, apenas.
- d. II, III e IV, apenas.
- e. I, II, III e IV.

### Questão 3

[...] organizações sempre almejam entregar produtos satisfatórios para clientes e usuários (alta qualidade), dentro de custos e prazos compensadores (alta produtividade) e previsíveis (alta previsibilidade). Para atingir esses objetivos, as organizações têm que entender seus próprios processos de negócio, a fim de aperfeiçoá-los e obter resultados cada vez melhores. No caso de organizações produtoras de software, os processos de negócio são os processos relativos ao ciclo de vida do software.  
(PAULA FILHO, 2019, p. 4)

De acordo com as informações apresentadas no quadro a seguir, faça a associação dos feitos contidos na Coluna A com seus respectivos autores na Coluna B.

#### COLUNA A

- |   |  |
|---|--|
| I. Neste tipo de manutenção, os requisitos do sistema sofrem alterações e devem atender às novas solicitações da organização. Geralmente, ocorre com mais frequência.   |  |
| II. Nesse sentido, o autor apresenta duas perspectivas em que menciona que ações de correção na codificação são menos onerosas do que erros de projeto, ou seja, a fase de planejamento é extremamente importante, pois pode ocasionar necessidades de mudanças após a finalização do sistema, saindo muito mais caro, uma vez que todas as etapas terão de ser refeitas. |  |
| III. Ocorre quando as funcionalidades precisam se adaptar às mudanças presentes no atual contexto em que o software precisa ser utilizado. Nesse cenário, mudanças não funcionais também podem acarretar esse tipo de manutenção, como alteração nos componentes físicos do sistema, mudanças de sistema operacional, entre outros.                                       |  |

Assinale a alternativa que apresenta a associação correta entre colunas.

- a. I-1; II-3; III-2.
- b. I-1; II-2; III-3.
- c. I-2; II-1; III-3.
- d. I-2; II-3; III-1.
- e. I-3; II-1; III-2.

#### *REFERÊNCIAS*

PAULA FILHO, W. de P. **Engenharia de software: projetos e processos.** 4. ed. Rio de Janeiro: LTC, 2019.

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.

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

# **FOCO NO MERCADO DE TRABALHO**

## **CICLO DE VIDA NAS DIFERENTES ABORDAGENS**

*Izabelly Soares de Morais*



Fonte: Shutterstock.

**Deseja ouvir este material?**

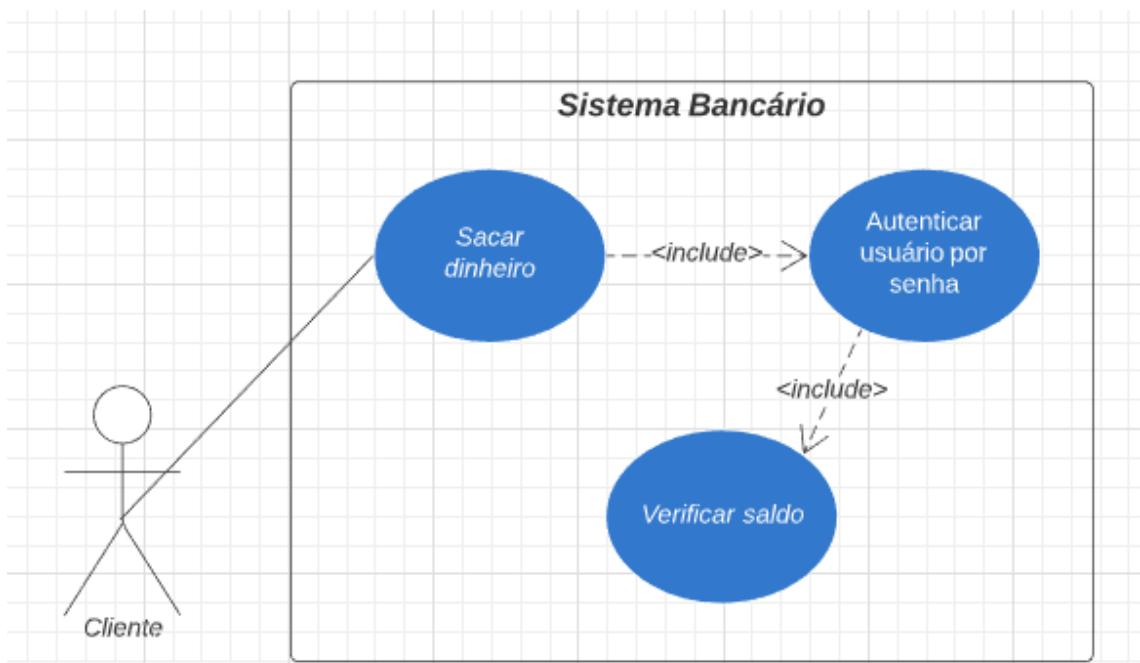
### ***SEM MEDO DE ERRAR***

Na sua atuação dentro do setor de tecnologia de um banco, você deve demonstrar à sua equipe como a funcionalidade de sacar dinheiro na conta corrente está sendo executada na atualidade, o objetivo é que, juntos, possam propor manutenção e atualização do software.

O primeiro passo é acessar uma ferramenta que possibilite a elaboração do diagrama de caso de uso. Como exemplo, você pode acessar a *lucidchart*, por ser uma ferramenta web que disponibiliza diversos tipos de diagramas. Além disso, lembre-se de que o objetivo do diagrama é representar a funcionalidade por meio de um recurso lúdico, que facilite a compreensão do sistema.

Como exemplo, você pode apresentar um diagrama como a figura abaixo:

Figura 4.15 | Exemplo de caso de uso



Fonte: elaborada pela autora.

O exemplo do diagrama apresenta o cliente como ator. Nele, é possível visualizar que a funcionalidade de “sacar dinheiro” pode depender de outras, tais como “autenticar usuário por senha” e “verificar saldo”. Por meio do diagrama, todos poderão compreender como a interação entre o usuário e o sistema ocorre e como a funcionalidade é executada; além disso, é possível notar sua dependência com outras. Nesse caso, trazem segurança a transação bancária.

O segundo passo é compartilhar seu diagrama de caso de uso com o professor e os demais colegas, para que, juntos, possam debater sobre quais melhorias podem ser realizadas em seu sistema e, caso necessite de alguma manutenção, como ela pode ser executada, por exemplo.