

Unidade 1

Fundamentos de Projetos de Software

Aula 1

Introdução a Projeto de Software

Introdução a Projeto de Software

Este conteúdo é um vídeo!



Para assistir este conteúdo é necessário que você acesse o AVA pelo computador ou pelo aplicativo. Você pode baixar os vídeos direto no aplicativo para assistir mesmo sem conexão à internet.

Dica para você

Aproveite o acesso para baixar os slides do vídeo, isso pode deixar sua aprendizagem ainda mais completa.

Estudante, esta videoaula foi preparada especialmente para você. Nela, você irá aprender conteúdos importantes para a sua formação profissional.

Vamos assisti-la? Bons estudos!

Ponto de Partida

Com as grandes transformações digitais, o desenvolvimento de software é um dos setores que têm apresentado crescimento rápido e constante transformação, exigindo o aprendizado de novas habilidades e aptidões e de profissionais com perfil inovador para cargos que sequer existiam até então. Você deve ter percebido que essa é uma realidade em qualquer segmento de mercado que se pense!

Outra coisa que você deve ter percebido é o quanto gestores e empreendedores estão procurando a reestruturação ou a pivotagem em negócios no Brasil, tudo devido a esses momentos de instabilidade aqui no Brasil (como um reflexo do que o mundo está passando). A área de TI – Tecnologia da Informação é uma das mais lembradas, tanto como ferramenta de inovação quanto por ser um recurso adaptável para suportar a agressividade dos concorrentes.

PROJETO DE SOFTWARE

Investir em tecnologia não é voltar-se apenas para a inovação, mas é uma forma de sobreviver no mercado e de se tornar ou permanecer competitivo (ZIMMERMANN, 2001).

Agora você deve estar pensando em como a disciplina Projeto de Software vai te ajudar nesse grande complexo e desafiador segmento, não é mesmo? Bem, aproveite para refletir sobre o conteúdo abordado com a sua realidade para conseguir obter a competência necessária para o desenvolvimento de suas tarefas profissionais com habilidade. Você aprenderá que muitas técnicas e metodologias serão úteis em alguma fase da sua vida profissional.

Fique atento, pois pretendemos que você conclua essa aula capaz de elaborar um plano de projeto que contemple pessoas que vão se envolver e engajar, vão promover o sucesso do projeto de software e, principalmente, vão talhar todas as demandas para uma gestão adequada dos recursos materiais ou solucionar possíveis problemas.

Bons estudos!

Vamos Começar!

Importância da gestão de projetos

A melhor maneira de compreender um assunto é conhecer um pouco da origem dos termos estudados sobre o tema. “Projeto” vem do latim *projectus* e significa “Plano, planejamento que se faz com a intenção de realizar ou desenvolver alguma coisa.” (PROJETO, 2020).

Se você realiza algo que tenha relevância, um produto que gerará resultados úteis para uma pessoa ou organização, deve ter um planejamento, o qual deve ser desenvolvido antes, com a criação de um plano; durante a execução do projeto, o plano poderá passar por adequações/atualizações conforme as necessidades emergentes. Pela complexidade dessas atualizações, entendimento dos impactos das adequações é que se faz necessária a gestão do projeto de software.

Para desenvolver uma aplicação de software, comece a pensar quais elementos são imprescindíveis para a realização das tarefas do desenvolvimento dessa aplicação: quem (pode ser o patrocinador ou usuário-chave) definirá as funcionalidades (podemos chamar de requisitos funcionais), qual banco de dados e linguagem de programação serão adotados para a implementação, quando deverá ser concluído o produto de software, entre outros elementos.

Ao agrupar os primeiros elementos, você perceberá que muitos detalhes são necessários para o bom andamento na execução das atividades. Pense na duração de uma tarefa ou atividade, ou fase do projeto, pode estabelecer horas, semanas, meses ou anos, certo?

Escolho a tarefa “Definição das funcionalidades” para demonstrar que existem muitas variáveis ao longo desse procedimento:

PROJETO DE SOFTWARE

- Consigo definir sozinho todas as funcionalidades da aplicação?
- As funcionalidades que estou imaginando hoje são importantes para todos que vão utilizar o software?
- Em quantas horas/dias consigo definir todas as funcionalidades dessa aplicação?
- Posso definir algumas funcionalidades e aprimorá-las ao longo do projeto?
- Se eu começar a desenvolver com uma determinada linguagem de programação, será que a aplicação vai atender a todas as necessidades em função das demandas futuras?

Estudante, você concorda que com apenas algumas hipóteses podemos ter a percepção da necessidade da gestão de um projeto de desenvolvimento de software?

Agora que você já percebeu que vários elementos estão inter-relacionados e são interdependentes, ou seja, todos estão relacionados para resolver ou gerar uma solução, é possível imaginar quais são as competências e habilidades que um gerente de projetos deve reunir para garantir o sucesso de um empreendimento.

Compreender o papel do profissional que conduzirá adequadamente o projeto é uma das maneiras mais diretas para aprender o que é Projeto de Software como uma disciplina essencial na vida profissional da TI. Harry Kezner (2020) disse que “o gerente de projetos do futuro deverá ser responsável por concluir o projeto no prazo, no orçamento e, também, se responsabilizar pelo retorno financeiro”. Segundo o *Project Management Institute* (PMI®), organismo que reúne as melhores práticas, técnicas e experiências acerca da gestão de projetos no mundo, esse profissional tem que ter três talentos fundamentais, conforme representado pela Figura 1:

- Em Técnicas em gestão de projetos – reunindo as boas experiências em projetos anteriores.
- Em liderança – o comportamental que proporciona a boa relação entre pessoas e eficiências na execução das tarefas.
- Em gestão estratégica e negócios – um alinhamento necessário para garantir que necessidades e mudanças estão sendo atendidas.



Figura 1 | Triângulo de talentos® do PMI. Fonte: Camargo (2019, p. 19).

Estudante, aproveite a sua experiência para aumentar as competências em Projeto de Software e a Gestão de Projetos; quando você tem referências anteriores (sejam pessoais ou profissionais, sejam recentes ou antigas), o aprendizado será maior. Inclusive, você poderá compreender como as novas técnicas podem ser incorporadas às metodologias tradicionais que podem potencializar soluções certeiras.

Equipe do projeto e matriz de responsabilidades

As ações feitas no desenvolvimento de projeto de software são basicamente realizadas por pessoas: profissionais de TI, especialistas das mais diversas áreas e experiências, empreendedores, gestores, usuários de aplicativos, público, entre outros.

PROJETO DE SOFTWARE

Estudante, como você poderia montar um time para o seu projeto? E as pessoas que não são do seu time podem afetar no desenvolvimento de software.

Bem, o foco aqui é utilizar técnicas e instrumentos para distribuição das atribuições relacionadas ao projeto de software. Identificar o papel dessas pessoas no time do projeto ou colaboradores com requisitos funcionais, ou não funcionais, bem como restrições e premissas importantes que farão do software útil, importante e de sucesso.

Devemos considerar que existem fatores humanos em um projeto que precisam ser gerenciados. Cada pessoa tem suas particularidades: apetites, aspirações, objetivos profissionais, relacionamentos, temperamentos e caráter. Essas características podem se destacar e interferir nos projetos de forma positiva ou negativa. Aqui visamos analisar a formalização da gestão de projetos e das estruturas de ferramentas, defini-las e explicar responsabilidades.

Bloco 1

Matriz de habilidade e responsabilidade	Habilidades	
	Técnicas	Interpessoais
Nome da pessoa	Engenheiro civil	Web designer
Manoel Perfecto	X	X

Bloco 2

Habilidades	
Interpessoais	
Criativo	Agregador

Quadro 1 | Modelo de matriz de habilidade e responsabilidade. Fonte: adaptada de Menezes (2018, p. 118).

A partir da matriz elaborada, todas escolhas e decisões terão grandes chances de acerto, pois o projeto receberá os resultados adequados conforme as habilidades técnicas e interpessoais de cada membro do projeto. A figura 2 mostra (um exemplo) de matriz de habilidade e responsabilidade que os responsáveis podem gerar e adequar conforme a natureza, extensão e orçamento de cada projeto de software.

As diferenças entre os indivíduos se tornam mais complexas quando se inicia a interação entre eles, contudo a gestão de *stakeholders* é possível quando se trabalha equilibrando três fatores, segundo Menezes (2018, p. 118), aumentando ou diminuindo a intensidade conforme as diversas situações em que o projeto passa:

PROJETO DE SOFTWARE

Autoridade: é definida pela posição que a pessoa ocupa na organização e, principalmente, pelo poder que lhe é conferido e como ele é exercido na condução das atividades e na delegação.

Comunicação: refere-se aos recursos e mecanismos disponibilizados para que as informações possam fluir e as decisões possam ser tomadas na organização. Outro traço da comunicação está no grau de formalidade ou de informalidade existente.

Atividade: mostra como é feita a divisão do trabalho na organização, qual sua abrangência, a quantidade, a forma de ser conduzida, a sequência e os instrumentos fornecidos.

Com o time de projeto organizado e os demais *stakeholders* também planejados, você está muito mais preparado para engendar no seu projeto de software.

Porém, ainda existem outros fatores e providências que você precisará conhecer, caso necessite usar para garantir o sucesso do projeto de software.

Siga em Frente...

Problemas na gestão de projetos, conflitos no projeto

Ao mesmo tempo que fatores, características e elementos fundamentais para o bom desenvolvimento de projeto de software são conhecidas, adaptadas e melhoradas e aplicadas, você precisa conhecer alguns problemas já reconhecidos e experimentados para evitar ou saber investigar e dirimir no menor tempo possível. Ou seja, caso aconteça algo inconveniente ou indesejado no seu projeto, o prejuízo deve ser contido o mais breve possível.

Por falar em resolver situações adversas, mediante investigação, o fator imprescindível é ter dados e informações que o interessado consiga pesquisar segundo Cavalcanti (2016 p. 215) a “alimentação de uma base histórica, deve-se realizar ainda uma ou mais sessões para o registro de lições aprendidas, envolvendo o GP e a equipe”.

Ah, estudante, nunca pense que problemas não existirão ou que conflitos são coisas que acontecem somente nos projetos de outros! Então, continuemos com mais situações que poderão servir de lições para você em seu projeto.

Podemos lançar mão de um fluxo simples, mas que poderá ser eficiente para estancar com rapidez as consequências indesejadas, acompanhe a Figura 2.



Figura 2 | Caminho básico para resolver conflitos. Fonte: elaborada pelo autor.

As consequências de um conflito são imprevisíveis quando não se sabe a causa e se não houver tratamento em tempo hábil. Em geral, os problemas podem ser resolvidos pelos próprios envolvidos quando são simples, no entanto, em alguns casos pode ser necessário contar com mediadores ou facilitadores (do próprio time ou externos) dependendo do grau do desgaste presente. Inclusive, um problema relativamente pequeno, quando não solucionado, poderá crescer até causar a suspensão do projeto inteiro.

Você deve evitar em seu projeto a todo custo o seu cancelamento, seja no início ou ao final dele. As metodologias que colaboram naturalmente a “correção” de percurso do projeto são as incrementais ou evolutivas ao possuírem gatilhos nos quais os principais envolvidos avaliam os resultados até então conquistados e tratam também os problemas percebidos desde a última avaliação.

Agora você comprehende que um projeto pode ter problemas, mas também sabe que é possível tratá-los de maneira padronizada ou negociada, facilitando a manutenção da continuidade, resolvendo os conflitos e corrigindo a rota, ajustando o plano inicial, sempre visando o sucesso do seu projeto de software.

Vamos Exercitar?

Se você estava curioso para conhecer as possibilidades de conduzir um projeto de software para entregar tudo o que prometeu ao seu cliente/usuário, acreditamos que esteja bem mais tranquilo. Certo que você está no início da disciplina, então estudará outras técnicas e conceitos que darão a você a segurança necessária para o bom desenvolvimento de um software.

Faça uma revisão do assunto tratado e pense na possibilidade de gerir um projeto, ou pelo menos ser o protagonista das ações necessárias para a sua boa gestão, identificação de *stakeholders* e seus papéis. Por fim, preparar esses envolvidos a resolver todo e qualquer percalço que apareça durante a laboração das tarefas no desenvolvimento de software.

Lembre-se: com a gestão do projeto você poderá dimensionar necessidades de recursos, prazos e pessoas envolvidas para liderar e gerenciar o desenvolvimento das atividades ao desfecho de aceite e satisfação pelo software entregue.

Saiba mais

1. As entregas sucessivas aumentam as chances de sucesso do projeto

Conheça o processo iterativo, aquele que determina espaços temporais (*timeboxing*) que invoca gestores e demais *stakeholders*, analisar o que foi empreendido e validar se ficou consoante o plano, como também avaliar a conjuntura para providenciar os ajustes necessários para continuar o projeto.

PROJETO DE SOFTWARE

Para tanto, sugerimos a leitura do trecho 6.3 Processo Iterativo e Incremental, página no livro [Gestão ágil de projetos](#), de Robson Camargo, Thomaz Ribas (2019), disponível na Biblioteca Virtual.

2. Planejamento da gestão de pessoas

A leitura proposta demonstra recursos em planejamento e liderança para o desenvolvimento adequado do time de desenvolvimento de software, páginas 108 a 120.

Fonte: CAVALCANTI, F. R. P. **Fundamentos de gestão de projetos**: gestão de riscos. São Paulo: Atlas, 2016.

A partir desse estudo, você compreenderá que um projeto poderá ser desenvolvido com maior chance de sucesso, sempre focado no objetivo final do software projetado, mas entregando a cada ciclo/intervalo, parte do software útil e validável.

Estudante, continue estudando e pesquisando, você encontrará alternativas e técnicas em gestão de projetos e liderança no gerenciamento de atividades para o seu projeto de software.

Referências

CAMARGO, R.; RIBAS, T. **Gestão ágil de projetos**. São Paulo: Saraiva Educação, 2019. [Minha Biblioteca]

CAVALCANTI, F. R. P. **Fundamentos de gestão de projetos**: gestão de riscos. São Paulo: Atlas, 2016. [Minha Biblioteca]

MAXIMIANO, A. C. A.; VERONEZE, Fernando. **Gestão de projetos**: preditiva, ágil e estratégica. Barueri: Atlas, 2022. [Minha Biblioteca]

MENEZES, L. C. M. **Gestão de projetos**. 4. ed. São Paulo: Atlas, 2018. [Minha Biblioteca]

Aula 2

Ciclo de Vida do Projeto de Software

Ciclo de Vida do Projeto de Software

Este conteúdo é um vídeo!



Para assistir este conteúdo é necessário que você acesse o AVA pelo computador ou pelo aplicativo. Você pode baixar os vídeos direto no aplicativo para assistir mesmo sem conexão à internet.

Dica para você

Aproveite o acesso para baixar os slides do vídeo, isso pode deixar sua aprendizagem ainda mais completa.

Estudante, esta videoaula foi preparada especialmente para você. Nela, você irá aprender conteúdos importantes para a sua formação profissional.

Vamos assisti-la? Bons estudos!

Ponto de Partida

Saber o grau de importância de compreender o que é um ciclo de vida de projeto é também saber que existe uma sequência de passos para chegar ao objetivo final, a entrega de um software com benefícios percebidos. Essa sequência de tarefas a serem realizadas é a determinação da fase inicial, intermediária até a final; desde o momento da ideia do sistema, até quando se faz a entrega, tendo o software desenvolvido e validado.

Pois bem, você, estudante, já se imaginou estar em frente ao novo empreendimento em que te disseram que desejam um software (com várias funcionalidades) e que precisará entregar pronto (em um prazo curto) para garantir que a empresa sobreviva à grande oscilação financeira? É o momento de saber selecionar o ciclo de vida de projeto mais adequado para este desenvolvimento. Para tanto, você necessitará reconhecer as características entre os tipos de ciclo de vida e suas vantagens.

Agora que você está diante desse desafio, desfrute do conteúdo explanado nessa aula e se capacite para estar apto a elaborar um plano de desenvolvimento de software adotando o ciclo de vida adequado numa situação real da sua vida profissional, em sua empresa ou numa equipe virtual de projeto de software. Desejamos uma boa aula!

Vamos Começar!

Estudante, boas-vindas à aula sobre ciclo de vida! Queremos que você comprehenda e justifique a adoção de um determinado tipo de ciclo de vida de projeto, portanto, pedimos atenção total quanto à relevância no uso desse conceito no desenvolvimento de software. A partir desse conceito é que você poderá ter o domínio para o bom gerenciamento de um projeto de software.

Ciclo de vida clássico

Entenda ciclo de vida recuperando de sua memória de como é feito uma comida usando uma receita. Isso mesmo! Vamos lá: inicialmente, você tem em mente (ou em foto) a comida que gostaria de entregar ou comer; a primeira providência é escolher uma receita e os ingredientes; ao começar a cozinhar/preparar você deverá seguir uma receita e pode mudar alguns detalhes do seu preparo; você vai escolher os materiais e recursos (faca, panela, fogo/forno, liquidificador, entre outros); preparar a comida seguindo o passo a passo do modo de fazer; após pronto, você vai colocar à mesa, ou entregar numa embalagem. Ao final, é importante saber se a comida foi apreciada, não é mesmo?!

Em projeto de software, também é obrigatório ter a descrição de tarefas em uma determinada sequência de execução, em que um procedimento precisa ser executado antes de outros sucessivamente. Desta maneira, estabeleceu-se nomenclaturas clássicas para definir o ciclo de vida (PMI, 2017), praticadas ao longo do tempo entre os gestores e equipes de gerenciamento de projetos: iniciação, planejamento, execução e finalização. Essa denominação é vantajosa por conta da facilidade na comunicação entre os envolvidos e no acompanhamento e monitoramento das atividades que estão pré-estabelecidas para cada fase do ciclo de vida do projeto de software.

A Figura 1 ilustra como podemos compreender um ciclo de vida clássico, em que as atividades são desenvolvidas, semelhante ao que se percebe no processo de preparo de uma receita, porém representada em 4 fases distintas.



Figura 1 | Ciclo de vida clássico. Fonte: elaborada pelo autor.

As tarefas a serem laboradas em cada fase do ciclo de vida no projeto de software estão relacionadas com o uso das disciplinas próprias do processo de construção de um software, tais como: análise e elicitação de requisito, projeto e implementação, análise e modelagem de dados, teste e validação do software e assim por diante.

Na primeira fase, com o nome sugestivo, iniciação, são definidos o escopo do produto e o escopo do projeto, na qual se obtém a permissão para iniciar. A importância nessa fase é ter os objetivos alinhados às operações que devem agregar valor ao negócio da organização.

Na fase seguinte, os principais *stakeholders* se comprometem em participar para alcançar os objetivos definidos, cumprindo com as premissas e restrições de projeto e de produto,. Inclusive é na fase de planejamento que é definido como a gestão do projeto irá monitorar e controlar atividades, recursos e pessoas durante toda a sua execução.

Bem, como você percebe, na Execução é a fase qual tudo acontece e se detalha, se cria, se constrói, valida e faz a entrega do software. Esta é a fase em que os profissionais de TI são mais exigidos; além do indispensável registro de todos os passos realizados para que o

PROJETO DE SOFTWARE

monitoramento/controle do projeto seja realizado e, quando inevitável, se faz a depuração/investigação de causas para analisar os impactos indesejados e problemas detectados.

Depois que as tarefas essenciais de produção e entrega do software são concluídas, a última fase do ciclo de vida é a Finalização. Tão importante quanto as demais fases, é o momento de obter um aceite formalizado de quem usufruirá dos benefícios do software, bem como criar a relação de melhorias, pela experiência vivenciada ao longo das demais fases, para serem aproveitadas em novos projetos, ou naqueles que estão em andamento.

Até aqui você percebeu ser possível elaborar um plano de projeto de software com foco no desenvolvimento de uma aplicação, certo?

Contudo, como gestor ou como integrante de um time de projeto de software, você precisará conseguir identificar, comparar e justificar o uso das variações de ciclo de vida conforme o grau de incerteza dos requisitos funcionais, não funcionais e de incerteza dos recursos tecnológicos; e, mais ainda, em função do grau da frequência de entrega ou de mudanças em requisitos.

Parece incompreensível, certo? Porém, você perceberá que faz muito sentido conhecer e adotar um ciclo de vida conforme a conjunção exige.

Siga em Frente...

Ciclo de vida preditivo e adaptativo

Você perceberá que não existe um ciclo de vida correto e outro incorreto, mas terá capacidade de comparar e justificar a sua escolha por um dos tipos: preditivo ou adaptativo.

O **preditivo** tem como característica a execução de todas as tarefas planejadas para cada fase. Exemplo: todas as necessidades de projeto (recursos, informações, pessoas, entre outras) são planejadas na fase de planejamento; se começar a fase de execução, entende-se que todas as funcionalidades do software estão devidamente esclarecidas e os requisitos definidos, sem a necessidade de replanejamento. A fase de finalização é quando todo o software é entregue em uma única vez. O andamento do projeto é realizado linearmente, da iniciação à finalização. Adota-se esse método quando existe baixo grau de mudanças e que o usuário pode aguardar, ou prefere a entrega do software completo.

Outra maneira de compreender o ciclo de vida preditivo está no tempo de desenvolvimento do projeto. Supondo que o projeto esteja previsto para concluir em 4 meses, sua execução ocorrerá da seguinte maneira: 1 semana para a iniciação; 3 semanas para o planejamento; 11 semanas para a execução e 1 semana para finalização.

O método **adaptativo** é bastante procurado na atualidade em função da possibilidade de entregas constantes, em curíssimo prazo, de funcionalidades novas e úteis aos negócios da organização.

Naquelas circunstâncias em que o nível de demanda é alto e de complexidade exige grande interação com o grupo operacional, especialistas de negócio fazem parte das definições de requisitos, cenário propício quando há requisitos voláteis, ou seja, aqueles sujeitos a mudanças constantes, é recomendável adotar o modelo adaptativo. Esse método é reconhecido por ágil, possui vantagens em relação aos demais quando a realidade é dinâmica e o comprometimento com a entrega constante seja o foco do projeto de software.

Em projetos que o método adaptativo é o escolhido, é comum que os *stakeholders* fiquem vivenciando intensamente a rotina com o time de TI, seja nas decisões ou esclarecimento de dúvidas de regras de negócios, ou na escolha das tecnologias, ou de serviços que integram ao sistema.

Ciclo de vida incremental e iterativo

O método **incremental** planeja o desenvolvimento do projeto em blocos e permite que o cliente do projeto usufrua do sistema à medida que as entregas planejadas sejam realizadas.

O projeto é concluído assim que todos os incrementos forem entregues, ou seja, o software total é representado por todas as entregas do projeto. É característico em ambiente operacional em que se exigem entregas planejadas com baixo grau de mudanças de requisitos.

Faz-se uso dessa modalidade quando a equipe de TI cogita desenvolver as funcionalidades do sistema previamente definidas, porém apenas o detalhamento das necessidades específicas do usuário é realizado incrementalmente, no decurso do projeto de software.

No método **iterativo**, quando o cenário exige precisão do software para sistematizar regras de negócios altamente complexas, no qual o time de TI está implementando parte dos processos operacionais de nível específico, é necessário que a participação do time de operações esteja presente em interações, até diárias.

A escolha do modelo iterativo é porque é inútil arriscar implementar códigos indesejados por falta de detalhamento durante a construção do software. Aqui nesse método, os times de operações, comumente, ficam alocados fisicamente junto ao ambiente de TI, fazendo parte dessa interação que proporcionará resultados desejados quando a entrega é realizada, sem a criação de documentações detalhadas de requisitos.

No decurso do projeto de software iterativo, a prática da prototipação é bastante utilizada, facilitando para o usuário experimentar o comportamento do software e o ensaio com novas tecnologias.

Para o desenvolvimento de sistemas que exigem inovação, pode ser adotado o modelo iterativo, no qual boa parte do projeto ocorrerá a empiricismo com tecnologias ou técnicas variadas para encontrar uma solução adequada à realidade do momento para a organização.

Vamos Exercitar?

Veja você no comando de um projeto de alto nível, em que o ambiente exige um sistema frequentemente atualizado à realidade de mercado; ainda, requer entregas de funcionalidade com periodicidade curta.

Nesta conjuntura, vale lembrar que o método escolhido proporcionará ao time estar capacitado rapidamente, atingindo maturidade a curto prazo. Aliás, essa identificação facilitará muito a todos os *stakeholders* com o envolvimento na velocidade e intensidade necessárias ao projeto de software.

Faça a consideração necessária quanto à Tabela 1 que demonstra sucintamente os diferentes ciclos de vida possíveis para o gerenciamento de um projeto:

Ciclo de vida	Grau de mudanças	Frequência de entrega
Preditivo	Baixo	Baixa
Incremental	Baixo	Alta
Iterativo	Alto	Baixa
Adaptativo	Alto	Alta

Tabela 1 | Diferenças entre os ciclos de vida. Fonte: elaborada pelo autor.

O ciclo de vida preditivo também é chamado de waterfall (cascata) largamente praticado em épocas que os sistemas não demandavam mudanças rápidas. Diferentemente do atual momento em que a tendência é se aproximar das pessoas/usuários de sistemas cada vez mais necessitados de aplicativos adaptáveis à rotina, cujo intuito é permitir um rápido aprendizado; por isso que o método ágil está sendo largamente praticado a partir dos anos 2000.

Saiba mais

1. Importância do *feedback* em projeto de software

Entenda a prototipação como uma poderosa ferramenta na qual o time de TI juntamente com o time de operações poderá criar procedimentos a fim de alcançar resultados inovadores. Leia o item que detalha Prototipação e Teste, na página 49 a 51, do livro: [*Gestão ágil de projetos*](#), dos autores CAMARGO, R.; RIBAS, T. São Paulo: Saraiva Educação, 2019.

Estudante, reflita sobre o método adaptativo quando se reflete que a prototipação pode ser um recurso usado; compreenda a importância do *feedback* no processo de desenvolvimento de software.

PROJETO DE SOFTWARE

2. Uma abordagem adaptativa aplicada em sistema da saúde

Compreenda sobre o ciclo curto e a adaptação do modelo ágil, lendo o item 2.1.3 Abordagem adaptativa em projetos no artigo: FONSECA, N. R. et al. [Gestão de projetos ágeis: um levantamento bibliométrico entre os anos de 2017 e 2021](#). Revista Tecnológica da Universidade Santa Úrsula. 2023-1. Universidade Santa Úrsula, v. 6, n. 1, 2023.

3. Comparação entre o ciclo de vida preditivo e adaptativo

Compare a consideração da abordagem preditiva em projetos, lendo o item 2.1.2 Abordagem preditiva em projetos. Você perceberá as diferenças entre os métodos de desenvolvimento de software, conforme o artigo FONSECA, N. R. et al. [Gestão de projetos ágeis: um levantamento bibliométrico entre os anos de 2017 e 2021](#). Revista Tecnológica da Universidade Santa Úrsula. 2023-1. Universidade Santa Úrsula, v. 6, n. 1, 2023.

Referências

CAMARGO, R.; RIBAS, T. **Gestão ágil de projetos**. São Paulo: Saraiva Educação, 2019. [Minha Biblioteca]

IFRAIM FILHO, R.; CIERCO, A. A. **Governança, ESG e Estrutura Organizacional**. São Paulo: Actual, 2022. Disponível em: <https://integrada.minhabiblioteca.com.br/#/books/9786587019536/> . Acesso em: 28 set. 2023. [Minha Biblioteca]

MAXIMIANO, A. C. A.; VERONEZE, F. **Gestão de projetos: preditiva, ágil e estratégica**. Barueri: Atlas, 2022. [Minha Biblioteca]

Aula 3

Processos de projeto de Software

Processos de projeto de Software

Este conteúdo é um vídeo!



Para assistir este conteúdo é necessário que você acesse o AVA pelo computador ou pelo aplicativo. Você pode baixar os vídeos direto no aplicativo para assistir mesmo sem conexão à internet.

Dica para você

Aproveite o acesso para baixar os slides do vídeo, isso pode deixar sua aprendizagem ainda mais completa.

Estudante, esta videoaula foi preparada especialmente para você. Nela, você irá aprender conteúdos importantes para a sua formação profissional. Vamos assisti-la? Bons estudos!

Ponto de Partida

Estudante, aqui você é quem reconhecerá como os processos podem auxiliar no projeto de software. Ao conhecer a história BDParty, que é comum a muitos brasileiros, você associará cada tarefa/atividade a ser desenvolvida na BDParty conforme os processos de um projeto.

Se você está pensando que uma festa de aniversário é diferente de um projeto, relembraremos que toda festa é inédita (acontecerá diferentemente por conta do local, das pessoas, dos momentos de pessoa e das circunstâncias), os recursos são distintos em função da intervenção das pessoas na montagem e no preparo, ou da cozinha.

Enfim, assim que você estudar os processos envolvidos em projeto de software, temos certeza de que as características de projeto serão avistadas por você também nas festas de aniversário!

Vamos Começar!

Vale lembrá-lo de que conhecer os processos de gestão de projeto te deixará ao lado dos melhores profissionais de TI no mercado de trabalho, o conceito abordado vai te deixar familiarizado com o que está sendo praticado atualmente. Por isso, fique atento para refletir na sua prática as competências em processos para você aumentar a sua habilidade como profissional de TI.

Processo de iniciação e planejamento

Começando, vamos te mostrar a **iniciação**, uma etapa na qual os projetos são divididos em outros menores, visando aumentar a capacidade de gestão de projetos complexos ou megaprojetos. Com esse desmembramento, os *stakeholders* decidem pela continuidade ou não para a etapa seguinte, ou postergar quando se percebe que o projeto é inviável para o momento, inclusive podem ter outros projetos mais importantes para serem aprovados.

Em uma decisão complexa que envolve sempre muito risco, os gestores da empresa e a equipe de gestão do projeto dependem de informações confiáveis, atualizadas e reais. O TAP (termo de abertura do projeto) te ajudará reunir esses insumos, com premissas e restrições, valores de

PROJETO DE SOFTWARE

orçamento, prazo e demais recursos; pelo TAP, ilustrado na Tabela 1, você que obterá uma autorização formal dos responsáveis, *stakeholders* e patrocinadores.

Exemplo de termo de abertura do projeto (TAP)	
Gestor do projeto	Normalmente definido pelos patrocinadores (<i>sponsors</i>).
Principais pontos	Todos os envolvidos estarão informados sobre quando, de que forma e por quem deverão ser entregues as atividades, auxiliando no processo de comunicação.
Justificativa	Ter o termômetro que indica quais projetos ou etapas devem ser executados primeiro e quais devem aguardar o melhor momento para serem inseridos na gestão.
Restrições e premissas	As limitações de custos, a disponibilidade de profissionais e a infraestrutura da empresa onde será implantado o sistema deverão ser considerados antes e durante todo o planejamento das etapas do projeto, inclusive o prazo final para a conclusão do projeto.
Escopo	Devem ser descritos todos os requisitos solicitados pelos <i>stakeholders</i> de forma clara e completa, evitando-se questionamentos futuros nas entregas, exceto se o produto não atender rigorosamente ao solicitado ou definido em padrões, ou referências de qualidade.

Tabela 1 | Termo de Abertura do Projeto. Fonte: elaborada pelo autor.

Várias maneiras para descrever o TAP podem utilizadas, busque uma alternativa que seja objetiva, sucinta e completa, e rápida para elaborar a sua. Conheça outros modelos, exemplo no site [Sebrae Canvas](#).

Na fase de **planejamento** você pode usar o conceito de escopo de projeto e escopo de produto, isso facilitará na descrição e escolha das pessoas responsáveis pelas tarefas.

O escopo do projeto descreve todo o trabalho necessário para a realização da entrega do produto de software; quais processos e as atividades de cada um deles serão executadas, o fluxo de sequência dessas atividades e prazos. O plano das atividades do projeto estruturado adequadamente proporcionará a entrega do produto de software, conforme desejado.

Enquanto o escopo do produto define as características que o produto precisa ter, dessa forma, é feita uma descrição detalhada de tudo o que o cliente solicitou e de possíveis alterações feitas pelos *stakeholders*, considerando os requisitos funcionais e não funcionais; sempre procurando atender às necessidades e expectativas dos interessados.

Siga em Frente...

Processo de execução e controle

A fase de **execução** é a maior das fases e nela que tudo acontece em relação à produção do software. É certo que nessa fase o gestor dispensa menos esforços, porém concomitantemente o esforço dele é enorme em função do processo de **controle**. Os dois processos são os que exigem dos *stakeholders* comprometimento total, tanto na realização das atividades planejadas quanto na resolução de problemas quando ocorrem.

Na execução é que acontece o *design* de projeto e implementação de código-fonte com as devidas tarefas de integração, gerenciamento de configuração, refatoração, análise e construção do banco de dados, documentações e testes.

Nesse momento, a equipe vai se mobilizar para realizar as atividades previstas, porém devem ficar atentos em função de possíveis fatos indesejados, aqueles imprevistos. Neste caso, é crucial a análise do que os causou e de como retomar o curso para o que foi planejado.

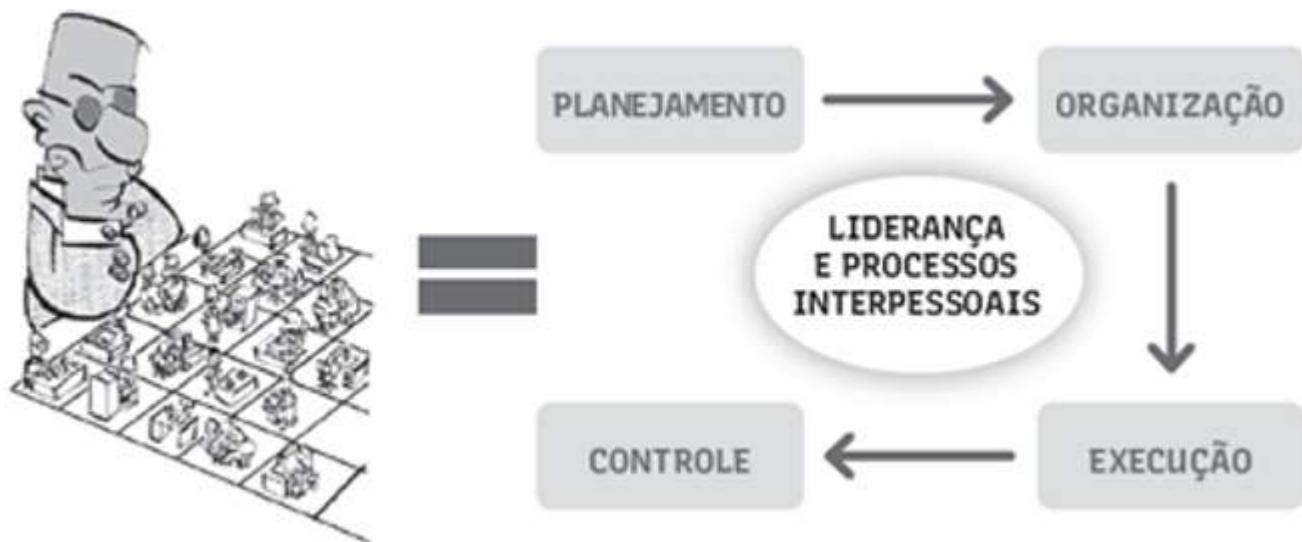


Figura 1 | Funções do processo gerencial. Fonte: Maximiano (2022, p. 11).

A Figura 1 ilustra bem a atuação do gestor, sempre monitorando as atitudes, fazeres dos profissionais de TI, interpelações de outros profissionais e intercorrências. A responsabilidade do **controle** do projeto é do gestor, sempre promovendo o adequado funcionamento dos recursos, inclusive suceder à comunicação do estado do projeto aos interessados conforme planejado.

Processo de encerramento e avaliação

Nesse processo, deixar documentado os acertos e erros é extremamente relevante, de modo que as lições aprendidas sejam usadas como referência em próximos projetos. Verifica-se os resultados obtidos para concluir se o escopo do projeto foi alcançado e se as especificações do produto foram atendidas.

No entanto, o ponto crítico está na aceitação do produto ou serviço finalizado, pelo cliente final ou patrocinador. Avaliar se as funcionalidades foram alcançadas, ou seja, se os requisitos funcionais e não funcionais foram implementados adequadamente e conforme as necessidades de negócio.

A formalização do encerramento é obrigatória, da mesma forma que na fase de iniciação foi dado aceite para começar o projeto.

Estudante de Projeto de Software, acreditamos que você já está preparado para identificar uma metodologia de Processos e a importância deles.

Bons estudos!

Vamos Exercitar?

A facilidade verificada ao contar com os Processos ou Grupo de Processos em Projeto de Software está em avisar as pessoas envolvidas e obter o comprometimento com o projeto. Em situações reais, existem casos em que dezenas de profissionais de TI estão trabalhando no mesmo projeto (presencial ou remotamente), atuando por vários meses ou anos. Nessas circunstâncias, o gestor deve exercer o papel de liderança mantendo todos motivados e realizando suas atribuições, entregando o melhor dos resultados, em cada tarefa do projeto.

Sendo assim, as distribuições em processos deixam mais evidentes em que momento as atividades deverão ser executadas. Outro aspecto relevante é a visão clara de progresso do projeto, que por sua vez possibilita uma comunicação acertada a todos os *stakeholders*.

Deixamos aqui a Tabela 2, com a ilustração de como os processos podem ser implementados para o caso BDParty.

Característica de projeto	O resultado é único e divertido, é temporário porque acontecerá apenas uma vez e tem um escopo bem definido: para comemorar e ser lembrado!
Iniciação	O dia da festa será dd/mm às hh:mm para o Melhor Estudante Premissas e restrições:

	<ul style="list-style-type: none"> • Tipo de festa: para amigos e familiares, festa surpresa. • Valores: do tamanho do bolso. • Aquisições: comes e bebes, decoração. • Qualidade do que será servido. • Responsável pelas fotos e pelos vídeos. • Recuperar da memória: <i>feedback</i> da festa anterior.
Planejamento	<p>Das pessoas: maior comprometimento possível (Suzuki, João, Akemi e Hideo), da responsabilidade de cada um.</p> <p>Compras (João): o melhor produto e serviço que puder comprar.</p> <p>Local (Hideo): reservar</p>
Execução	<p>Comprar (João), contratar (Akemi), locar espaço (Hideo), programar entregas (João), avisar os responsáveis (Suzuki)</p> <p>Sorrir, receber convidados, saudar o aniversariante, comer, beber, cantar, fotografar, conversar, contar história, desejar alegria, se divertir</p>
Monitoramento e controle	<p>(Suzuki) Está faltando alegria? Está faltando comida/bebida? Fez a foto oficial? Tem pratinho para comer o bolo?</p>
Encerramento e Avaliação	<p>O aniversariante se divertiu? Todos estão satisfeitos? O que faltou e que podemos anotar para o próximo? As contas foram pagas?</p> <p>Divulgar registros, fotos, vídeos, depoimentos. Agradecer cada pela participação e organização.</p>

Tabela 2 | Processos do BDParty. Fonte: elaborada pelo autor.

Você percebeu como é fácil distribuir as atividades pelos processos ou grupo de processos, agora faça uma reflexão para outros cenários em sua realidade profissional.

Bons estudos!

PROJETO DE SOFTWARE

Saiba mais

Estudante, chegou o momento para você aprofundar no seu aprendizado. Deixamos aqui alguns tópicos importantes sobre essa aula:

1. Roteiro para a gestão do projeto

Leia na página 12 o item que descreve como o gestor de projeto tomará decisões, ao longo dos processos planejar, organizar, executar e encerrar o projeto.

MAXIMIANO, A. C. A.; VERONEZE, F. [Gestão de projetos: preditiva, ágil e estratégica](#). Barueri: Atlas, 2022. [Minha Biblioteca]

2. Processo para o Scrum

Estudante, leia o item 8.5.2 Scrum Master, na página 148, e aprofunde sobre processos compreendendo o papel o Scrum Master em um dos métodos ágeis. A importância do gestor/líder, por ser quem deve fazer cumprir os eventos e a manutenção dos artefatos da metodologia.

CAMARGO, R.; RIBAS, T. [Gestão ágil de projetos](#). São Paulo: Saraiva Educação, 2019. [Minha Biblioteca]

3. Avaliar as atividades de projeto no Scrum

Leia na página 164 o item 8.8.5 Retrospectiva da Sprint. É parte fundamental a reunião de retrospectiva, como oportunidade que o time tem para avaliar como estão trabalhando juntos, a cada iteração do projeto.

CAMARGO, R.; RIBAS, T. [Gestão ágil de projetos](#). São Paulo: Saraiva Educação, 2019. [Minha Biblioteca]

Estudante, continue se capacitando, pois a sua competência depende do esforço no aprendizado. Através dos estudos estará mais bem preparado para realizar suas atividades profissionais com excelência.

Referências

CAMARGO, R.; RIBAS, T. [Gestão ágil de projetos](#). São Paulo: Saraiva Educação, 2019. [Minha Biblioteca]

FILHO, W. P. P. [Engenharia de Software - Projetos e Processos](#), v. 2, 4 ed. Rio de Janeiro: LTC, 2019. [Minha Biblioteca]

MAXIMIANO, A. C. A.; VERONEZE, F. **Gestão de projetos**: preditiva, ágil e estratégica. Barueri: Atlas, 2022. [Minha Biblioteca]

WYSOCKI, R. K.; MARQUES, A. S. **Gestão eficaz de projetos**: o ambiente organizacional de gerenciamento de projetos. São Paulo: Saraiva Educação, 2020. [Minha Biblioteca]

Aula 4

Arquitetura de Software

Arquitetura de Software

Este conteúdo é um vídeo!



Para assistir este conteúdo é necessário que você acesse o AVA pelo computador ou pelo aplicativo. Você pode baixar os vídeos direto no aplicativo para assistir mesmo sem conexão à internet.

Dica para você

Aproveite o acesso para baixar os slides do vídeo, isso pode deixar sua aprendizagem ainda mais completa.

Estudante, esta videoaula foi preparada especialmente para você. Nela, você irá aprender conteúdos importantes para a sua formação profissional. Vamos assisti-la? Bons estudos!

Ponto de Partida

O impacto da Arquitetura de Software no processo de desenvolvimento está relacionado com o nível de conhecimento e habilidade dos profissionais de TI. Por sua vez, tem uma relação direta no prazo do projeto, na escolha da tecnologia, no entendimento das regras de negócio com o grupo operacional e decisões das funcionalidades do sistema de software.

Vejamos: o sistema faz análise de uma coleção de dados de natureza variada coletados ao longo do tempo e em vários locais. Esse processo de análise tem como base histórica para indicar alternativas para certas operações ou providências táticas de tratamento no setor agrícola. Estamos diante de um Sistema de Monitoramento na Produção Agrícola; e para facilitar chamaremos de SMPA e você é um dos responsáveis pela construção e implantação do software.

PROJETO DE SOFTWARE

Estudante, agora é compreender o que é arquitetura de software, estilos e padrões para que você possa tomar decisões e proporcionar o melhor de um projeto de software nesse segmento de mercado que cresce a cada safra. Que tal?!

Vamos Começar!

Boas-vindas a mais um conteúdo importante, um dos conceitos relevantes na decisão de um projeto de software. A partir dessa competência, você será um profissional de TI diferenciado no setor de desenvolvimento de software. Numa analogia com a construção de um edifício, a Arquitetura é mais que a parte física de como as paredes, pisos, telhados e cores são apresentados, é como cada parte do edifício se integra, desde a entrada até os diversos espaços/ambientes, de vários tamanhos, funções específicas ou genéricas; e, ainda, de como os demais edifícios ao redor interagem com esse que está sendo arquitetado.

O que é arquitetura de software

Ao projetar uma Arquitetura de um Software valerá, se pudermos, conforme detalha Pressman (2021, p. 182):

- (1) analisar a efetividade do projeto no atendimento dos requisitos declarados,
- (2) considerar alternativas de arquitetura em um estágio em que fazer mudanças de projeto ainda é relativamente fácil
- (3) reduzir os riscos associados à construção do software.

Alcançar a arquitetura adequada de um software pode levar um tempo, está atrelada a maturidade de como os componentes, ou partes do software, devem interagir; isso dependerá da velocidade no aprofundamento dos negócios que o engenheiro de software terá oportunidade de desenvolver.

Um componente da arquitetura de um software poderá ser um módulo de um sistema, ou uma classe orientada a objetos. Por outro lado, poderá ser aumentado para representar um banco de dados. Ou seja, um componente da arquitetura deve ter as propriedades que possibilitem a interação com os outros componentes do sistema.

Siga em Frente...

Estilos e padrões arquitetônicos

O estilo de arquitetura será construído conforme o software a ser criado para um sistema computacional, e cada estilo descreve uma categoria, de acordo com Pressman (2021, p. 186):

PROJETO DE SOFTWARE

- (1) um conjunto de componentes (p. ex., um banco de dados, módulos computacionais) que realiza uma função exigida por um sistema
- (2) um conjunto de conectores que habilitam a “comunicação, coordenação e cooperação” entre os componentes
- (3) restrições que definem como os componentes podem ser integrados para formar o sistema
- (4) modelos semânticos que permitem a um projetista compreender as propriedades gerais de um sistema por meio da análise das propriedades conhecidas de suas partes constituintes.

Um componente da arquitetura de um software poderá ser um módulo de um sistema, ou uma classe orientada a objetos. Por outro lado, poderá ser aumentado para representar um banco de dados. Ou seja, um componente da arquitetura deve ter as propriedades que possibilitem a interação com os outros componentes do sistema.

Um padrão de arquitetura impõe uma transformação no projeto de arquitetura; uma regra sobre a arquitetura, detalhando como o software tratará algum aspecto de sua funcionalidade em termos de infraestrutura (p. ex., concorrência); tendem a tratar de questões comportamentais específicas no contexto da arquitetura (p. ex., como as aplicações em tempo real tratam a sincronização ou as interrupções).

A estrutura básica de uma arquitetura em camadas é caracterizada quando cada uma realiza operações progressivamente e se tornam mais próximas do conjunto de instruções de máquina; considerada de camada central. À medida que as camadas se expandem para a parte externa, logo após a camada central, aparece a intermediária que fornece serviços utilitários; em seguida, a camada de interface com o sistema operacional e na mais externa, os componentes atendem às operações da interface do usuário.

Decisões sobre a arquitetura de software

Os engenheiros de software experientes têm habilidade para escolher os estilos/padrões de arquitetura, para tanto, eles podem seguir algumas orientações para tomar decisões sobre a arquitetura de software, dentre elas, que devem ser analisadas separadamente, ou em conjunto:

- **Economia:** cuidado com recursos desnecessários, portanto, depende da abstração para evitar a complexidade em demasia.
- **Visibilidade:** a motivação pela arquitetura escolhida deve estar tecnicamente clara e comunicada ao time eficazmente.
- **Espaçamento:** é o grau de separação por interesses, podendo ficar muito fragmentado, com perda de visibilidade.
- **Simetria:** os projetos simétricos são mais fáceis de entender, compreender e comunicar; podendo ser simetria estrutural ou comportamental.
- **Emersão:** comportamento e controle emergentes e auto-organizados frequentemente são o segredo da criação de arquiteturas de software expansíveis, eficientes e econômicas; são

PROJETO DE SOFTWARE

sistemas flexíveis.

O ciclo de vida de um sistema de software requer atenção quanto à manutenção da arquitetura, além da manutenção em funcionalidades, ou correções, ou grau de manutecibilidade. Significa que o projetista, ou o engenheiro de software, tem o desafio de aplicar abstrações na arquitetura, com o potencial de adicionar uma estrutura que melhora a legibilidade e a facilidade de manutenção do código-fonte.

Estudante, você que deseja aprofundar na construção de um determinado padrão arquitetônico de um software deve se dedicar tempo para comunicar a sua escolha a todos do time, isso garantirá que o software se manterá com os mesmos princípios do início do projeto de software.

Vamos Exercitar?

Escolhendo a arquitetura em camadas, para o software SMPA, podemos sugerir uma distribuição da seguinte maneira:

- **Camada de dispositivos IoT:** na qual procede a coleta de dados de sensores e dispositivos IoT espalhados pela fazenda, p. ex. medir temperatura, umidade do solo, níveis de água, qualidade do solo e muito mais.
- **Camada de processamento e análise de dados:** os dados são processados, limpos e analisados para obter informações úteis, podendo conter técnicas de aprendizado de máquina para prever condições e necessidades futuras.
- **Camada de aplicação web:** um aplicativo web acessível a partir de dispositivos como computadores e smartphones. Aqui, os agricultores e gerentes podem acessar informações em tempo real sobre o estado da fazenda, receber alertas e tomar decisões informadas.
- **Camada de armazenamento de dados:** os dados processados são armazenados em um banco de dados que pode ser relacional, NoSQL ou uma combinação de ambos.

Perceba que o estilo de arquitetura exemplificado permite um entendimento claro de como o sistema está desenvolvido. Aqui vale lembrar que na manutenção e evolução do software deve ser mantida essa estrutura, assim garantirá agilidade na resposta para novas funcionalidades.

Lembrando também que um componente importante pode estar previsto: sistema de alerta, pelo qual os alertas e notificações são enviados aos agricultores em tempo real quando ocorrem eventos importantes, como uma previsão de chuva pesada ou problemas com os sensores.

Saiba mais

1. A importância da arquitetura de software na gestão de projetos

PROJETO DE SOFTWARE

Leia os itens: 2.2.3 Arquiteturas de Software baseada em Serviços, 2.3 Computação em Nuvem, 2.4 Adaptabilidade de Software, para reconhecer exemplos de arquitetura de software.

Fonte: SILVA, M. A. P. [Uma arquitetura de software adaptativa baseada em arquétipo OpenEHR para sistemas de informação em saúde](#). Tese. Doutorado em Ciência da Computação. Universidade Federal de Pernambuco, 2022.

2. Estilos de arquitetura

Leia o item 10.3.1 Uma breve taxonomia dos estilos de arquitetura, página 187, na qual você conhecerá alguns estilos de arquitetura são descritos: Centralizadas em dados; Fluxo de dados; Chamada e retorno; Orientados a objeto e Em camadas.

Fonte: PRESSMAN, R. S.; MAXIM, Bruce R. [Engenharia de software](#). Porto Alegre: AMGH, 2021. [Minha Biblioteca]

Referências

FILHO, Wilson de Pádua P. **Engenharia de Software - Projetos e Processos**. v. 2, 4 ed., Rio de Janeiro: LTC, 2019. [Minha Biblioteca]

MAXIMIANO, A. C. A.; VERONEZE, F. **Gestão de projetos**: preditiva, ágil e estratégica. Barueri: Atlas, 2022. [Minha Biblioteca]

PRESSMAN, R. S.; MAXIM, B. R. **Engenharia de software**. Porto Alegre: AMGH, 2021. [Minha Biblioteca]

Aula 5

Encerramento da Unidade

Videoaula de Encerramento

Este conteúdo é um vídeo!



Para assistir este conteúdo é necessário que você acesse o AVA pelo computador ou pelo aplicativo. Você pode baixar os vídeos direto no aplicativo para assistir mesmo sem conexão à internet.

Dica para você

PROJETO DE SOFTWARE

Aproveite o acesso para baixar os slides do vídeo, isso pode deixar sua aprendizagem ainda mais completa.

Estudante, esta videoaula foi preparada especialmente para você. Nela, você irá aprender conteúdos importantes para a sua formação profissional. Vamos assisti-la? Bons estudos!

Ponto de Chegada

Estudante, boas-vindas ao momento em que vamos consolidar a competência da disciplina que é compreender a importância de Projeto de Software, a sua gestão, a escolha de processos adequados à natureza da aplicação e à arquitetura de software no processo de desenvolvimento de software.

Para tanto, você deve revisar os conceitos fundamentais da necessidade de gerenciar um projeto de software, além de caracterizar e delimitar o produto que planeja entregar.

Dando sequência ao nosso estudo, vamos identificar os processos e como as tarefas serão realizadas em cada fase do ciclo de vida, dando ênfase na finalidade da segregação de fases, o que facilitará a comunicação entre os envolvidos no projeto.

É Hora de Praticar!



Este conteúdo é um vídeo!

Para assistir este conteúdo é necessário que você acesse o AVA pelo computador ou pelo aplicativo. Você pode baixar os vídeos direto no aplicativo para assistir mesmo sem conexão à internet.

O diretor de operação da PrOdont aspira agregar um time de TI em seu programa de crescimento da clínica de odontologia, unindo a força e talentos dos serviços atuais com as vantagens dos sistemas modernos.

Os desafios estão no atendimento ao cliente/paciente da PrOdont aprimorando a interação mediante um aplicativo de comunicação, agendamento e histórico de tratamento. Por sua vez, o gerenciamento dos recursos da clínica em decorrência dos atendimentos e tratamentos realizados também deve ser pensado e executado.

PROJETO DE SOFTWARE

Ao conhecer a Arquitetura de Software, você saberá escolher o estilo e padrão adequados, conforme o sistema de software que está desenvolvendo.

Para isso, é fundamental responder as seguintes perguntas:

- A arquitetura de software que estou adotando no meu projeto facilitará a manutenção do software após a implantação, em fase inevitável de evolução do sistema?
- O time de desenvolvimento tem a competência necessária para resolver problemas e conflitos que podem surgir ao longo do Projeto de Software?
- Os processos que compreendem o ciclo de vida do projeto estão perceptíveis a todos do time de TI, facilitando a comunicação e resolução de conflitos ou problemas?

Estudante, compreendamos a aplicação das competências estudadas!

Como início de projeto, você pode detalhar os papéis dos responsáveis pelas áreas de coordenação das atividades do desenvolvimento de software, analistas de negócio e de dados especializados em clínica odontológica. Complemente essas informações com os arquitetos de software com experiência em componentes para serviços/tratamentos da odontologia, designer especializado em UX e o responsável pelos requisitos funcionais pelo time de operações.

Faz-se necessário a escolha de um ciclo de vida adequado à realidade do projeto, para criar soluções ao longo do tempo do desenvolvimento e implantação de software, para compor o sistema da PrOdont. Como exemplo, poderia ser o modelo iterativo prevendo entregas periódicas e muitas mudanças devido às necessidades constantes de regras da clínica.

Quanto à escolha da arquitetura do software, você poderá dar prioridade no aspecto da facilidade em compreender o aspecto estrutural e comportamental. Por exemplo: a simetria na qual os códigos são mais fáceis de entender, compreender e comunicar, a qual pode ter simetria estrutural ou comportamental, visando, assim, funcionalidades de fácil aprendizado pelas diversas áreas da clínica.

Estudante, você deve deixar explícita a necessidade de garantir que o *design* das interfaces serão prioridade, dando ênfase na utilização da aplicação de maneira rápida pelos pacientes. E por parte dos profissionais da clínica, também que terá facilidade nas operações para o registro e consulta dos tratamentos realizados, num fluxo de atividades de acordo com cada paciente.

Sucesso no seu projeto!

1 PROJETO DE SOFTWARE INTRODUÇÃO À GESTÃO DE PROJETOS

RESULTADOS ESPERADOS

- Avaliar os fundamentos da gestão de Projeto de Software
- Escolher ciclo de vida e processos adequados para o desenvolvimento de software
- Explicar como a Arquitetura de Software facilitará o desenvolvimento de software



CICLO DE VIDA

Composta por Atividades metodológicas do Desenvolvimento de Software, compreende-se em fases que contemplam as práticas para iniciar, planejar, construir e entregar o software, de maneira controlada e avaliada; que apoia a comunicação entre os envolvidos



TIME DE TI E RESPONSABILIDADES

O time de TI sempre apto para os desafios do desenvolvimento de software, saberá resolver os conflitos e colocar em prática as determinações acordadas com todos os stakeholders, principalmente com o time de operações



ARQUITETURA DE SOFTWARE

Explanar e obter aceitação da arquitetura de software adequada à natureza dos negócios que o software atenderá, visando um produto adaptável e manutenível nas condições técnicas do time de TI e viável a longo prazo



CONTROLE DA EXECUÇÃO

Medir, analisar e ajustar os resultados esperados conforme o plano do projeto, sempre colaborando para alcançar os objetivos de um produto útil e viável, decidindo por ferramentas e tecnologias adequadas ao Projeto de Software



ENCERRAMENTO E AVALIAÇÃO

Consolidação do software desenvolvido dada pela fase em que todos são comunicados e participam das avaliações de como as atividades do ciclo de vida foram realizadas e de como o time de TI resolveu problemas e conflitos, além da avaliação de satisfação do projeto entregue



Figura 1 | Introdução à gestão de projetos. Fonte: elaborada pelo autor.

PROJETO DE SOFTWARE

FILHO, W. P. P. **Engenharia de software - Projetos e Processos - V. 2, 4 ed.** Rio de Janeiro: LTC, 2019. [Minha Biblioteca]

MAXIMIANO, A. C. A.; VERONEZE, F. **Gestão de projetos: Preditiva, Ágil e Estratégica.** Barueri: Atlas, 2022. [Minha Biblioteca]

PRESSMAN, R. S.; MAXIM, B. R. **Engenharia de software.** Porto Alegre: AMGH, 2021. [Minha Biblioteca]

Unidade 2

Metodologias em Gestão de Projeto de Software

Aula 1

Métodos ágeis em Projeto de Software

Métodos ágeis em Projeto de Software

Este conteúdo é um vídeo!



Para assistir este conteúdo é necessário que você acesse o AVA pelo computador ou pelo aplicativo. Você pode baixar os vídeos direto no aplicativo para assistir mesmo sem conexão à internet.

Dica para você

Aproveite o acesso para baixar os slides do vídeo, isso pode deixar sua aprendizagem ainda mais completa.

Estudante, esta videoaula foi preparada especialmente para você. Nela, você irá aprender conteúdos importantes para a sua formação profissional. Vamos assisti-la?
Bons estudos!

Ponto de Partida

Um grupo de desenvolvedores americanos resolve em 2001 escrever um documento reunindo experiências vividas em que foram percebendo a necessidade de reagir rapidamente às mudanças e que não ficassem reféns de antigos processos e “melhores práticas” enraizadas nos processos da empresa.

O documento concebia, então, um modelo que permitisse às empresas e às equipes ter espaço para experimentar e errar, de modo que aprendessem rapidamente com seus erros para poderem ajustar, crescer e criar produtos e serviços de maneira eficaz. Além do fazer bem-feito, também fazer a coisa certa.

Estudante, queremos trazer para você esse conceito para poder aplicar em situações que façam sentido e demonstrem realmente resultados a partir da metodologia ágil. Imagine que você esteja num time de desenvolvimento de software para o setor de logística (setLOG) do ramo de alimentos no qual os principais coordenadores de setores conhecem suas rotinas, mas estão necessitando repensar as atuais aplicações em software.

Muito bem, esteja preparado para conhecer os conceitos em Manifesto Ágil que vem ganhando novos adeptos na maioria dos times de desenvolvimento de software.

Vamos Começar!

Bem-vindo ao mundo da Metodologia Ágil! Para compreender essa metodologia, você deve partir das necessidades de quem realmente utilizará o sistema, antes mesmo de pensar em qual tecnologia será usada para construir o software (linguagem de programação, ou sistema de banco de dados), até mesmo fazer uma relação direta com experiências anteriores.

Manifesto Ágil

O termo ágil está relacionado diretamente em aprender rapidamente e até mesmo errar rápido. Isso mesmo, parece estranho! Perceba que a ideia é agir em ciclos curtos, quanto mais cedo errar tentando, menor será o custo ou prejuízo. E nesse ciclo de fazer, testar, errar e aprender, as soluções serão melhores e adequadas às pessoas e às organizações.

Lembre-se das atividades no modelo de desenvolvimento de software Waterfall, em que os processos são muito bem definidos (entender, analisar, construir, testar, entregar e manter); eles são eficientes em ambientes estáveis e com pouca incerteza.

Contudo, no mundo moderno, vivemos em um momento em que as mudanças acontecem em velocidade altíssima (em vários lugares ao mesmo tempo), surgindo o novo, deixamos de pensar no velho (volátil). Por isso, as coisas que hoje parecem estar certas, logo passam a não ser tão úteis, ficando difícil avaliar as ameaças (incertas) pela velocidade que as mudanças acontecem. Eventos que antes pareciam distantes passaram a se relacionar rapidamente, dando resultados inesperados (complexos). Ao mesmo tempo que existe diversidade, a ambiguidade se tornou comum, sendo necessário avaliar os cenários e as necessidades sob diferentes aspectos; num ambiente ambíguo, esse contexto gera conflito e até incapacidade de compreensão.

Portanto, estudante, cada vez mais, agir rapidamente para esclarecer, ter habilidade em aprender o novo e alcançar soluções adequadas por meio de ideias colaborativas, maiores são as chances

de desenvolver produtos úteis. Ou seja, continuar pronto para reaprender será melhor para você e todos que os cercam.

Siga em Frente...

Valores da metodologia ágil

Você perceberá que os quatro valores estão inter-relacionados, sempre com o foco em construir softwares atualizados e adequados aos indivíduos:

- **Indivíduos e interações:** por mais que processos sejam importantes para desenvolvimento de uma atividade, antes de tudo é preciso compreender as pessoas, as necessidades delas e como elas funcionam; por meio da alta frequência das interações com os indivíduos, maior é a certeza de que você construirá um produto útil.
- **Produto em funcionamento:** a prática da interação constante com os indivíduos requererá pouca documentação, pois a experimentação das funcionalidades do software, com o próprio software, associadamente com o time de operações, propiciará um software melhor e mais completo. Lembre-se que testar e errar rapidamente, avaliar e corrigir logo em seguida é ser ágil.
- **Colaboração com o cliente:** se antes a entrega do software estava baseada em contrato e nas suas cláusulas tão detalhadas, agora cliente e prestador de serviços, usuários de sistemas e desenvolvedores de software trabalham em colaboração. Esse trabalho conjunto causará menor desgaste ou conflitos, pois todos almejam um produto realmente útil.
- **Responder a mudanças:** só é possível entregar funcionalidades atualizadas numa realidade tão dinâmica e complexa quando as pessoas colaboram continuamente, com interações agregando necessidades correntes. Pois muitas vezes seguimos um plano que já está aquém do necessário ou totalmente diferente da realidade do momento.

Princípios da metodologia ágil

Compreender cada um dos 12 princípios vai te deixar mais próximo da prática ágil, pois será possível perceber que precisamos estar atentos em todos esses princípios para nos conduzir ao desenvolvimento de software com mais rapidez, eficiência e eficácia (Camargo, 2019, p. 103):

- 1 - Nossa maior prioridade é satisfazer o cliente através da entrega contínua e adiantada de software com valor agregado.
- 2 - Mudanças nos requisitos são bem-vindas, mesmo tardiamente no desenvolvimento. Processos ágeis tiram vantagem das mudanças, visando à vantagem competitiva para o cliente.
- 3 - Entregar frequentemente software funcionando, de poucas semanas a poucos meses, com preferência à menor escala de tempo.

PROJETO DE SOFTWARE

4 - Pessoas de negócio e desenvolvedores devem trabalhar diariamente em conjunto por todo o projeto.

5 - Construa projetos em torno de indivíduos motivados. Dê a eles o ambiente e o suporte necessário e confie neles para fazer o trabalho.

6 - O método mais eficiente e eficaz de transmitir informações para e entre uma equipe de desenvolvimento é através de conversa face a face.

7 - Software funcionando é a medida primária de progresso.

8 - Os processos ágeis promovem desenvolvimento sustentável. Os patrocinadores, desenvolvedores e usuários devem ser capazes de manter um ritmo constante indefinidamente.

9 - Contínua atenção a excelência técnica e bom design aumenta a agilidade.

10 - Simplicidade: a arte de maximizar a quantidade de trabalho não realizado é essencial.

11 - As melhores arquiteturas, requisitos e designs emergem de times auto-organizáveis.

12 - Em intervalos regulares, a equipe reflete sobre como se tornar mais eficaz e então refina e ajusta seu comportamento de acordo.

Os princípios ágeis estão descritos para que você possa se capacitar profissionalmente para atender às equipes modernas. Mais que agilidade, os novos times de desenvolvimento de software primam pela colaboração, dos pormenores, disciplina e sustentabilidade.

Vamos Exercitar?

Estudante, você está diante ao projeto de software da setLOG, com a responsabilidade de produzir uma solução esperada pelos clientes: facilidade de acesso às informações; controle de restrição das informações; integração das movimentações dos produtos com as empresas participantes do projeto.

Agir ágil neste projeto é imprescindível para que os clientes da setLOG tenham acesso à programação dos produtos a serem disponibilizados e quando estiverem prontos para coleta. Como se trata de alimentos perecíveis, existe pouquíssima folga entre o “time” do produto na expedição e o momento da coleta.

Bem, todos esses detalhes estão sendo descobertos pelo time ágil em função da prática da interação com o time de operações. Nem todos esses detalhes são possíveis de serem documentados, ou muito menos serem tratados em contrato.

PROJETO DE SOFTWARE

Para complicar um pouco, a setLOG recebeu a encomenda de um doce da fruta copacaki (fictícia), que ainda não estava na programação. Como atender a esse pedido, requer embalagem específica e tempo de transporte/armazenamento diferenciado?

Você deve praticar um dos valores ágeis para atender à demanda e não deixar de atender esse caso, por meio do “Responder a mudanças”.

Muito bem, chegando ao fim dessa aula, estudante, você está preparado para encarar desafios complexos e aberto ao diálogo com os clientes e indivíduos envolvidos com o projeto. Agora continue estudando para se manter atualizado.

Saiba mais

Estudante, acesse outros materiais, nos quais fará leituras atraentes e terá maior conhecimento e independência em sua atuação profissional.

1. Processo ágil, princípios e política

Leia os itens 3.3 O que é processo ágil?, 3.3.1 Princípios da agilidade, 3.3.2 A política do desenvolvimento ágil, página 40 a 41. Você reconhecerá melhor os processos Ágil.

Fonte: PRESSMAN, R. S.; MAXIM, B. R. [Engenharia de software](#). Porto Alegre: AMGH, 2021. [Minha Biblioteca].

2. Estilo de liderança

Leia o item 21.2.1 Estilo de liderança, página 196. Ele vai relacionar as atitudes por tipo de liderança: Ordem em comparação ao Aprendizado.

Fonte: MAXIMIANO, A. C. A.; VERONEZE, F. [Gestão de Projetos: preditiva, ágil e estratégica](#). Barueri: Atlas, 2022. [Minha Biblioteca].

Desejamos bons estudos e agradecemos a sua participação!

Referências

CAMARGO, R.; RIBAS, T. **Gestão ágil de projetos**. São Paulo: Saraiva Educação, 2019.

CAVALCANTI, F. R. P. **Fundamentos de gestão de projetos**: gestão de riscos. São Paulo: Atlas, 2016.

MAXIMIANO, A. C. A.; VERONEZE, F. **Gestão de projetos:** preditiva, ágil e estratégica. Barueri: Atlas, 2022.

Aula 2

Scrum

Scrum

Este conteúdo é um vídeo!



Para assistir este conteúdo é necessário que você acesse o AVA pelo computador ou pelo aplicativo. Você pode baixar os vídeos direto no aplicativo para assistir mesmo sem conexão à internet.

Dica para você

Aproveite o acesso para baixar os slides do vídeo, isso pode deixar sua aprendizagem ainda mais completa.

Estudante, esta videoaula foi preparada especialmente para você. Nela, você irá aprender conteúdos importantes para a sua formação profissional. Vamos assisti-la?
Bons estudos!

Ponto de Partida

Se um restaurante especializado em comida japonesa está sendo “ameaçado” pela concorrência, como você pode contribuir com o seu conhecimento em TI para a situação ser equilibrada?

Bem, estudante de Projeto de Software, o desafio nessa aula é deixar o profissional bem próximo das estratégias e ações da Yagura Nipo Sabores para conseguir se manter no mercado da culinária nipônica em uma grande cidade.

Imagine que o seu time de desenvolvedores conheça na teoria os conceitos da metodologia ágil, mas que está prestes a praticar um framework puramente ágil no desenvolvimento de software desse restaurante que precisa se reinventar.

O desafio é implementar o Scrum em pouquíssimo tempo, obter o máximo de resultado no desenvolvimento de software e assessorar a Yagura Nipo Sabores nesse clima de concorrência.

Agora, você reconhecerá detalhadamente o Scrum, vamos lá!

Vamos Começar!

Estudante, agora manteremos o foco em Scrum!

O framework amplamente utilizado na gestão de projetos de desenvolvimento ágil de software, Scrum, é bem conhecido pelo diagrama claro e compacto que ilustra todos os passos do seu fluxo de trabalho, conforme a Figura 1.

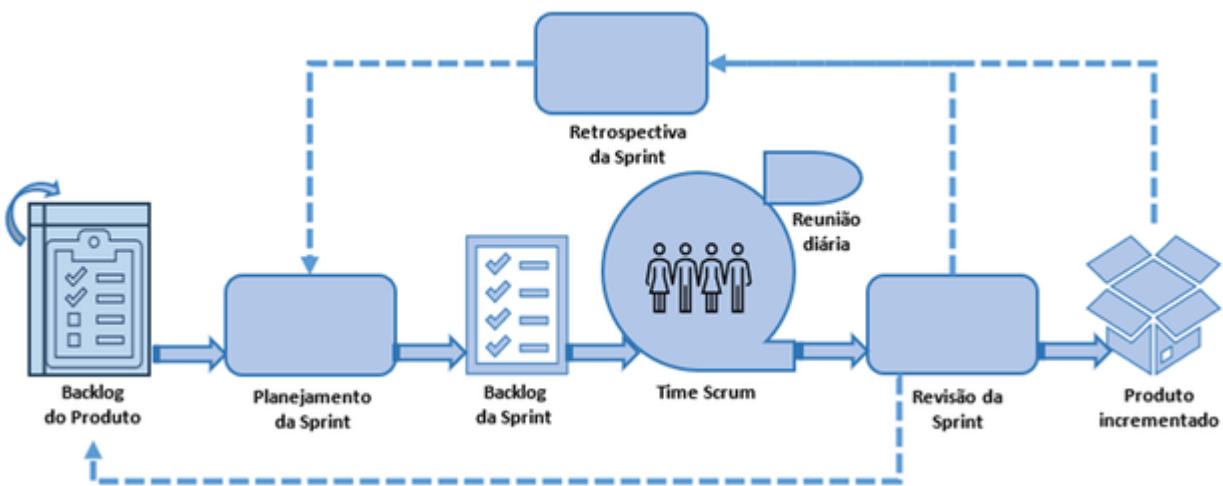


Figura 1 | Framework Scrum. Fonte: adaptado de Camargo (2019, p. 143).

Te convidamos a conhecer os principais recursos do Scrum, os quais garantem o bom funcionamento do modelo apropriado para projeto de software.

Artefatos do Scrum

O Product Backlog é uma lista ordenada de tudo o que é conhecido como necessário no produto. É a única fonte de requisitos para quaisquer alterações e implementações a serem feitas no produto.

O Product Backlog é dinâmico e passa por constantes mudanças para identificar no que o produto precisa para ser apropriado, competitivo e útil, conforme as necessidades do sistema. O Product Backlog ficará completo somente na última Sprint, ao concluir o projeto.

O Sprint Backlog é uma parte dos itens do Product Backlog selecionados para o Sprint, parte que será o incremento do produto, quando se dará o *status* de “Concluído” a ele. O Sprint Backlog é um plano com detalhes suficientes para que as mudanças em andamento possam ser compreendidas na Reunião Diária (Daily Scrum).

Sobre o software pronto a ser entregue e útil ao time de operações, ao final da Sprint, considera-se o incremento do Scrum. Esse artefato de software é a consolidação da implementação efetuada pelo Time Scrum, com todas as funcionalidades e modificações, desde a primeira Sprint.

Responsabilidades do Scrum

As responsabilidades no Scrum são bem definidas para garantir uma abordagem eficiente e colaborativa no desenvolvimento de produtos.

O Time Scrum que entrega, a cada Sprint, produtos de forma iterativa e incremental, maximizando oportunidades de *feedback*. Entregas incrementais de produtos “Prontos” garantem uma versão potencialmente útil do produto funcional e estão sempre disponíveis.

O Product Owner é o único responsável pelo gerenciamento do Product Backlog, incluindo, atualizando e validando cada item (história ou requisito). Faz parte desse gerenciamento:

- Expressar claramente os itens do Backlog do Produto;
- Ordenar os itens do Product Backlog para melhor atingir objetivos e missões;
- Otimizar o valor do trabalho realizado pelo Time de Desenvolvimento;
- Garantir que o Product Backlog seja visível, transparente e claro para todos e mostre os itens com os quais o Time Scrum trabalhará na próxima Sprint.

O Scrum Master é um líder servidor do Time Scrum, quem ajuda todos interessados a entender suas interações com o Time Scrum e auxilia todos a mudar essas interações para maximizar o valor criado pelo Time Scrum.

Siga em Frente...

Eventos do Scrum

Uma característica fundamental é cumprida pelos 5 eventos, a duração máxima de cada, timeboxing (ou timeboxed). Essa limitação é um dos fatores mais importante do framework Scrum.

Estudante, compare cada um dos eventos e saiba explicá-los, assim estará apto a fazer parte do time Scrum e contribuir com o sucesso do Projeto de Software.

1. Sprint

Fatores que levam o Scrum a determinar uma Sprint como evento principal do seu framework, ciclos de duração fixa (2 a 4 semanas) com outros eventos obrigatórios:

- Se você reservar um período para se concentrar para executar uma sequência de tarefas juntamente com várias pessoas, com resultado esperado definido e esclarecido, a probabilidade de sucesso será maior;
- Todas as Sprints de um projeto têm um período fixo (timeboxing) para o time conseguir medir a sua velocidade de desenvolvimento, ou seja, descobrir e aperfeiçoar a quantidade de entrega numa iteração do projeto.
- As informações claras, objetivas e sucintas sobre a entrega do que havia sido planejado facilitarão a comunicação.
- Para aumentar as chances de sucesso no projeto, faz-se avaliação do produto e do processo a cada Sprint, além do planejamento no seu início, assim, a cada iteração o produto será incrementado.
- O tempo de uma Sprint tem uma duração na qual a visibilidade do resultado é alcançável.

Ainda é possível ocorrer impedimentos ou problemas que impactarão o sucesso da Sprint, podendo ser resolvido, resultado redimensionado ou cancelamento da Sprint.

2. Planejamento da Sprint

Este evento é realizado para que todos os envolvidos (Time, Product Owner e Scrum Master) estejam cientes de todos os detalhes do resultado esperado na

Sprint, numa duração máxima de 4 a 8 horas.

O planejamento deve prever disponibilidade das pessoas envolvidas e o detalhamento de cada item deve estar claro para todos.

3. Reunião diária

Evento, de alguns minutos, sempre no mesmo horário, no qual todos do Scrum Team falem sobre sua participação no projeto, principalmente do impedimento/gargalo, além de verificar o progresso do início da Sprint para reconhecer possível atraso.

As resoluções dos impedimentos devem ser tratadas logo após a reunião diária.

4. Revisão da Sprint

Reunião própria, de 2 a 4 horas, no qual o Product Owner avaliará o resultado, em relação ao software que será entregue ao cliente/usuário, composto por todos os itens planejados na Sprint.

O preparo para a reunião será extremamente produtivo se esta for bem objetiva e eficiente, caso contrário, o resultado poderá ser caótico.

5. Retrospectiva da Sprint

Reunião específica, na qual o time e Scrum Master fará uma avaliação de atividades, ferramentas, recursos tecnológicos, técnicas, entre outros, servindo de aprendizado para as próximas Sprints ou projetos. Nesse evento, o foco é a Inspeção da Sprint, ou seja, o time deve se inspecionar; em seguida, deve criar um plano de melhoria para as próximas.

Todos os itens verificados para melhoria devem ser discutidos e divulgados a todos os impactados; idealmente, todos devem expor a sua experiência.

Estudante, agora você está preparado para trabalhar com o Scrum, continue estudando para ir se aprofundando e aumentando habilidades em metodologias ágeis em Projeto de Software.

Vamos Exercitar?

Compreenda como a Yagura Nipo Sabores será competitiva com a estratégia de negócio aliada à tecnologia da informação, por meio da sua participação com os conhecimentos em Scrum.

A partir dos itens do Sprint Backlog devidamente elicitados pelo Product Owner e a possibilidade de o Time Scrum esclarecer dúvidas na definição dos itens a serem desenvolvidos na reunião de Planejamento da Scrum, há maior segurança para a execução da Sprint.

Para arquitetar adequadamente o software, a Revisão da Sprint permite que o time acompanhe todos os procedimentos que estão sendo aplicados no Programa de Fidelização no qual o restaurante melhorou, oferecendo recompensas mais atraentes, descontos exclusivos e eventos especiais para clientes fiéis. O Scrum Master acessou o time de operações sempre que o time teve dúvidas nas complexas regras de negócios na classificação de clientes fiéis.

Estudante, a partir desse cenário, você deve ter percebido o poder do Scrum no desenvolvimento de software, no qual o time poderá se adaptar às novas realidades enfrentadas por seus clientes.

Continue estudando e obtendo maior segurança, na prática ágil exigida em times de Projeto de Software.

Saiba mais

Estudante, convidamos você para elevar o seu conhecimento e habilidade para ser referência em desenvolvimento ágil. Saiba mais acessando os conteúdos sobre comunicação e inspeção. Assim, você estará mais bem preparado para exercer suas atividades profissionais!

1. Comunicação em projeto de software

PROJETO DE SOFTWARE

Leia o item 8.8.3 Reunião Diária (Daily Scrum), página 160, que trata sobre o progresso diário da Sprint.

Fonte: CAMARGO, R.; RIBAS, T. [Gestão ágil de projetos](#). São Paulo: Saraiva Educação, 2019. [Minha Biblioteca]

2. Aprendizado pela transparência – inspeção e adaptação

Leia o item 8.8.5 Retrospectiva da Sprint, páginas 163 a 166, e extraia uma síntese sobre a reunião mais importante do Scrum para o time que busca melhoria contínua.

Fonte: CAMARGO, R.; RIBAS, T. [Gestão ágil de projetos](#). São Paulo: Saraiva Educação, 2019. [Minha Biblioteca].

c. Vantagens usando Scrum – *feedback*

Leia o item 3.1 Scrum, páginas 146 a 150, e entenda como o Scrum possibilita mais entregas em menor tempo.

Fonte: SANTOS, H. D. V.; CARVALHO, F. A. S.; CAPUCHO, I. M. Relação entre a utilização do framework Scrum e a obtenção de vantagem competitiva. [Ágora: revista de divulgação científica](#), [S. I.], v. 28, p. 140-160, 2023.

Desejamos bons estudos e agradecemos a sua participação!

Referências

CAMARGO, R.; RIBAS, T. **Gestão ágil de projetos**. São Paulo: Saraiva Educação, 2019.

MAXIMIANO, A. C. A.; VERONEZE, F. **Gestão de projetos**: preditiva, ágil e estratégica. Barueri: Atlas, 2022.

PRESSMAN, R. S.; MAXIM, B. R. **Engenharia de software**. Porto Alegre: AMGH, 2021.

SANTOS, H. D. V.; CARVALHO, F. A. S.; CAPUCHO, I. M. Relação entre a utilização do framework Scrum e a obtenção de vantagem competitiva. [Ágora: revista de divulgação científica](#), [S. I.], v. 28, p. 140-160, 2023. Disponível em:

<https://www.periodicos.unc.br/index.php/agora/article/view/4696> . Acesso em: 24 out. 2023.

Aula 3

Abordagens inovadoras em Projeto de software

Abordagens inovadoras em Projeto de software

Este conteúdo é um vídeo!



Para assistir este conteúdo é necessário que você acesse o AVA pelo computador ou pelo aplicativo. Você pode baixar os vídeos direto no aplicativo para assistir mesmo sem conexão à internet.

Dica para você

Aproveite o acesso para baixar os slides do vídeo, isso pode deixar sua aprendizagem ainda mais completa.

Estudante, esta videoaula foi preparada especialmente para você. Nela, você irá aprender conteúdos importantes para a sua formação profissional. Vamos assisti-la? Bons estudos!

Ponto de Partida

Pensemos na seguinte situação: após conhecer a estratégia de um empreendimento, para então iniciar o desenvolver ou ampliar as funcionalidades do sistema, se torna bem menos complexo.

Agora, estudante, pensemos num passo antes! Caso você esteja diante de uma situação que precisa descobrir quais lacunas de produto ou serviço (inovação) que o empreendimento gostaria de atender, você poderá recorrer às técnicas para abstrair além das regras existentes. Em seguida, você deve montar a estratégia de mudanças em software.

Bem, então, proponho compreendermos como a Shitake Food aplicará técnicas como Design Thinking, MVP (*Minimum Viable Product*) e Lean e Six Sigma para alavancar seus negócios na produção e distribuição de cogumelos da culinária nipônica. Dentre muitas mudanças que a iniciativa pretende, uma delas é diversificar seus negócios em cogumelos desidratados, incrementar com receitas exclusivas convidando chefs criativos, kits de cultivo doméstico, entre outros.

Vamos Começar!

Seja bem-vindo ao mundo das inovações em projeto de software. Uma vez que você está locado em projetos de mudanças radicais ou novidades em soluções de TI, te convidamos a conhecer técnicas largamente úteis para o desenvolvimento de software.

Design Thinking em Projetos de TI

O segredo para entender e ter sucesso nessa prática é perder o medo de errar! Parece estranho, mas o Design Thinking é ser disruptivo, por isso quem tem essa proposta jamais desistirá após sucessivos erros. Perceba: isso é diferente de querer errar! Aqui, você precisa querer diferente, ser irreverente, ousado ou atrevido!

Portanto, é melhor “falhar” em laboratório experimentando do que no mercado, o que se chamam de “*fail fast*”; quanto mais cedo errar, maior será a experiência para encontrar uma alternativa certa.

A construção de alguma alternativa deve passar pelo processo Design Thinking, o qual possui três princípios:

- **Empatia:** pensar primeiro nas pessoas, antes mesmo do papel de empresário, consumidor, usuário de sistema, entre outros.
- **Colaboração:** criar algo em pares (apesar da criação ser algo individual), aqui o princípio é ser cocriador, inclusive envolvendo profissionais de outros setores, diferentes perfis e pontos de vista.
- **Experimentação:** buscar uma opção, que seja aceita, ou que funciona, pode passar por inúmeras tentativas, por isso quanto antes percorrer pela experiência, mais acertada será opção.

Então, esteja preparado para acertar, errando antes!

MVP – *Minimum Viable Product*

Primeiramente, você não deve confundir esse conceito com produto mais simples possível a ser entregue. Entregar algo (software) somente para cumprir o prazo previsto, por exemplo, não responde à necessidade de um MVP.

Portanto, estudante, o MVP deve sempre atingir o resultado esperado. Isso significa que o software entregue atenderá às necessidades, com certeza! Por isso, esse é o detalhe do MVP, saber se o software teve sucesso, ou seja, alcançou os resultados esperados.

O MVP pretende validar um certo conjunto de possibilidades de negócio, no menor tempo e esforço possível. Num exemplo, conforme a Figura 1, a entrega do primeiro produto é funcional para a pessoa poder se locomover de um local para outro; até que o quarto produto tenha mais conforto e segurança.



Figura 1 | Produto entregue usando MVP. Fonte: adaptado de Camargo (2019, p. 131).

Veja que o projeto está sendo entregue conforme a necessidade da pessoa, sendo considerados os custos e prazos compatíveis. Agora, reflita sobre as possibilidades em projeto de software.

Siga em Frente...

Lean e Six Sigma

Estudante, você sabe que valores em software se trata de um fator de utilidade e benefícios ao usuário/cliente. Sabe também que o desenvolvimento de software é dependente de pessoas projetando e codificando, ou seja, de horas debruçadas em construções e testes. Pois bem, fica simples compreender que perseguir a qualidade no processo de um projeto de software contempla também reduzir o esforço do desenvolvimento e maximizar valor entregue. Esse é o conceito Lean!

O Lean é um processo gradativo que se aplica em qualquer processo produtivo para reduzir custos (principalmente o desperdício), melhorar a qualidade do software e ampliar o valor para o cliente/usuário. Lembrando que o retrabalho em desenvolvimento de software é um grande vilão, quando se trata de desperdício de horas corrigindo erros!

Da mesma forma que o Lean, o Six Sigma também é um conceito de melhoria da qualidade no processo. Ambos requerem ferramentas da administração e da gestão da qualidade para serem implementados. Apesar de serem iniciativas indiretas do processo de desenvolvimento de software, exercem importante influência nas organizações ou time de desenvolvimento que buscam constante eficiência, respeito às pessoas e foco na entrega de valor ao cliente.

Inicialmente se estabelecem estrategicamente os processos que planeja monitorar e melhorar, criam-se indicadores de medição relevantes às características que deseja aumentar a qualidade. Nessa primeira etapa é fundamental ter foco em conhecer e coletar a quantidade de defeitos. Em seguida, analisam-se os índices com o intuito de melhorar os processos, portanto se aplica a melhoria de maneira controlada, aferindo o ganho conquistado.

O Six Sigma tem como principais benefícios:

- O engajamento do time que executa as atividades produtivas, principalmente o Time Scrum ou equipe de desenvolvimento.
- O combate ao desperdício, tanto em tempo quanto de recursos, em projeto de software está relacionado diretamente ao tempo de construção e teste do código-fonte e aos recursos computacionais (hardware e software).
- Um fator relevante em software é quanto às falhas detectadas, em ambiente operacional, que deveriam ter sido verificadas durante a construção, chamados de bugs que roubam o tempo do desenvolvedor com atividades de correção.
- A padronização de processos proporcionará melhoria contínua, tanto na execução da atividade quanto no uso de ferramentas.

O Six Sigma tem como conceito o cálculo do DPMO (*Defects Per Million Opportunities*). Antes, entendamos o que é DPO (Defeitos por oportunidade);

Se um item de software tem possibilidade de 1500 defeitos, então para um cenário de 800 itens de software no projeto, foram identificados 1100 defeitos, então temos o DPO igual a $1100 / 800 * 1500$, ou seja 0,000917 que por sua vez fica em 9166,666667 defeitos por milhão, ou seja, 0,916666667% de defeitos, ou seja, 99,08333333% de conformidade.

Este é um cálculo básico para medição da qualidade pelo Six Sigma (6σ) e poderá ser empregado para medir o nível de conformidade do produto do Projeto de Software.

Vamos Exercitar?

Voltando ao cenário desafiador Shitake Food, no qual você é o protagonista do projeto de software, você deve idealizar a diversificação do negócio em cogumelos desidratados, incremento de produtos com receitas exclusivas convidando chefs criativos, inovação com kits de cultivo doméstico e com certeza precisará contar com muitas pessoas, especialistas, talentos, além do seu conhecimento.

No primeiro momento, acreditamos bem produtiva e válida a sua participação no processo de Design Thinking, conhecendo as pessoas e como elas trabalham (empatia), suas realidades e necessidades no processo de criação de novos produtos e diversificação dos produtos atuais.

Que tal dar uma solução de fluxo contínuo, segundo Lean, para a concepção do kit de cultivo doméstico? O kit poderá ter referências em quantidade, tipo de cogumelo, local de entrega e prazo para retirada. Uma funcionalidade no software que faria todo fluxo de montagem à logística de entrega, a partir do pedido de um cliente, passando por testes, seria o ideal.

Estudante, agora que você tem as competências em abordagens inovadoras em projeto de software, continue exercitando e se aprofunde no esse assunto, o qual é crescente em desenvolvimento de software.

PROJETO DE SOFTWARE

Saiba mais

Convidamos você a estudar alguns itens importantes que agregarão valor ao seu conhecimento, possibilitando a você integrar o que aprendeu com a sua prática profissional.

1. Exemplo de MVP

Leia o item 7.5 Definindo o MVP com story mapping, páginas 132 a 133, que exemplifica o uso do MVP.

Fonte: CAMARGO, R.; RIBAS, T. [Gestão ágil de projetos](#). São Paulo: Saraiva Educação, 2019. [Minha Biblioteca]

2. Lean

Leia o item 6.1.1.3 Filosofia Lean de Gestão, páginas 104 a 107, e compreenda o Lean em gestão.

Fonte: CAMARGO, R.; RIBAS, T. [Gestão ágil de projetos](#). São Paulo: Saraiva Educação, 2019. [Minha Biblioteca]

Desejamos bons estudos e agradecemos a sua participação!

Referências

CAMARGO, R.; RIBAS, T. **Gestão ágil de projetos**. São Paulo: Saraiva Educação, 2019. [Minha Biblioteca]

FILHO, Wilson de Pádua P. **Engenharia de software - Projetos e processos**, Vol. 2, 4 ed. Rio de Janeiro: LTC, 2019. [Minha Biblioteca]

MAXIMIANO, A. C. A.; VERONEZE, F. **gestão de projetos**: preditiva, ágil e estratégica. Barueri: Atlas, 2022. [Minha Biblioteca]

PRESSMAN, R. S.; MAXIM, B. R. **Engenharia de software**. Porto Alegre: AMGH, 2021. [Minha Biblioteca]

Aula 4

Diferenças e semelhanças dos Envolvidos no Projeto

PROJETO DE SOFTWARE

Diferenças e semelhanças dos Envolvidos no Projeto



Este conteúdo é um vídeo!

Para assistir este conteúdo é necessário que você acesse o AVA pelo computador ou pelo aplicativo. Você pode baixar os vídeos direto no aplicativo para assistir mesmo sem conexão à internet.

Dica para você

Aproveite o acesso para baixar os slides do vídeo, isso pode deixar sua aprendizagem ainda mais completa.

Estudante, esta videoaula foi preparada especialmente para você. Nela, você irá aprender conteúdos importantes para a sua formação profissional. Vamos assisti-la? Bons estudos!

Ponto de Partida

Aqui, estudante, o seu desafio é desenvolver funcionalidades acerca de um sistema que irá revolucionar grandes centros urbanos, usando o gatilho da sustentabilidade e da falta de espaço para mobilidade.

Há muito tempo cresce a demanda de locação de patinete elétrica, porém os sistemas ainda necessitam de adaptações às mudanças/acomodações em relação ao sistema de agendamento, escolha do tipo do patinete e autonomia, local de devolução, abandono e recuperação e a manutenção do veículo.

Sem contar que a demanda por novas funcionalidades ou variações das existentes se tornou constante devido ao aumento de pessoas que aderiram a essa modalidade no trânsito de uma cidade movimentada.

Você vai se envolver com a aplicação das habilidades de um gerente de projeto em conjunto com iniciativas ágeis do Product Owner para aumentar as chances de sucesso em Projeto de software ScooterEasy.

Compreendamos também quais são as responsabilidades dos demais *stakeholders* do projeto em desenvolvimento.

Vamos Começar!

Papel do gerente de projetos

Vamos falar de causas em projetos, pelas quais o gerente exerce grande importância! São características diversas em função do alto grau de incerteza e escala, impostas pela competitividade nas organizações. Ainda, a interoperabilidade entre o legado e aos integrantes de terceiros, se tornam num amaranhado complexo.

Após vários anos de experiências e de compartilhamento com outros colegas, Kouzes (2014 *apud* PRESSMAN, 2021 p. 493) elencou práticas de líderes de projeto na área da tecnologia:

- Seja o modelo: Os líderes devem praticar o seu discurso. Eles demonstram o comprometimento com a equipe e com o projeto por meio do sacrifício compartilhado;
- Inspire e crie uma visão compartilhada: Os líderes reconhecem que não é possível liderar sem seguidores. É importante motivar os membros de equipe a ligarem os seus sonhos pessoais aos objetivos da equipe;
- Questione o processo: Os líderes devem tomar a iniciativa de procurar maneiras inovadoras de melhorar o seu próprio trabalho e o das suas equipes;
- Capacite os outros a agir: Crie confiança e facilite relacionamentos para promover as habilidades de colaboração da equipe.
- Incentive o espírito: Comemore as conquistas dos indivíduos.

Estudante, você deve perceber que não existe uma fórmula simples e mágica para todas as situações e projetos na sua atuação profissional. Saiba que ambos, gerente de projeto e líder ágil, devem ter habilidade em ser colaborativo, mantendo a equipe coesa e alinhada a cada tipo de projeto que venham desenvolver.

Se na estrutura tradicional o gerente determina o que fazer e “empurra” atividades ao time, determinando “o que” e “como” fazer. Ao contrário é a metodologia ágil, na qual o time tem autonomia para puxar atividades, comprehende “o que” fazer em parceria com o líder de projeto e o time que decide “como fazer”.

Papel do Product Owner

O principal papel do PO é o gerenciamento de recursos financeiros do projeto de software. Consequentemente, ele estabelece a prioridade do Time Scrum na Sprint, dos itens a serem desenvolvidos, inclusive o gerenciamento dos itens que converterão em valores à organização e, consequentemente, o retorno sobre o investimento.

Se o PO tem o domínio do Product Backlog e do Sprint Backlog, é ele também o responsável pela avaliação ao final da Sprint, momento em que reorganiza as prioridades para as próximas Sprints, com base no *feedback* recebido. Caso necessite, ele tem o poder de cancelar a Sprint ou todo projeto, principalmente quando verificar que não terá o retorno esperado. Por meio do Quadro 1, é possível visualizar as responsabilidades do Product Owner, comparado ao do Scrum Master e do Time Scrum.

PROJETO DE SOFTWARE

Atividades	Scrum Master	Product Owner	Time Scrum
Escopo		Primário	Secundário
Custos		Primário	
Qualidade			Primário
Aquisições		Primário	
Comunicação com o cliente		Primário	Secundário
Engajar e inspirar o time	Primário		
Remover impedimentos	Primário		Primário

Quadro 1 | Atividades em um Projeto de Software. Fonte: adaptado de Camargo (2019, p. 154).

A comunicação entre o cliente e Time Scrum se tornou imprescindível em épocas de mudanças tão rápidas e amplas, em função da competitividade nos negócios. Quando a organização se reinventa, os sistemas sofrem modificações que demandam manutenções nos softwares. Para garantir que esse complexo ambiente colabore com o desenvolvimento do sistema, o PO garante a comunicação, recurso básico para a continuidade das evoluções.

Siga em Frente...

Responsabilidade do *stakeholders*

Uma iniciativa providente em todo projeto de software é a procura incessante das pessoas certas com os dados necessários para conceber as funcionalidades do software. Sem essa definição, faltará um norte adequado para o projeto. Com os questionamentos adequados, encontrará os *stakeholders* que vão nortear o produto necessário:

- Quem está por trás das solicitações desse software são as pessoas oficiais?
- Quem usará e se beneficiará da solução/software pronto?
- Todas as funcionalidades elencadas fazem parte do mesmo software?
- Essas são as pessoas que sabem quando cada funcionalidade será útil?
- Qual será o resultado econômico da solução?

Na tentativa de obter o máximo de precisão das funcionalidades do software, com as pessoas oficiais, na época adequada, você entrará os verdadeiros *stakeholders*. Estes serão os responsáveis pela condução e definição do Backlog do Produto.

Estudante, acreditamos que você tenha percebido que a responsabilidade dos *stakeholders* é informar e direcionar as necessidades ao Product Owner no gerenciamento dos itens do produto.

Vamos Exercitar?

Os papéis do gerente e do Product Owner em Projeto de Software são profundamente testados diariamente. Eles são responsáveis pela motivação do time e na resolução de conflitos e desenvolvimento eficiente das atividades-fim. Porém, sem a participação dos *stakeholders*, o projeto não teria material/informação necessário para construir e evoluir os softwares, base para o trabalho do Product Owner.

Podemos elencar alguns dos *stakeholders* para o projeto ScooterEasy, porém saibamos que não são os únicos:

- Para ter a visão e estratégia: os fundadores são responsáveis pela definição dos objetivos de negócios e garantem que o desenvolvimento do software esteja alinhado a longo prazo.
- Para o gerenciamento do projeto: o gerente de projeto é responsável por planejar, coordenar e supervisionar todo o processo de desenvolvimento de software.
- Desenvolvedores de software: são eles que escrevem o código do aplicativo, garantindo que ele funcione corretamente e atenda aos requisitos técnicos.
- Designer de Interface de Usuário (UI) / Experiência do Usuário (UX): são responsáveis por criar uma interface de usuário atraente e intuitiva, garantindo uma experiência positiva para os usuários do aplicativo.
- Engenheiro de hardware: responsáveis pela integração do software com os patinetes.
- Equipe de qualidade e testes: responsáveis por testar o aplicativo para identificar bugs, garantir que ele funcione corretamente e relatar quaisquer problemas aos desenvolvedores para correção.
- Investidores: fornecem financiamento para o desenvolvimento do software e esperam um retorno sobre o investimento.
- Usuários finais: são essenciais para dar *feedback* sobre o aplicativo, relatar problemas e usar o software como pretendido, têm participação valiosa para aprimorar o aplicativo continuamente.

Saiba mais

1. Comunicação em projeto de software

Leia o item 2.2.1 Scrum que trata sobre a comunicação.

Fonte: DIAS, P. N. ; CUNHA, P. Relação entre habilidades interpessoais e de equipe e o gerenciamento das comunicações em projetos ágeis. [Boletim do Gerenciamento](#), [S.I.], v. 33, n. 33, p. 1-10, nov. 2022. ISSN 2595-6531.

PROJETO DE SOFTWARE

2. Os pilares e papéis do Scrum

Leia o item 2.3 Framework Scrum do artigo e extraia uma síntese sobre os papéis do Scrum.

Fonte: CARNEIRO, R. A. O valor percebido pelo cliente na gestão de projetos de software, utilizando o framework scrum. [Revista de Gestão de Projetos: Dialnet](#), 2022.

3. Envolvidos e requisitos do sistema

Leia o item 7.2 Estabelecimento da base de trabalho e entenda como encontrar os *stakeholders* certeiros e os requisitos funcionais alinhados ao produto do projeto de software.

Fonte: PRESSMAN, R. S.; MAXIM, B. R. [Engenharia de software](#). Porto Alegre: AMGH, 2021. [Minha Biblioteca]

Referências

CAMARGO, R.; RIBAS, T. **Gestão ágil de projetos**. São Paulo: Saraiva Educação, 2019.

FILHO, R. I.; CIERCO, A. A. **Governança, ESG e Estrutura Organizacional**. São Paulo: Actual, 2022.

PRESSMAN, R. S.; MAXIM, B. R. [Engenharia de software](#). Porto Alegre: AMGH, 2021.

Aula 5

Encerramento da Unidade

Videoaula de Encerramento



Este conteúdo é um vídeo!

Para assistir este conteúdo é necessário que você acesse o AVA pelo computador ou pelo aplicativo. Você pode baixar os vídeos direto no aplicativo para assistir mesmo sem conexão à internet.

Dica para você

Aproveite o acesso para baixar os slides do vídeo, isso pode deixar sua aprendizagem ainda mais completa.

PROJETO DE SOFTWARE

Estudante, esta videoaula foi preparada especialmente para você. Nela, você irá aprender conteúdos importantes para a sua formação profissional. Vamos assisti-la?
Bons estudos!

Ponto de Chegada

Olá, estudante! Para desenvolver a competência desta Unidade, a qual consiste em entender como as abordagens e metodologias ágeis e inovadoras contribuem no desenvolvimento de software e são contempladas no projeto de software, você deverá conhecer primeiramente os conceitos fundamentais do Manifesto Ágil, Scrum, Design Thinking, Lean e Six Sigma e responsabilidades do *stakeholders*.

Bem, você tem essa oportunidade de se conectar com a sua realidade por conceitos que trabalhamos aqui para compreender o quanto cada um deles vai te auxiliar no projeto de software.

Primeiramente, acreditamos que o uso de valores e princípios da metodologia ágil nos conecta a essa situação de mudanças e de urgência que vivenciamos diariamente. Em vários momentos do dia é possível perceber que existe alguma informação nova e que nos ajuda, ou nos obriga a encontrar soluções diferentes a problemas atuais ou antigos.

Portanto, a partir desse paradigma você já começa bem-posicionado para ter a empatia com as pessoas que serão beneficiadas com o software do seu projeto. Ou seja, deve ter despertado em você o desejo de construir produtos que funcionem já na primeira versão disponibilizada ao time de operações; da mesma forma, estar muito atento às mudanças ao redor do cenário que pode influenciar o sistema, para poder readaptar o software ao novo e ao diferente.

Estudante, compreender o framework Scrum é fundamental na prática da disciplina quanto aos eventos, todos eles timeboxing, que estabelecem claramente momentos de planejamento, controle, avaliação e melhoria do produto e do projeto de software. Devemos também entender os papéis de definição de prioridades, resolução de impedimentos e de colaboração às necessidades de requisitos funcionais e não funcionais do sistema.

Avançando mais, você consegue perceber que se deparar com problemas complexos, ou na construção de soluções inovadores, poderá relacionar os benefícios do Design Thinking, do Lean e Six Sigma, como também no MVP. Essas metodologias associam objetivos de desenvolvimento de produtos e de otimização de processos; consequentemente geram melhorias da qualidade no projeto de software.

É Hora de Praticar!

PROJETO DE SOFTWARE



Este conteúdo é um vídeo!

Para assistir este conteúdo é necessário que você acesse o AVA pelo computador ou pelo aplicativo. Você pode baixar os vídeos direto no aplicativo para assistir mesmo sem conexão à internet.

O seu desafio está no projeto SobMedida, no qual embrenharemos em situações diversas, pois o time de desenvolvimento deve construir soluções em software para atender às demandas complexas e constantes, vejamos!

Os primeiros sistemas foram concebidos por métodos tradicionais, nos quais o gerente do projeto fez um excelente trabalho de análise, construção e entrega de um software que atendeu a SobMedida por muitos anos. Porém, com as frequentes demandas da competitividade, o software se tornou muito complexo para sofrer manutenções e evoluções.

Além do software, a SobMedida estava sem alternativas para identificar e remover obstáculos no setor de suprimentos, que impediam o setor de produção atingir suas metas, por problemas de fornecimento de materiais, em função de problemas técnicos ou atrasos.

Outro desafio estava na recorrência de falhas de software, ao longo do ciclo de vida. As modificações implementadas nas aplicações foram deteriorando o código-fonte; a ponto de impedir que novas implementações fossem feitas.

Estudante, perante esse cenário, você já consegue perceber que existe a necessidade de experimentar um novo método de desenvolvimento, certo?

Portanto, resgate suas competências em métodos ágeis, papéis de Scrum Master, Design Thinking e Product Backlog para colaborar na recolocação da SobMedida em seu mercado de atuação.

- Como é possível obter benefícios do Scrum num ambiente em que todos só conhecem a metodologia tradicional?
- Seria possível utilizar a metodologia Ágil juntamente com a metodologia Design Thinking?
- Ao gerenciar os requisitos do sistema, por meio do Product Backlog, como serão inseridos itens de correção do software que foram identificados no ambiente de operações?

Desvendaremos juntos soluções para que a SobMedida se reposicione no mercado, com sistema versátil e dinâmico no tratamento das informações.

Lembremos que é preciso adaptar os princípios e as práticas ágeis às necessidades específicas de uma fábrica de móveis SobMedida. O Scrum Master desempenha um papel fundamental na criação de um ambiente ágil que promove eficiência, qualidade e satisfação do cliente. Vale lembrar que nem sempre é possível solucionar todas as demandas com recursos computacionais e com a equipe de TI.

Inicialmente, a situação caótica no setor de suprimentos deve ser analisada por pessoas especializadas dessa área. Portanto, gestores da SobMedida e Scrum Master propiciam a participação de profissionais, outros *stakeholders*, experientes em cadeia de suprimentos. Neste caso, a decisão é a utilização do método Design Thinking, pelo qual obteremos uma solução em processos que será subsídio para novas implementações em software. Parte do resultado do Design Thinking será uma relação de requisitos funcionais, itens que farão parte do Product

PROJETO DE SOFTWARE

Backlog; outra parte serão modificações nos processos operacionais e estruturais da SobMedida.

O Scrum Master garantirá que os conceitos ágeis serão benéficos ao time de operações e aos dirigentes da SobMedida, pois um dos valores ágeis é a prática da colaboração com o cliente, juntos, construindo as soluções em sistemas. Na formação da equipe, o Scrum Master deve criar um ambiente colaborativo e auto-organizado, definindo os papéis adequados às necessidades da SobMedida, principalmente valorizar as interações contínuas para compreender e atender as necessidades dos indivíduos.

O último ponto, gestão de mudanças e de erros, fica a cargo do Product Owner na priorização dos itens do Product Backlog. E, juntamente com o Time Scrum, compor a Sprint Backlog durante o evento do planejamento da sprint, escolhendo as melhorias relevantes do sistema e, concomitantemente, as correções obrigatórias a serem implementadas na sprint.

Por meio das práticas ágeis, por algumas iterações, Scrum Master, Product Owner e demais *stakeholders* estimam a estabilização das aplicações, aumentando a confiança na operação do sistema e alcançando as metas em novas funcionalidades aderentes à estratégia da SobMedida. Bem, agora você tem o poder de aplicar em sua vida profissional a metodologia ágil e as abordagens inovadoras em projeto de software. Continue estudando para aumentar a sua competência e as habilidades em desenvolvimento de software.

Estudante, aposte em si, aumentando o seu conhecimento, buscando novas formas conceituadas e experimentadas por profissionais da gestão e de TI. O objeto foi elaborado para você se lembrar de alguns dos fundamentos das metodologias em gestão de projeto de software.

PROJETO DE SOFTWARE

PROJETO DE SOFTWARE

2
METODOLOGIAS EM GESTÃO DE PROJETO DE SOFTWARE

VALORES ÁGIL



Indivíduos e Interações: as pessoas são fundamentais para alcançar os resultados esperados pelas organizações, através da contínua interação

Produto em funcionamento: software com funcionalidade implementada a cada iteração, garantirá a estabilidade no ambiente de operações

Colaboração com o cliente: time ágil vivencia os desejos e necessidades do cliente.

Responder às mudanças: manter a arquitetura do software preparada para receber modificações constantes.

Eventos do Scrum (Timeboxing)

Sprint: evento principal que resultará um software em funcionamento sob condições planejadas e controladas pelos times de TI e de operações.

Planejamento da Sprint: estudo dos itens a serem implementados na Sprint, visando atender às prioridades dos negócios do cliente.

Reunião diária: identificação de impedimentos e consolidação do progresso da Sprint.

Revisão da Sprint: validação do produto de software.

Retrospectiva da Sprint: verificação dos processos do desenvolvimento de software.

SCRUM



DESIGN THINKING



Empatia: pensar primeiro nas pessoas, antes mesmo do papel de empresário, consumidor, usuário de sistema

Colaboração: criar algo em pares, ser cocriador, envolvendo outros setores, diferentes perfis e de ponto de vista

Experimentação: passar por inúmeras tentativas, através da experiência, chegar na opção certa

ENVOLVIDOS NO PROJETO

Criar oportunidades de comemoração e confiança é papel do **gerente de projeto**, além de incentivar a romper obstáculos para desenvolver soluções inovadoras.

Equipes de TI são formadas por líderes que conduzem profissionais com autonomia para “puxar” atividades e decidem “como” desenvolver soluções adequadas no software.

Por um lado, o Time de TI se responsabiliza pela qualidade do software via código-fonte estáveis, por outro o **Product Owner** define prioridades em escopo e orçamento do projeto, ficando ao **Scrum Master** o papel para engajar os stakeholders e preparar o ambiente criativo e inovador.



Figura 1 | Metodologias em Gestão de projeto de software. Fonte: elaborada pelo autor.

 Anhanguera

PROJETO DE SOFTWARE

CAMARGO, R.; RIBAS, T. **Gestão ágil de projetos**. São Paulo: Saraiva Educação, 2019.
CAVALCANTI, F. R. P. **Fundamentos de gestão de projetos**: gestão de riscos. São Paulo: Atlas, 2016.
MAXIMIANO, A. C. A.; VERONEZE, F. **Gestão de projetos**: preditiva, ágil e estratégica. Barueri: Atlas, 2022.
PRESSMAN, R. S.; MAXIM, B. R. **Engenharia de software**. Porto Alegre: AMGH, 2021.

Unidade 3

Ferramentas e Qualidade em Projeto de Software

Aula 1

Gestão de riscos

Gestão de riscos

Este conteúdo é um vídeo!



Para assistir este conteúdo é necessário que você acesse o AVA pelo computador ou pelo aplicativo. Você pode baixar os vídeos direto no aplicativo para assistir mesmo sem conexão à internet.

Dica para você

Aproveite o acesso para baixar os slides do vídeo, isso pode deixar sua aprendizagem ainda mais completa.

Estudante, esta videoaula foi preparada especialmente para você. Nela, você irá aprender conteúdos importantes para a sua formação profissional. Vamos assisti-la?
Bons estudos!

Ponto de Partida

Ao pensar em riscos, estamos pensando diretamente em prejuízo. Se você está pensando dessa forma, então é claro que vai tomar as devidas providências no seu planejamento para que durante a execução das atividades todo comportamento/resultado que indique um prejuízo seja tratado adequadamente, certo?

PROJETO DE SOFTWARE

Assim, seja bem-vindo ao estudo do tema que coloca em desconforto qualquer projeto de software, principalmente porque estamos falando de projetos e nesse campo sempre existe um grau de inovação, justamente porque “pisamos nesse terreno” mais propício a resultados indesejados.

Vejamos: se você é o gestor do projeto de controle da qualidade na produção de azeite de oliva, AZQuali, certamente irá se embrenhar nas possíveis oscilações que aumentam as chances de diminuir a qualidade do azeite, correto?

Pense se você, apenas você, dominaria todas as fontes de oscilações que impactam na qualidade do azeite. Ter a participação de especialistas em determinadas áreas do conhecimento facilitaria muito no controle de riscos da produção, processamento, armazenamento, ou na entrega do azeite.

Você está pronto para conhecer quais recursos utilizaremos para o bom gerenciamento de riscos? Tenha em mente, estudante, que compreender sobre o mapeamento, a natureza e a matriz de risco serão uma boa base para iniciar essa área complexa do projeto de software.

Vamos Começar!

Muito bem, estudante, esclareceremos o que é um possível risco! Identificar riscos é utilizar-se de experiências anteriores (projeto ou iteração anterior) ou técnicas de identificação de risco. Lembre-se que você deve considerar a participação dos *stakeholders* (das áreas financeira, produção, mercado, ou especialista do setor). Resgatemos aqui um exemplo: se o seu software é para o setor de agrícola, então convide especialistas em solo, clima, plantio, manejo, colheita, adubação, entre outros.

Mapeamento de risco e natureza do risco

Quando você se prepara para realizar o mapeamento de risco, a primeira providência é identificar os possíveis riscos, em seguida, classificá-los pela sua origem (processo) auxiliará na detecção das causas. Vejamos, se você elabora um mapa de prováveis problemas/impedimentos no projeto, o próximo passo é idealizar uma possível resolução para cada risco.

Dentre as técnicas para elencar riscos e construir o mapeamento, é obter situações seguindo a WBS (*Work Breakdown Structure*) tanto do projeto quanto do produto de software. Você poderá considerar fonte de riscos as reuniões (diárias, de revisão, de retrospectiva, ou planejamento de iteração/Sprint). Recorde-se que a reunião de Retrospectiva da Sprint tem como uma das finalidades elencar problemas e soluções; da mesma forma acontece na fase de Finalização/Encerramento de projeto no modelo tradicional ou sequencial.

Um exemplo de risco (quando você está elencando os riscos), relacionado ao projeto, é a “alta rotatividade de profissional de banco de dados”. Bem, nesse caso, o risco poderia estar descrito de forma diferente também, mas o importante é que o gestor do projeto tenha esse apontamento

no seu mapa de risco, assim como a descrição de qual seria o efeito, o impacto e as alternativas de soluções.

Compreendamos o que é natureza do risco! Os problemas/impedimentos no projeto de software estão relacionados a alguma natureza, segundo a ISO 31000:2018 (p. 11), “a natureza e o tipo de incertezas que podem afetar resultados e objetivos”, página 12, “natureza e valor dos ativos e recursos”; na página 13, “a natureza e magnitude das consequências”. Ou seja, os fatores e o relacionamento entre eles devem ser considerados conforme o projeto em análise: depende dos ativos e do montante das consequências. É uma atribuição intrínseca ao segmento, extensão, ou característica do projeto de software.

Após conhecer o conceito de natureza do risco, para o nosso exemplo: “alta rotatividade de profissional de banco de dados”, classificaremos como de natureza: gestão de projeto. Esse nome está relacionado com recurso ou fator, que um projeto de software necessita, e que deveria estar previsto, em algum momento do projeto, antes de sofrer impacto.

À medida que novos riscos surgem, podem ser criadas natureza de risco dependendo das consequências, ou pertencimento de recurso específico, ou do grau de incerteza. Esperamos que você compreenderá ao conhecer um exemplo de matriz de risco.

Siga em Frente...

Matriz de risco

Entende-se por matriz de risco uma relação consolidada de todos os riscos analisados e em análise, na qual os envolvidos no projeto conferem e alimentam cada risco elencado nela. O propósito da matriz de risco é a redução da probabilidade e do impacto de eventos negativos, sendo que durante a elaboração dessa matriz, possam identificar e priorizar riscos, de preferência, antes que eles aconteçam; e conste, também, o respectivo plano de ação para eles.

- Alguns aspectos analisados para cada risco que podem estar na matriz, tais como, fontes de risco, conforme cita (CAVALCANTI, 2016):
- **Instabilidade no escopo:** sem a visão clara do escopo, ou por saber que mudanças importantes ocorrerão ao longo do desenvolvimento do software; necessidade de inovação após o início do projeto também é fator crítico, porém é comum ocorrer.
- **Fornecedores:** parceiros desenvolvedores de componentes ou de equipamentos integradores podem causar impactos negativos ao projeto, desde atrasos até falhas de construção.
- **Envolvimento de partes interessadas:** falta ou indisponibilidade do *stakeholder* convededor, ou especialista em processos específicos, de requisitos funcionais complexos, podem afetar diretamente, durante a definição, no esclarecimento de dúvidas, ou na validação do produto.
- **Estimativa não realista:** comum em projeto de software, por falta de experiência ou falta de entendimento do requisito funcional; muitas vezes a estimativa é feita por pessoas que não

são os construtores. A falta de conhecimento da linguagem de programação, ou de outros recursos relacionados também afeta essa estimativa.

Dentre as classificações de riscos, podemos citar, segundo Cavalcanti (2016, p. 173), mas não atenha somente a estas, pois a natureza do projeto pode exigir outras categorias, ou até mesmo o desmembramento:

- **Técnico:** relacionado a linguagem de programação, gerenciador de banco de dados, sistema operacional, hardware, protocolos de redes, demais softwares.
- **Requisitos:** regras de negócios e funcionalidades especificadas pelo time de operações.
- **Externo:** envolvem fatores além do domínio da organização.
- **Organizacional:** todos os recursos de pares ao do time de TI, ex: financeiro, administrativo, recursos humanos, entre outros.
- **Gestão do projeto:** associado a aspectos da gestão de pessoas da TI, comunicação, integração, *stakeholders*, relação com time de operações, entre outros.

Os fatores ou classificações de risco são importantes norteadores para encontrar ou direcionar a resolução ou minimização das consequências, possíveis ou prejuízos. Ao classificar um item de risco, o projeto exigirá maior envolvimento de pessoas específicas e facilitará a mitigação e resolução.

Estudante, agora que você tem a competência em Riscos, continue se atualizando e experimentando a construção da matriz de riscos, a sua habilidade nesse quesito aumentará a qualidade do projeto de software.

Vamos Exercitar?

Estudante, você compreenderá melhor o gerenciamento de riscos. Para tanto, propomos que faça um exercício interpretando a matriz de risco, para o projeto AZQuali, auxiliando na produção do azeite. Veja a Tabela 1 que ilustra a matriz de riscos no projeto, onde se lê Categoría poderia ser Natureza.

Risco	Categoría	Causa	Ação e efeito
Invasão de privacidade	Gestão do projeto	Falta de política de segurança e de uso de Internet	Implantar políticas de segurança para evitar invasões e uso indevido de dados
Falha na projeção de receitas	Requisitos	Regras definidas sem comprovação de dados	Mitigação das regras de projeção para evitar prejuízos financeiros
Acesso para danificar os dados	Técnico	Falha na segurança e de controle de acesso	Configurar o sistema de segurança para evitar

			acesso indevido a dados
Informações incorretas	Técnico	Usuário operando o sistema inadequadamente	Bloquear usuário para evitar dúvidas e prejuízo por decisão inadequada
Vigilância sanitária impedindo a venda	Externo	Especialista não informou que deveria estar na aplicação	Implementar a configuração da regulamentação digital para evitar multas

Tabela 1 | Matriz de risco para AZQuali. Fonte: elaborada pelo autor.

Os riscos elencados podem ter sido experimentados anteriormente ou informados pelos especialistas no segmento de mercado.

A categoria classificada para cada risco contribuirá na identificação das pessoas que vão colaborar na investigação. Enquanto, a causa é a fonte de problema/impedimento.

Após a análise da causa, as ações são obrigatórias para corrigir/ajustar o que foi identificado como risco. Mesmo após as ações serem implementadas, o acompanhamento deve acontecer constantemente, para evitar prejuízo.

Saiba mais

1. Análise de riscos – matriz de probabilidade e impacto

Leia o item “Análise qualitativa de riscos”, página 182 a 192, sobre o tratamento de risco utilizando a matriz de probabilidade e impacto.

CAVALCANTI, F. R. P. [Fundamentos de gestão de projetos: gestão de riscos](#). São Paulo: Atlas, 2016.

2. Identificação de riscos – matriz de riscos

Leia o item 252 Identificação de riscos, página 199, que mostra uma tabela de identificação de riscos com o respectivo tratamento.

FILHO, W. P. P. [Engenharia de Software - Projetos e Processos](#), v. 2. 4ed., Rio de Janeiro: LTC, 2019.

3. Tratamento de riscos – análise quantitativa

Leia o item “Análise quantitativa”, página 195 a 198, para conhecer como efetuar a análise de valores de probabilidade. Em seguida, ouça o [Podcast Interpretando a Análise Quantitativa e o Valor Esperado](#).

CAVALCANTI, F. R. P. [Fundamentos de gestão de projetos: gestão de riscos](#). São Paulo: Atlas, 2016.

Referências

ABNT. **Gestão de Riscos – diretrizes**. NBR ISO 31000:2018. Associação Brasileira de Normas Técnicas. 2018.

CAMARGO, R.; RIBAS, T. **Gestão ágil de projetos**. São Paulo: Saraiva Educação, 2019.

CAVALCANTI, F. R. P. **Fundamentos de gestão de projetos: gestão de riscos**. São Paulo: Atlas, 2016.

MAXIMIANO, A. C. A.; VERONEZE, F. **Gestão de projetos: preditiva, ágil e estratégica**. Barueri: Atlas, 2022.

PRESSMAN, R. S.; MAXIM, B. R. **Engenharia de software**. Porto Alegre: AMGH, 2021.

Aula 2

Gestão da Qualidade em Projeto de Software

Gestão da Qualidade em Projeto de Software

Este conteúdo é um vídeo!



Para assistir este conteúdo é necessário que você acesse o AVA pelo computador ou pelo aplicativo. Você pode baixar os vídeos direto no aplicativo para assistir mesmo sem conexão à internet.

Dica para você

Aproveite o acesso para baixar os slides do vídeo, isso pode deixar sua aprendizagem ainda mais completa.

Estudante, esta videoaula foi preparada especialmente para você. Nela, você irá aprender conteúdos importantes para a sua formação profissional. Vamos assisti-la?
Bons estudos!

Ponto de Partida

Boas-vindas aos critérios fundamentais da qualidade, um aspecto facilitador da evolução e da inovação. Estudante, você consegue imaginar um processo sem considerar a qualidade? Apenas para você perceber a relação da qualidade com o processo: entendamos como seria uma pizza montada, assada e servida sem critérios de qualidade?

O processo de desenvolvimento de software também recorre a critérios da qualidade, segundo IEEE e ISO, utilizando-se de padrões impostos por clientes, outros *stakeholders*, ou livremente escolhidos pela organização, em seu projeto de software.

Vamos nos posicionar frente a uma situação da realidade para compreender os conceitos da qualidade em projeto de software, no segmento da produção de azeite, OliVirgem, na qual os aplicativos de software deverão ser desenvolvidos conforme evolução da olivicultura.

Você perceberá que os processos de software vão maturando, simultaneamente ao processo da olivicultura. Na olivicultura, é necessário manejos rigorosos contra pragas e doenças, da mesma forma que tarefas de inspeção são praticadas contra bugs no código-fonte.

Agora é contigo, tente comparar as atividades que está executando no seu cotidiano para compreender como a gestão da qualidade em software contribuirá com a construção de softwares melhores.

Vamos Começar!

Estudante, inicialmente, saiba que gerenciar a qualidade requer muito empenho e disciplina. A regularidade na execução das atividades é extremamente fundamental para evidenciar a evolução da qualidade em software. A habilidade em executar tarefas proporcionará repetitivamente a manutenção de indicadores da evolução da qualidade do produto ou do processo.

Sistema de qualidade do software

Naturalmente, todos os envolvidos buscam a qualidade, porém um resultado formal se dará por um sistema definido e seguido de forma disciplinar. Estabelecendo um sistema de qualidade, gestor, *stakeholders* e time de TI se comprometerão com a qualidade aceitável, sendo possível medir e melhorar periodicamente.

Um sistema da qualidade iniciará com os valores do projeto de software que o time entregará, sendo que todos os recursos serão utilizados para criar os valores da entrega e devem ser adequadamente gerenciados. Entende-se por valores as funcionalidades que proporcionam inovação, resultado positivo ou satisfação e lucratividade para as organizações.

Dentre os recursos são: escopo que compreende em funcionalidade e requisitos do sistema; tempo/prazo no qual os envolvidos desempenharão para construir o software; o próprio time de TI, recursos computacionais e depois itens compõe o custo. E, por fim, a divulgação de resultados atingidos representando a qualidade em forma de indicadores e relações dos resultados que representam os valores estabelecidos inicialmente. A Figura 1 demonstra a relação que existe entre três elementos fundamentais do desenvolvimento de software.

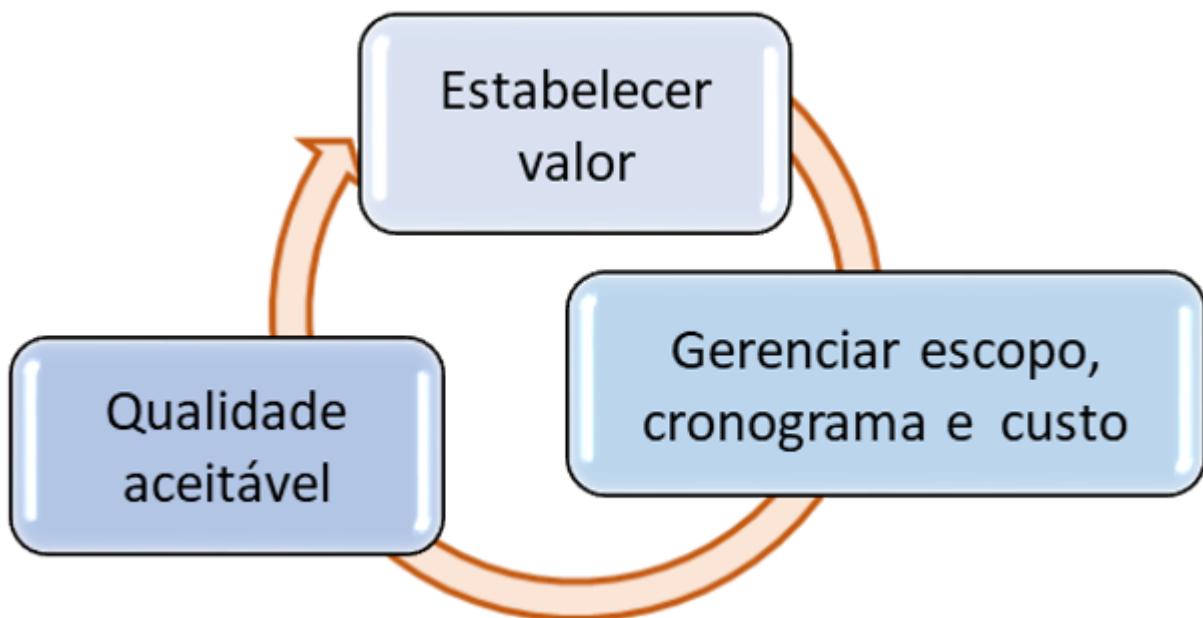


Figura 1 | Triângulo Ágil. Fonte: adaptado de Camargo (2019, p. 108).

Ciclicamente, os valores podem sofrer modificações; um padrão de qualidade desejado em um determinado projeto ou período pode mudar no próximo projeto, ou ao longo do mesmo projeto. Aceitar um padrão de qualidade é relativo aos valores estabelecidos e poderá sofrer modificações conforme o escopo, cronograma e custo.

Métricas de qualidade

Entendamos que a métrica, segundo IEEE, é uma medida quantitativa do grau com que um sistema, componente ou processo possui um dado atributo. Para IEEE, métricas utilizam termos fundamentais, analisemos:

- **Atributo:** propriedade física ou abstrata mensurável de uma entidade. Ex: o software possui histórias implementadas, então a quantidade de histórias entregues em uma Sprint é a capacidade do time.

- **Métrica:** medida quantitativa do grau com que um sistema, componente ou processo possui um dado atributo. Ex: correlação entre quantidade de histórias desenvolvidas na Sprint.
- **Medição:** ato ou processo de atribuir um número, ou uma categoria a uma entidade para descrever aquela entidade. Ex: processo de apuração/apontamento da medida.
- **Medida:** um número, extensão ou quantidade resultante de uma medição. Ex: quantidade de horas utilizada para implementar os itens da Sprint.

As métricas das qualidades têm propostas para garantir que os processos serão ajustados para minimizar as falhas, ou seja, além de conter o crescimento de erros, devem criar procedimentos que impeçam a aparição de novos defeitos.

Siga em Frente...

Maturidade em projeto de software

Se a métrica aponta os problemas ou necessidades de melhoria, a maturidade é indicação de que o processo tem estágios diferentes. Inicialmente, a organização é considerada sem classificação, à medida que se criam padrões e critérios, o enquadramento se torna possível rumo à maturidade em melhoria contínua por atitudes de gestão dos processos.

Estudante, faça uma reflexão sobre o modelo de maturidade, o qual ocorre tanto individualmente quanto organizacionalmente, no qual:

- Se inicia com habilidades mínimas e pouco domínio em conceitos/ferramentas;
- Se transita em aprendizado, experimentação, treino da disciplina e dos indicadores, com avaliação de resultados;
- Atinge um grau de aprimoramento contínuo, em processos organizacionais, junto da administração da proficiência dos profissionais.

A maturidade se alcança e é importante permanecer investindo constantemente no aprendizado da inspeção das práticas.

Para o CMMI, o modelo de maturidade tem os seguintes níveis:

- 1 – **Executada:** as atividades do projeto para entregar o produto.
- 2 – **Controlada:** as atividades e implantando processos de desenvolvimento.
- 3 – **Padronizada:** o processo: soluções, validações, gestão, gerenciamento de riscos.
- 4 – **Medido:** o processo está sendo gerenciado com base em métrica.
- 5 – **Em otimização:** os processos em melhoria.

Estudante, ao se tornar um profissional competente em desenvolvimento de software, o seu desafio é estabelecer a meta da melhoria e aperfeiçoamento constante em processos, metodologias e ferramentas. Continue estudando novidades de Projeto de Software.

Vamos Exercitar?

A partir da definição de uma métrica da qualidade, de alguma maneira, se instala o desejo de controlar e melhorar o processo.

Portanto, entenda como a OliVirgem melhora o processo conforme os apontamentos dos indicadores, mostrados na Tabela 1.

Termos	Descrição	Orientação e resultado
Atributo	O motivo pelo qual está sendo contada a quantidade de falhas no processo de desenvolvimento	Quantidade de falhas encontradas na Sprint
Métrica	Como a medição deve ser realizada para apurar a medida , visando suprir o atributo da qualidade	Falhas na execução no produto de software na Sprint, em implementações da Sprint Backlog
Medição	Efetuar a contagem da quantidade de falhas na Sprint, em teste de produto	Classificação das falhas encontradas na fase de teste do produto da Sprint
Medida	Quantidade de falhas na Sprint	8

Tabela 1 | Métrica da Qualidade OliVirgem. Fonte: elaborada pelo autor.

Toda métrica será aproveitada, criando expectativa de melhoria contínua. Devemos lembrar que o time de desenvolvimento ficará engajado no melhoramento da qualidade na execução das atividades e na entrega do produto, desde que seja implementado com sistematização.

Sendo assim, é fundamental adotar um modelo que estabeleça mecanismos para impulsionar o senso de qualidade. Dependendo da situação, o gestor necessitará de modelos mais robustos como o SPI, CMMI, entre outros.

Saiba mais

Estudante, aproveite e aprofunde conhecendo sobre a maturidade e métrica da qualidade.

1. Compreenda sistema da qualidade com o exemplo da Reunião de Revisão

Aprofunde o entendimento da qualidade num processo completo, leia o item 2.3.3 Reunião de revisão, página 92.

PROJETO DE SOFTWARE

FILHO, W. P. P. [Engenharia de software - Projetos e Processos](#), v. 2, 4 ed., Rio de Janeiro: LTC, 2019. [Minha Biblioteca]

2. Evitando o fracasso em projeto por meio da qualidade

Leia o item 21.1 Fatores Críticos de Sucesso, página 194 a 197, para conhecer melhor os fatores desfavoráveis em projetos e a importância da mudança de cultura.

MAXIMIANO, A. C. A.; VERONEZE, F. [Gestão de projetos: preditiva, ágil e estratégica](#). Barueri: Atlas, 2022. [Minha Biblioteca]

3. Maturidade organizacional

Leia o item 21.3 Maturidade Organizacional, página 198, para conhecer a maturidade rumo à qualidade.

MAXIMIANO, A. C. A.; VERONEZE, F. [Gestão de projetos: preditiva, ágil e estratégica](#). Barueri: Atlas, 2022. [Minha Biblioteca]

4. PMBoK, COBIT, ITIL, Seis Sigma – contribuindo com a maturidade

Leia o item 4. Outros Modelos, página 68, e conhecerá como esses modelos contribuem na maturação do processo da organização.

FILHO, W. P. P. [Engenharia de software – Projetos e processos](#), v. 2, 4 ed., Rio de Janeiro: LTC, 2019. [Minha Biblioteca]

Referências

CAMARGO, R.; RIBAS, T. **Gestão ágil de projetos**. São Paulo: Saraiva Educação, 2019.

CAVALCANTI, F. R. P. **Fundamentos de gestão de projetos**: gestão de riscos. São Paulo: Atlas, 2016.

FILHO, W. P. P. **Engenharia de software - Projetos e processos**, v. 2, 4 ed., Rio de Janeiro: LTC, 2019.

MAXIMIANO, A. C. A.; VERONEZE, Fernando. **Gestão de projetos**: preditiva, ágil e estratégica. Barueri: Atlas, 2022.

PRESSMAN, R. S.; MAXIM, B. R. **Engenharia de software**. Porto Alegre: AMGH, 2021.

Aula 3

Documentação do projeto

Documentação do projeto

Este conteúdo é um vídeo!



Para assistir este conteúdo é necessário que você acesse o AVA pelo computador ou pelo aplicativo. Você pode baixar os vídeos direto no aplicativo para assistir mesmo sem conexão à internet.

Dica para você

Aproveite o acesso para baixar os slides do vídeo, isso pode deixar sua aprendizagem ainda mais completa.

Estudante, esta videoaula foi preparada especialmente para você. Nela, você irá aprender conteúdos importantes para a sua formação profissional. Vamos assisti-la?
Bons estudos!

Ponto de Partida

Boas-vindas estudante! Você perceberá que a documentação de um projeto de software é imprescindível para alinhamento, ou para criar um vínculo de confiança.

Mas saiba também que documentos extensos são inúteis e podem até gerar dúvidas.

Imagine-se em um projeto para o desenvolvimento de aplicações de uma oficina para carros! O seu desafio é compreender quais documentações serão fundamentais, quais auxiliarão ou não, em um ambiente tão movimentado, de procedimentos operacionais diversos. Porém, o sistema deverá ser desenvolvimento para atender à demanda desse cenário dinâmico e carente de documentação.

Dentre os processos da MecanDoc, o gestor necessita de orçamento preciso, assim que o diagnóstico seja apontado pelo consultor da recepção. A avaliação inicial deve prever procedimentos anteriores do veículo na MecanDoc, perfil do motorista, além da própria condição do carro.

É comum e necessária a documentação para iniciar o desenvolvimento desse projeto da MecanDoc, acrescer documentos ao longo do projeto e a conclusão, também, deve prever alguns

registros decisivos e importantes. Quais e como eles devem ser criados e atualizados?

Lembre-se que a documentação será suporte para o desenvolvimento do software e para garantir que as regras de negócios sejam implementadas corretamente.

Vamos Começar!

Estudante, inicialmente, pense em uma situação bastante comum no desenvolvimento de software. Quando efetuamos uma reunião com várias pessoas, com assuntos importantes, relativos a requisitos decisivos para um procedimento/funcionalidade que será implementado em software, como é essa reunião? Vamos dividi-la em três momentos, inicialmente exige-se uma preparação com materiais e estratégias; em seguida, durante a reunião, a condução com foco no assunto é manter as decisões organizadas. Por fim, cita Cavalcanti (2016, p. 147), “o depois da reunião diz respeito à documentação, através de uma ata, e ao acompanhamento, por parte do GP, das ações definidas na reunião”.

Você percebe que a documentação deve existir em momentos importantes do projeto, afinal, juntar as decisões (iniciais e as mudanças) das regras, funcionalidades, *design*, procedimentos, se torna obrigatório para dirimir dúvidas futuras, ao longo do ciclo de vida do software.

Documentação como elemento de qualidade

As organizações que seguem a disciplina de gestão do conhecimento, independente da metodologia tradicional, sequencial, ou ágil, usam lições aprendidas. Essa documentação caminha em paralelo ao projeto de software, porém tem enorme influência na qualidade em atividades relacionadas ao desenvolvimento de software. A criação e manutenção desse documento tem menção em reuniões de Planejamento da Sprint, Reunião de Retrospectiva da Sprint e fase de Encerramento de Projetos.

Com podemos encontrar “registro das lições aprendidas pode ser feito de diversas formas: a documentação do projeto será revisada detalhadamente”, em Cavalcanti (2016, p. 215), o que torna o documento um importante elemento de qualidade, seguindo o propósito da organização com maturidade em seus processos.

Por que e quando documentar

Entender quando a documentação é importante e o motivo pelo qual ela se torna imprescindível, te tornará um profissional organizado e focado em auxiliar ou produzir documentos úteis. Digamos, que nem toda documentação é tão útil, pois se for extensa com exagero, ou redigida em uma linguagem pouco sucinta, ou com informações além das essenciais, ela se transformará em material inacessível. Enfim, cada projeto e cenário exigirão uma formalização compatível com os seus envolvidos.

Uma situação comum é “às vezes, os sistemas legados têm projetos inextensíveis, código de difícil entendimento, documentação deficiente ou inexistente, casos de teste e resultados que nunca foram documentados, um histórico de alterações mal gerenciado” conforme citou (PRESSMAN, 2021). Lembre-se que o seu sistema sairá da situação “em desenvolvimento” para ser um sistema legado.

Uma característica importante, se tratando de documentação, é como parte integrante da comunicação. Como seria a comunicação sem a parte documentada ou formalizada? Durante o plano de comunicação de um projeto de software, o gestor deve definir como e quando as documentações técnicas do produto serão disponibilizadas/encaminhadas ou divulgadas.

Você comprehende que alguns eventos, para qualquer modelo de processo (ágil ou tradicional), exigem que a documentação seja devidamente guardada. Por outro lado, o momento crucial relacionado à documentação é a fase de encerramento, no qual se juntam as várias documentações geradas no projeto.

Na fase de encerramento de um projeto de software, na qual se faz a verificação final das entregas, a formalização dessa etapa deve acontecer. Conforme Cavalcanti (2016, p. 214), é momento para o “arquivamento de toda a documentação gerada, física e eletrônica”.

Siga em Frente...

Riscos pela ausência de documentação

Uma das documentações essenciais em projeto de software está relacionada com o comprometimento entre *stakeholders*, ou envolvidos. Desde recursos financeiros, cronograma de entrega e de remuneração, até padrões de qualidade, requisitos, *feedbacks* técnicos ou administrativos.

Na fase de validação, ou enquanto os usuários estão avaliando o software recém-implantado, o time de TI deve receber um documento de *feedback*, conforme Pressman (2021, p. 184) “sem a documentação do que funcionou e do que não funcionou, é difícil para os engenheiros de software decidirem quando devem inovar e quando devem usar uma arquitetura criada anteriormente.”

A ausência de listas de conferência para revisão em projetos de softwares fatalmente, seria prejudicial para manter a qualidade no processo de desenvolvimento.

Como se verifica em Filho (2019, p. 96), “altamente recomendável que os autores façam revisões individuais prévias, usando listas de conferência, o que já permite a eliminação dos defeitos mais evidentes”. Nesse exemplo, se percebe a importância para evitar riscos em projetos.

Vamos Exercitar?

Assim que os requisitos do sistema começaram a ser definidos, o gestor da MecanDoc pediu aos seus consultores escolherem um procedimento prioritário para implementar, conforme o ponto de vista, individualmente. São três pessoas que estão trabalhando nessa função, cada um escreveu um procedimento diferente do outro. Vejamos como foi:

Consultor 1	Consultor 2	Consultor 3
Assim que inicia o atendimento, ao adicionar o carro, pela placa, o sistema deve mostrar os 3 últimos serviços realizados, com data, descrição do serviço e nome do consultor.	Durante a execução do serviço, ao escolher a peça a ser trocada no carro, o sistema deve mostrar a fotografia, descrição, fornecedor da peça e modo de instalação.	Ao finalizar o serviço, o sistema deve mostrar um checklist de verificação do que o testador do veículo deve verificar, conforme o serviço realizado, registrando data, hora e nome do testador que ficará vinculado à ordem de serviço.

Tabela 1 | Documentação de funcionalidades. Fonte: elaborada pelo autor.

Mesmo sendo 3 pessoas na mesma função, trabalhando na mesma empresa, percebe-se que cada uma externaliza vontade e prioridade distintas. Portanto, esse é um registro de documentação importante para administrar o Product Backlog, antes mesmo de o projeto iniciar. Sabe-se que a dinâmica de inclusão/modificação de funcionalidades será intensa, em função da experiência que cada consultor terá.

Fica aqui um exemplo de que uma documentação sucinta e objetiva, no momento adequado, proporcionará segurança ao Product Owner e ao gestor do projeto no monitoramento e priorização das atividades do projeto.

Saiba mais

Agora chegou a oportunidade para ler outros materiais diretamente relacionados às documentações.

1. Compreenda a documentação na fase de transição e implantação

Leia o item 2.5.2 Planejamento da implantação, página 373, o qual detalha a fase de implantação de sistemas.

FILHO, W. P. P. [Engenharia de Software - Projetos e Processos](#) - Vol. 2. 4. Ed. Rio de Janeiro: LTC, 2019. [Minha Biblioteca]

PROJETO DE SOFTWARE

2. A documentação em valores ágil

Leia o item 21.2.2 Valores, página 197, para compreender como a documentação/registro são imprescindíveis, inclusive na metodologia ágil.

MAXIMIANO, A. C. A.; VERONEZE, F. [Gestão de projetos: preditiva, ágil e estratégica](#). Barueri: Atlas, 2022. [Minha Biblioteca]

3. Teste da documentação e dos recursos de ajuda

Leia o item 21.12 Teste da documentação e dos recursos de ajuda, página 434, para compreender a documentação como recurso de ajuda.

PRESSMAN, R. S.; MAXIM, B. R. [Engenharia de software](#). Porto Alegre: AMGH, 2021. [Minha Biblioteca]

Referências

CAMARGO, R.; RIBAS, T. **Gestão ágil de projetos**. São Paulo: Saraiva Educação, 2019.

CAVALCANTI, F. R. P. **Fundamentos de gestão de projetos: gestão de riscos**. São Paulo: Atlas, 2016.

FILHO, W. P. P. **Engenharia de software - Projetos e Processos**, v. 2, 4 ed. Rio de Janeiro: LTC, 2019.

MAXIMIANO, A. C. A.; VERONEZE, F. **Gestão de projetos: preditiva, ágil e estratégica**. Barueri: Atlas, 2022.

PRESSMAN, Roger S.; MAXIM, Bruce R. **Engenharia de software**. Porto Alegre: AMGH, 2021.

Aula 4

Ferramentas para gestão de projetos

Ferramentas e Qualidade em Projeto de Software

Este conteúdo é um vídeo!



Para assistir este conteúdo é necessário que você acesse o AVA pelo computador ou pelo aplicativo. Você pode baixar os vídeos direto no aplicativo para assistir mesmo sem conexão à internet.

Dica para você

Aproveite o acesso para baixar os slides do vídeo, isso pode deixar sua aprendizagem ainda mais completa.

Estudante, esta videoaula foi preparada especialmente para você. Nela, você irá aprender conteúdos importantes para a sua formação profissional. Vamos assisti-la? Bons estudos!

Ponto de Partida

Olá, estudante, o seu projeto está aprovado para começar e ainda não sabe como acompanhar as atividades, ou como comunicar essa decisão? Acreditamos que você esteja ansioso para saber quais informações são relevantes para informar aos *stakeholders* (afinal não é necessário compartilhar todos os detalhes, a maioria não teria nem tempo para ler muita coisa).

O intuito é que você compreenda que existe diferenças no acompanhamento de projetos com ciclo de vida sequencial e ágil:

- Se no modelo sequencial, relembrmos, o planejamento é concebido com todos os requisitos conhecidos; e na fase seguinte, todos eles são elicitados, tornando a fase de análise e de arquitetura rica em detalhes. De certa forma, isso facilita a elaboração do design/componentização e, consequentemente, favorece a construção do código-fonte;
- Por outro lado, lembre-se que no modelo ágil os requisitos são conhecidos na fase de planejamento, porém somente os requisitos de iteração que serão elicitados, na atividade de planejamento da iteração. Portanto, na segunda iteração, o design/componentização passará pela etapa de reestruturação e o código-fonte pela refatoração.

Posto isso, queremos que você seja o gerente do projeto TicketExpress, um portal de venda de ingressos de eventos/shows, no qual perceberá as diferenças entre os modelos sequencial e ágil. Saberá escolher as informações a serem divulgadas aos *stakeholders*, como mostrar e quando enviar.

Vamos Começar!

Bem, você sabe que as ferramentas para a gestão de projeto de software devem auxiliar o gestor no planejamento, durante a execução das atividades, mostrando o avanço/evolução; até a conclusão/encerramento do projeto.

Ferramentas para modelos sequencial e ágil

Algumas ferramentas surgiram ainda quando os projetos, na grande maioria, eram gerenciados pelo modelo sequencial. Com o passar do tempo, otimizações foram ocorrendo em detrimento das necessidades de melhoria no monitoramento do projeto, porém as ferramentas não foram totalmente abandonadas.

O cronograma no formato Gantt é largamente utilizado quando o ciclo de vida Waterfall era a melhor opção para gerenciar e comunicar a situação da evolução do projeto, como se percebe na figura 1. Essa ferramenta pode ser utilizada por gestões do modelo ágil também.

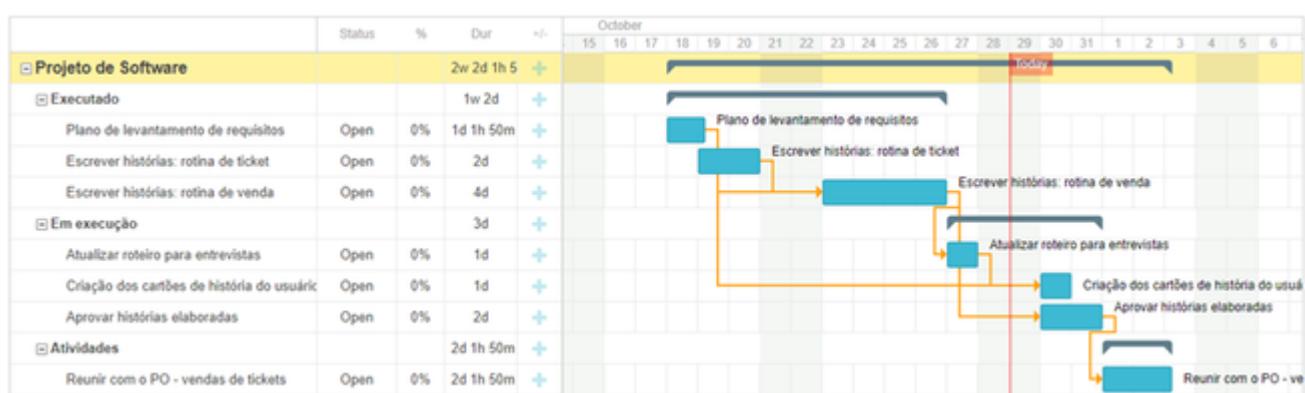


Figura 1 | Cronograma em Gantt. Fonte: elaborada pelo autor.

A visualização do Gantt é sugestiva quanto às atividades dependentes, como aquelas que podem ser executadas concomitantemente; outra característica é o agrupamento de atividades por módulo, ou por componente do projeto, ou por estágio. Uma das vantagens do gráfico de Gantt é quando ocorre mudança no prazo de uma atividade; ao ajustar a duração de uma atividade, as demais são ajustadas automaticamente por estarem vinculadas.

Uma visão, estilo Kanban, utilizada pelo modelo ágil tem como característica indicar quais atividades estão em execução, ao mesmo tempo, quais estão prontas para serem iniciadas e as que foram concluídas.

No modelo ágil, a gestão do tempo de projeto se torna uma responsabilidade menos onerosa ao gerente (ou Scrum Master). A metodologia estabelece iterações de curtíssimo prazo, 2 a 4 semanas, contribuindo imensamente na percepção de morosidade/rapidez na progressão ao longo do projeto.

Na Figura 2, é possível identificar a atividade que está com algum problema/impedimento necessitando da intervenção do Scrum Master para resolver.

PROJETO DE SOFTWARE

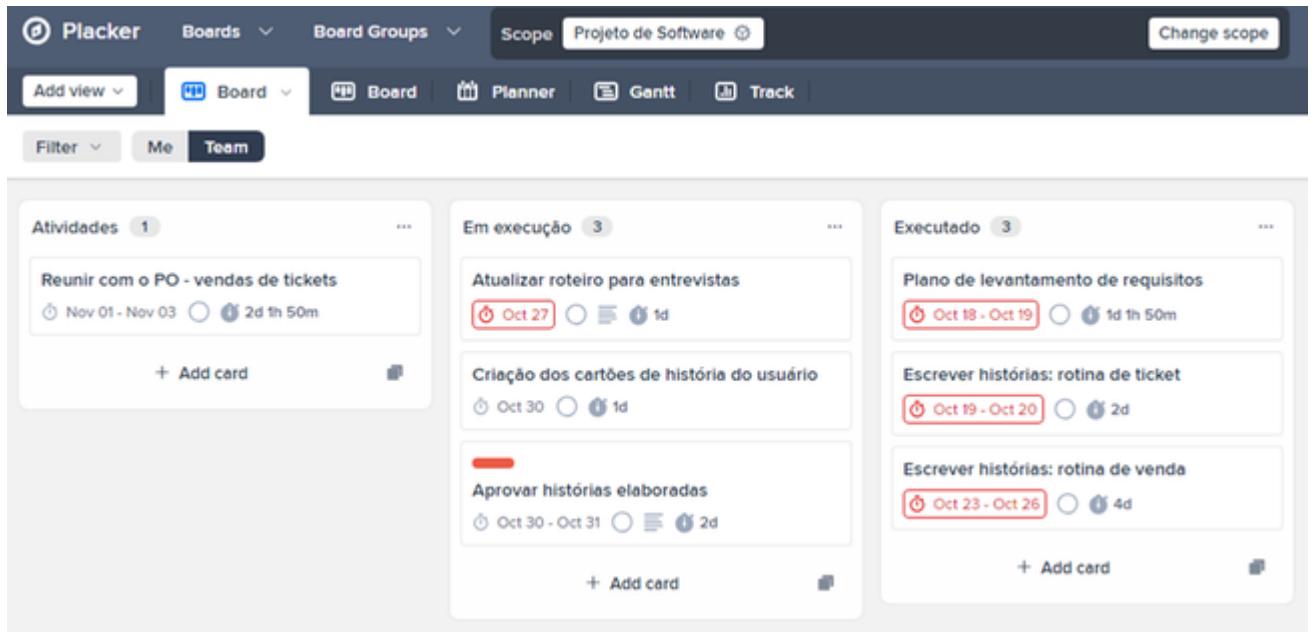


Figura 2 | Quadro de atividades. Fonte: elaborada pelo autor.

O quadro de atividades é uma ferramenta simples e objetiva, ficando à exposição o tempo todo, os envolvidos conseguem acompanhar e colaborar caso identifiquem problemas. No exemplo há três colunas, porém pode ser adotado o uso de mais colunas.

Algumas etapas do processo de desenvolvimento de software podem ser expressas por uma árvore, chamado de *Work Breakdown Structure* (WBS) ou Estrutura

Analítica do Projeto (EAP), conforme ilustra a Figura 3, uma espécie de decomposição hierárquica do processo de desenvolvimento de software.

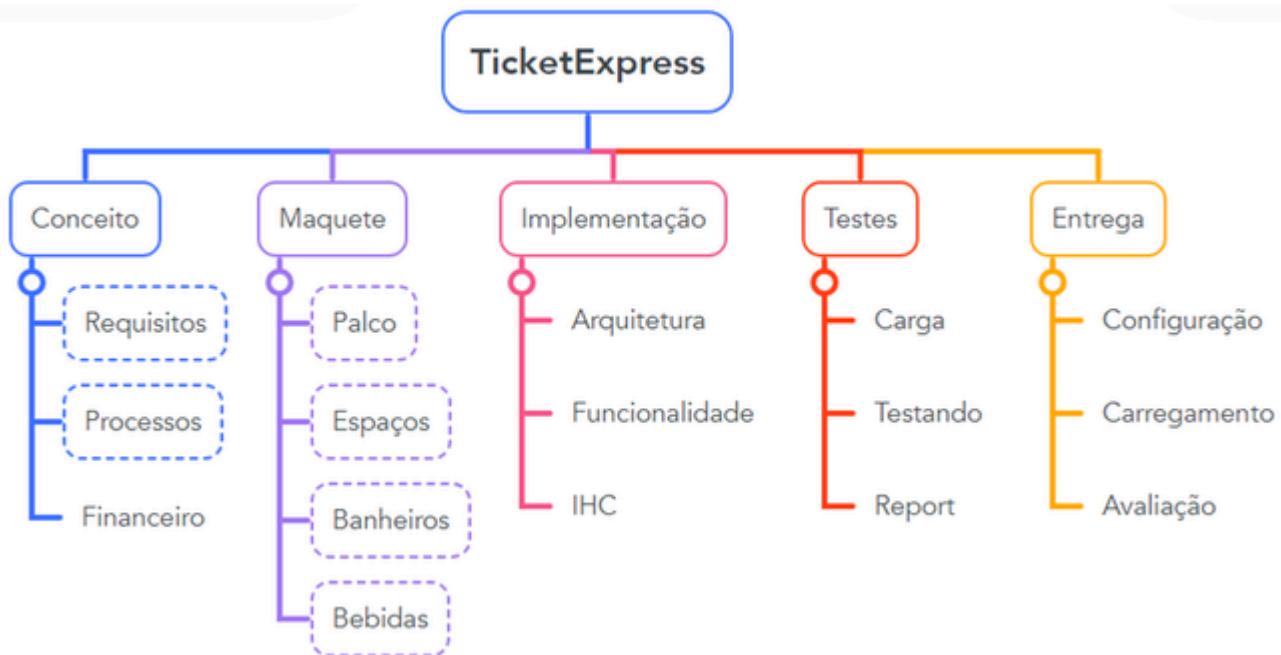


Figura 3 | Estrutura Analítica do Projeto. Fonte: elaborada pelo autor.

A WBS demonstra e mantém os *stakeholders* comunicados quanto ao andamento do projeto. Nesse exemplo, é mostrado que 5 atividades de projeto já foram concluídas, em linha pontilhada. O item “Maquete” apresentada na WBS do projeto ilustra a implementação do mapeamento do local do evento, no qual a aplicação permitirá interação com os usuários, para escolher/navegar virtualmente.

Siga em Frente...

Gestão da comunicação

Estudante, você percebe que as ferramentas que possuem aspectos visuais são informativas e agora compreendemos também como elas são fonte da comunicação. Isso mesmo, os gráficos e diagramas que auxiliam o gestor de projeto e, também, são um artefato importante na comunicação.

Você não deve se esquecer que os *stakeholders* são aqueles que merecem receber as informações representativas pela comunicação, conforme o seu envolvimento. Um patrocinador necessita saber da evolução do cronograma e dos valores gastos, periodicamente, com a projeção para o restante do projeto.

Para o time de TI, o progresso da iteração ou do projeto é extremamente importante. O processo de comunicação deve prever que o *stakeholder* que recebe a informação irá reagir (ou não) assim que interpretar o comunicado. Sendo assim, fique atento para elaborar um formato

PROJETO DE SOFTWARE

adequado da informação, seja por texto, gráfico ou voz/vídeo. A mídia utilizada para se comunicar também tem influência na interpretação da mensagem.

Outro fator importante é o momento que a informação deve chegar até o destinatário. Imagine, você no trânsito passar por um radar eletrônico fotográfico e logo na sequência ler uma placa “Passou pelo radar há 100 metros”, além de ser uma situação absurda, não seria adequada.

Um diagrama bastante útil no progresso da iteração do projeto é o Burndown, no qual é mostrado o avanço em cada Sprint, com base no total previsto. Como ilustrado na Figura 4, na Sprint 1 foram realizados 9 das 53 (aqui representada por Story Points), na Sprint 2 foram realizados mais 13 (somando 22), assim sucessivamente, até a Sprint 5 quando se consegue concluir o projeto entregando as 53 Story Points.

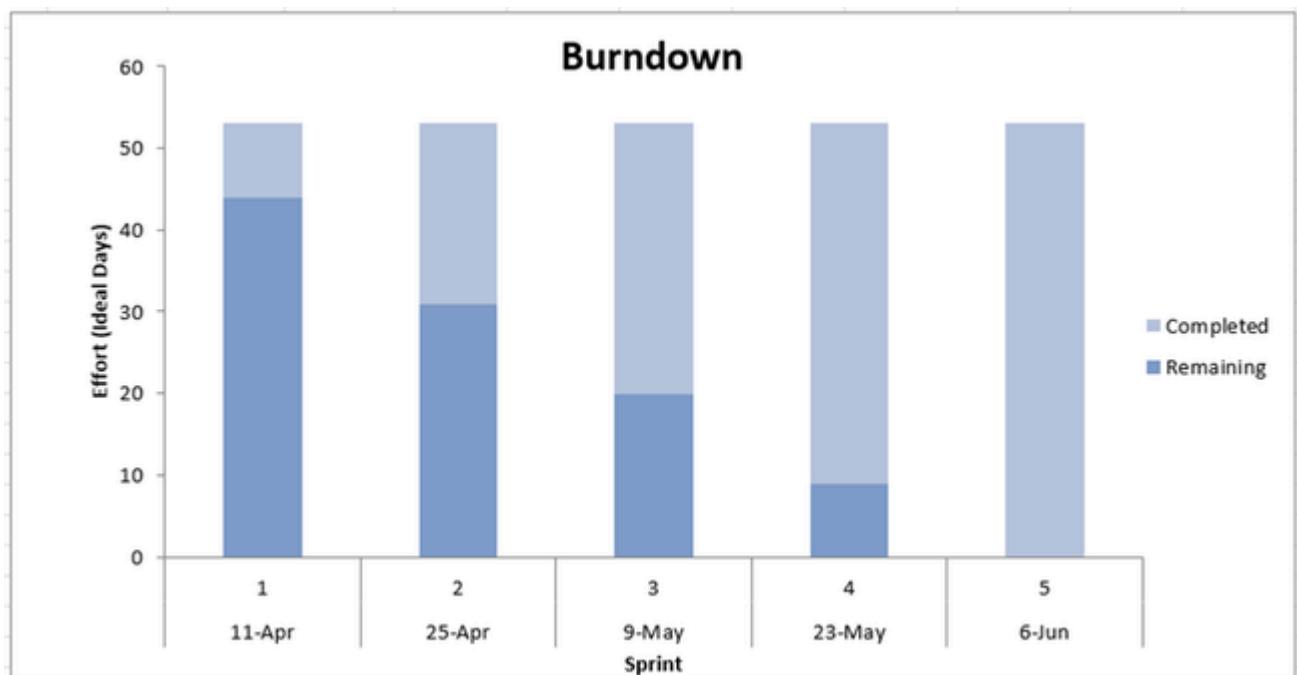


Figura 4 | Gráfico Burndown (por Story Points). Fonte: elaborada pelo autor.

Agora você está preparado para acompanhar seu Projeto de Software com maior eficiência, como também garantir uma comunicação adequada a todos os *stakeholders*.

Vamos Exercitar?

Estudante, retomamos o cenário da TicketExpress para apresentarmos o uso de ferramentas em projeto de software.

Uma árvore representada por uma WBS orienta o desenvolvimento dos requisitos, funcionando como um “mapa” das funcionalidades ou de operações, exprimindo a rotina ou práticas do usuário do sistema.⁴

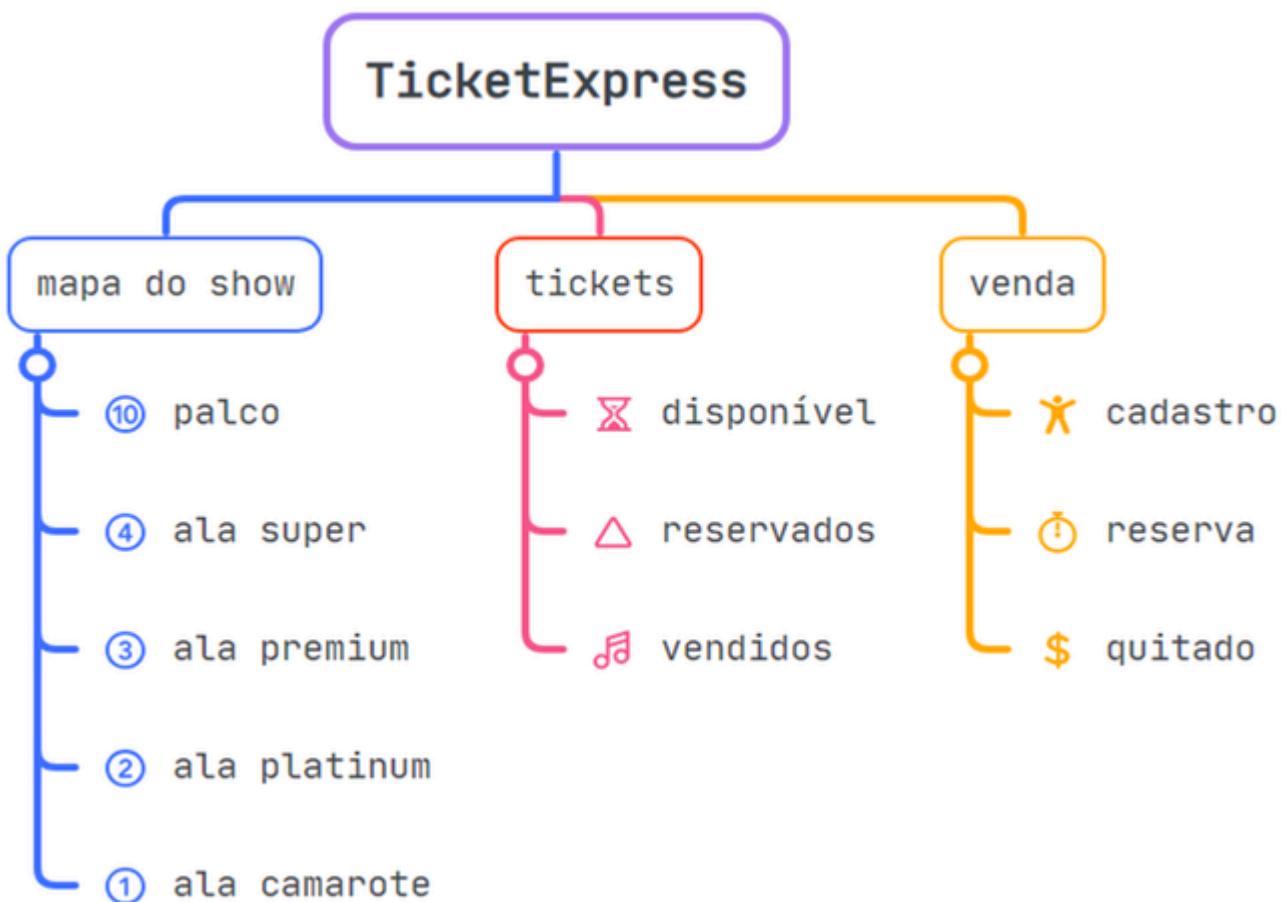


Figura 5 | WBS das funcionalidades do sistema. Fonte: elaborada pelo autor.

Esta figura ilustra o resultado de uma ferramenta que representa como o site de venda de ingressos será exibido/navegado pelo frequentador do evento. Experimente a ferramenta [MindMeister](#).

Saiba mais

1. Ferramenta de gestão e comunicação - Burndown

Leia o item 9.2.1 Monitorando a Release, página 180 a 186. Você vai compreender como o gestor poderá se beneficiar com a ferramenta Burndown, utilizando-a na comunicação eficiente também.

CAMARGO, R.; RIBAS, T. [Gestão ágil de projetos](#). São Paulo: Saraiva Educação, 2019. [Minha Biblioteca]

2. Entenda a construção de uma Estrutura Analítica

PROJETO DE SOFTWARE

Leia o item 4.3.2 Detalhamento do escopo e WBS, página 28. A EAP pode ser utilizada para representar as atividades/fases do projeto, ou para exprimir as partes do produto de software.

MAXIMIANO, A. C. A.; VERONEZE, F. [Gestão de projetos: preditiva, ágil e estratégica](#). Barueri: Atlas, 2022. [Minha Biblioteca]

3. Habilidades na comunicação evita conflitos e favorece a harmonia em projetos

Leia o item 3 A Relevância das habilidades interpessoais e de equipe no gerenciamento das comunicações em projetos ágeis, página 6.

DIAS, P. N.; CUNHA, P. Relação entre habilidades interpessoais e de equipe e o gerenciamento das comunicações em projetos ágeis. [Boletim do Gerenciamento](#), [S.I.], v. 33, n. 33, p. 1-10, nov. 2022. ISSN 2595-6531.

Referências

CAMARGO, R.; RIBAS, T. **Gestão ágil de projetos**. São Paulo: Saraiva Educação, 2019. [Minha Biblioteca]

CAVALCANTI, F. R. P. **Fundamentos de gestão de projetos: gestão de riscos**. São Paulo: Atlas, 2016. [Minha Biblioteca]

MAXIMIANO, A. C. A.; VERONEZE, F. **Gestão de projetos: preditiva, ágil e estratégica**. Barueri: Atlas, 2022. [Minha Biblioteca]

PRESSMAN, R. S.; MAXIM, B. R. **Engenharia de software**. Porto Alegre: AMGH, 2021. [Minha Biblioteca]

Aula 5

Encerramento da Unidade

Videoaula de Encerramento

Este conteúdo é um vídeo!

Para assistir este conteúdo é necessário que você acesse o AVA pelo computador ou pelo aplicativo. Você pode baixar os vídeos direto no aplicativo para assistir mesmo sem conexão à internet.

PROJETO DE SOFTWARE



Dica para você

Aproveite o acesso para baixar os slides do vídeo, isso pode deixar sua aprendizagem ainda mais completa.

Estudante, esta videoaula foi preparada especialmente para você. Nela, você irá aprender conteúdos importantes para a sua formação profissional. Vamos assisti-la?
Bons estudos!

Ponto de Chegada

Olá, estudante!

Para desenvolver a competência desta unidade, a qual é aplicar os conceitos de qualidade e gestão de riscos em ferramentas de gestão de projeto de software, você deverá conhecer primeiramente os conceitos fundamentais do mapeamento de riscos, maturidade em projeto de software, gestão da comunicação e documentação como elemento da qualidade.

O seu conhecimento em mapeamento de riscos vai permitir que você relate as causas de problemas e as respectivas consequências, porquanto idear ações paliativas e preventivas para diminuir a possibilidade de prejuízo. Saiba que a sua habilidade em gestão de riscos é uma vantagem para desenvolver softwares com qualidade e no prazo estimado.

É sabido que devemos evitar documentação extensa do projeto ou do produto! No entanto, saber produzir registros sucintos, que informam as partes interessadas e sinalizam a produtividade e a qualidade nas atividades, é bem-vindo em projetos de software.

Ainda nessa unidade, você encontra explanações relevantes para diferenciar e associar o uso da WBS/EAP com cronograma, como ferramentas para gestão de projetos. Por meio dos artefatos das ferramentas, você consegue explicar ao patrocinador e demais *stakeholders* como o projeto está sendo monitorado e gerenciado.

É Hora de Praticar!



Este conteúdo é um vídeo!

Para assistir este conteúdo é necessário que você acesse o AVA pelo computador ou pelo aplicativo. Você pode baixar os vídeos direto no aplicativo para assistir mesmo sem conexão à internet.

PROJETO DE SOFTWARE

Bem, estudante, agora é o momento para você refletir sobre a qualidade, a comunicação e os riscos em projetos de software. Então se imagine sendo um gestor do projeto AD em um conglomerado do direito, envolvendo advogados de segmentos diversos.

Já imaginou se a sua aplicação faz associação de um documento indevido, ou divergente, que um processo jurídico necessita? Como você previne, ou pelo menos diminui as chances desse tipo de erro acontecer, para evitar prejuízos em processos judiciais?

A participação dos gestores do escritório de advocacia está colaborando com o projeto da maneira necessária? Como seria possível aumentar o engajamento dos advogados nas decisões sobre requisitos funcionais do sistema AD?

Bem, esperamos que nesse cenário você possa compreender as possibilidades práticas dos temas abordados nessa unidade!

- As ferramentas em gestão são realmente úteis na gestão da qualidade?
- Por que a gestão da comunicação é fundamental na gestão de riscos?
- Qual é a importância da gestão de riscos em projeto de software?
- Qual é o benefício da documentação na qualidade do software?

Vejamos agora como seguir já que você que é o responsável pelo projeto AD. Conforme foi verificado, algumas situações necessitam de práticas de gestão para que o projeto tenha sucesso.

Construir a matriz de risco é um caminho adequado para você solicitar a ajuda de pessoas especialistas em documentos legais, leis, anexos e portarias que possam fazer parte de cada processo jurídico. Nessa prática, você conseguirá entender as causas e as respectivas consequências; sendo assim, junto dos especialistas, você vai prever na aplicação gatilhos que poderão evitar prejuízos em processos do escritório.

Nesse projeto, você perceberá que a mudança da cultura entre a equipe de TI e os advogados contribuirá imensamente, principalmente, durante o desenvolvimento do software. A comunicação sobre o andamento do projeto, mediante cronograma e WBS, entre outros artefatos, manterá os advogados envolvidos no projeto.

Objetivamente e em fases específicas, a sua atuação como gestor do projeto de software dominará o andamento das atividades e o gerenciamento da qualidade do produto final.

A Figura 1 ilustra os principais itens trabalhados na unidade, partindo da Qualidade em Projeto de Software, como tema principal, a qual depende de outros elementos: gestão de riscos, documentação do projeto, ferramentas em gestão e artefatos da comunicação.

Projeto de Software (u3)

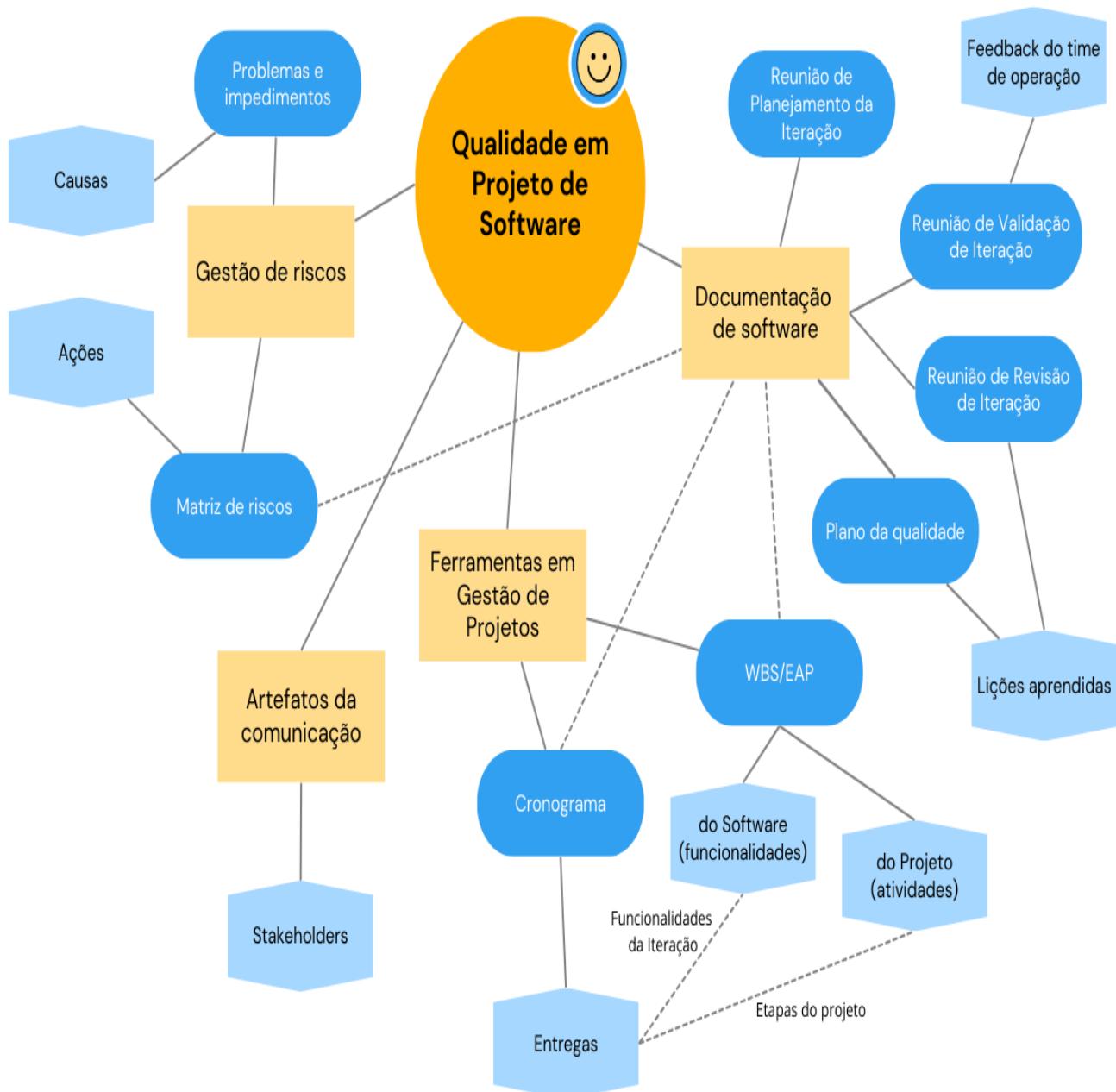


Figura 1 | Mapa mental. Fonte: elaborada pelo autor.

Os entrelaçamentos entre os itens são totalmente naturais. Ou seja, não fique restrito às linhas aqui apresentadas na Figura 1. Vejamos, *stakeholders* está diretamente associado a artefatos da

comunicação, mas este tem relacionamento em vários outros, tais como: cronograma, reuniões, gestão de riscos, etc.

Continue estudando sobre o desenvolvimento da qualidade dando foco em processos e em produtos. Afinal, entender a qualidade do produto te faz buscar novos processos ou estar preparado a experimentar novos processos te encorajará a desafios maiores em seu projeto de software.

Bons estudos!

ABNT. Gestão de Riscos – diretrizes. NBR ISO 31000:2018. Associação Brasileira de Normas Técnicas. 2018.

CAMARGO, R.; RIBAS, T. Gestão ágil de projetos. São Paulo: Saraiva Educação, 2019.

CAVALCANTI, F. R. P. Fundamentos de gestão de projetos: gestão de riscos. São Paulo: Atlas, 2016.

MAXIMIANO, A. C. A.; VERONEZE, F. Gestão de projetos: preditiva, ágil e estratégica. Barueri: Atlas, 2022.

PRESSMAN, R. S.; MAXIM, B. R. Engenharia de software. Porto Alegre: AMGH, 2021.

Unidade 4

Projeto de Software avançado

Aula 1

Gerenciamento de Configuração

Gerenciamento de Configuração

Este conteúdo é um vídeo!



Para assistir este conteúdo é necessário que você acesse o AVA pelo computador ou pelo aplicativo. Você pode baixar os vídeos direto no aplicativo para assistir mesmo sem conexão à internet.

Dica para você

Aproveite o acesso para baixar os slides do vídeo, isso pode deixar sua aprendizagem ainda mais completa.

PROJETO DE SOFTWARE

Estudante, esta videoaula foi preparada especialmente para você. Nela, você irá aprender conteúdos importantes para a sua formação profissional. Vamos assisti-la?
Bons estudos!

Ponto de Partida

Fala-se em gestão de configuração, também conhecimento como *Software Configuration Management* (SCM). Você sabe qual é verdadeira função do SCM?

Bem-vindos ao cenário de controle dos artefatos que compõem um produto funcionando. Ao consolidar um conjunto de itens de software estáveis, naturalmente teremos um software na versão instalável, junto ao time de operações.

Estudante, imagine-se num projeto de TI, ComponenteRastre, em que o desenvolvimento do software requer regras específicas para garantir que a arquitetura da aplicação seja rigorosamente seguida, é o SCM o mecanismo que permite tal rigor. Imagine que você está à frente de um cenário e necessita implementar técnicas para:

- Manter controle sobre as mudanças de código;
- Introduzir mecanismos para fazer solicitações oficiais de alterações;
- Avaliar as alterações propostas junto à equipe de desenvolvimento;
- Garantir que as mudanças sejam aceitáveis para o Product Owner;

Coletar dados estatísticos sobre os componentes do sistema de software, como informações determinando quais componentes do sistema são problemáticos.

A sua missão será experimentar os recursos do SCM para o bom gerenciamento de construção de código. Tenha consigo que compreender sobre o rastreamento das atividades da construção do código-fonte será a base para a prática nesse projeto de software.

Vamos Começar!

Inicialmente, é importante você perceber que o ambiente de desenvolvimento de software é cada vez mais exigente, por isso o construtor de código-fonte deve estar inserido em um processo específico e formal para verificar e avaliar cada modificação realizada.

Controle de versão

Quando você prepara o ambiente de desenvolvimento para implementar o SCM, deve saber identificar quatro importantes elementos, são eles:

Elementos de componente: conjunto de ferramentas acopladas em um sistema de gestão de arquivos (p. ex.: um banco de dados) que possibilita acesso à gestão de cada item de configuração de software;

- **Elementos de processo:** coleção de procedimentos e tarefas que definem uma abordagem eficaz de gestão de alterações (e atividades relacionadas) para todas as partes envolvidas na gestão, engenharia e uso do software;
- **Elementos de construção:** conjunto de ferramentas que automatizam a construção do software, assegurando que tenha sido montado o conjunto apropriado de componentes validados (p.e.: a versão correta);
- **Elementos humanos:** características de processo (abrangendo outros elementos de CM) usado pela equipe de software para implementar um SCM eficaz.

Um conceito apropriado para um ambiente controlado, segundo o IEEE (Instituto de engenheiros eletricistas e eletrônicos), se chama referência: uma especificação ou produto que tenha sido formalmente revisado e acordado, que depois serve de base para mais desenvolvimento e pode ser alterado somente por meio de procedimentos formais de controle de alteração.

Uma referência é um marco no desenvolvimento de software; caracterizada pelo fornecimento de um ou mais itens de configuração de software (SCIs) que foram aprovados em consequência de uma revisão técnica. Por exemplo: os elementos de um modelo de projeto foram documentados e revisados, ou erros foram encontrados e corrigidos. Uma vez que todas as partes do modelo foram revisadas, corrigidas e, então, aprovadas, o modelo do projeto torna-se uma referência, como ilustrado na Figura 1.

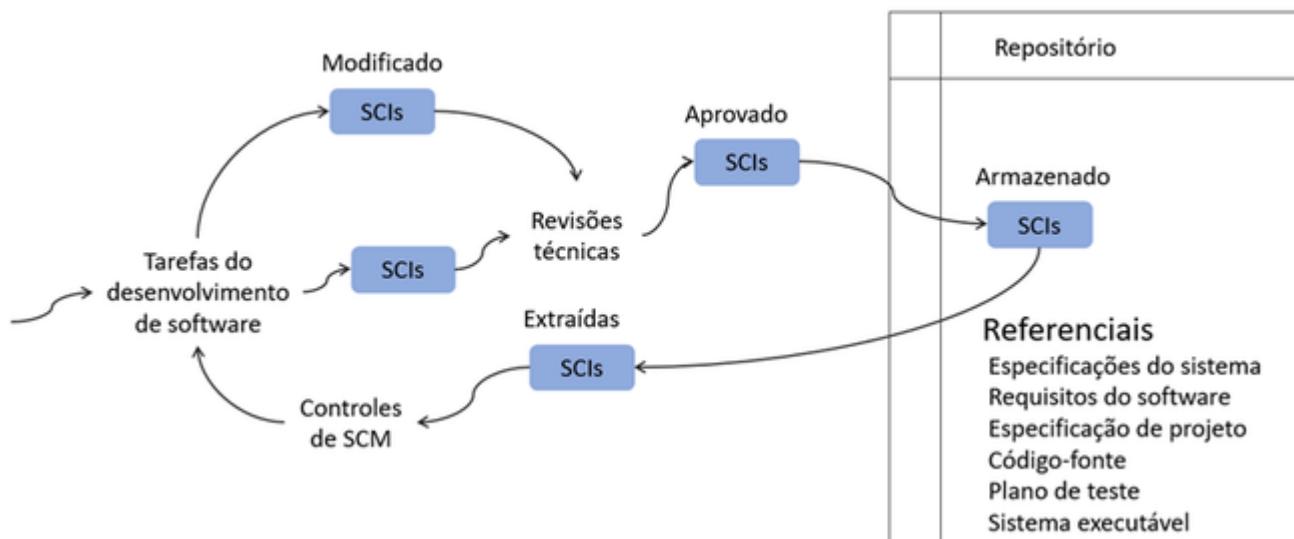


Figura 1 | SCIs referenciais (baseline) e o repositório. Fonte: adaptado de Pressman (2021, p. 441).

Estudante, é possível identificar os principais eventos processados que permitem a gestão da configuração dos SCIs pertencentes a uma determinada versão da aplicação e que, por sua vez, têm referência a uma determinada baseline.

Rastreamento de mudanças e configuração

Entende-se por rastreamento todos os SCIs que estão no projeto, utilizados em algum momento, ou aqueles que estão por vir no planejamento para se integrar e estão devidamente interrelacionados, assim pode haver astreabilidade entre eles.

São exemplos de artefatos: as histórias de usuários (cujo item foi identificado e está no Product Backlog), pelas quais se modelou dados, consequentemente se crio uma definição de banco de dados; que, por outro lado, a partir do objeto criaram-se códigos-fonte, planos de testes e seus componentes. Todos esses elementos podem ser rastreados se existirem as devidas referências em um repositório de SCM.

Seguindo as principais políticas verificadas, em gestão de configuração, temos algumas operações nas quais os engenheiros de software almejam trabalhar eficazmente. Conforme Pressman (2021, p. 440) são elas:

- Um mecanismo para assegurar que alterações simultâneas ao mesmo componente sejam adequadamente rastreadas, gerenciadas e executadas;
- Os engenheiros não interferem mutuamente de forma desnecessária na criação e no teste do código e na produção de artefatos de suporte;
- Ao mesmo tempo, eles tentam se comunicar e coordenar eficientemente;
- Usam ferramentas que ajudam a criar artefatos consistentes;
- Se comunicam e se coordenam, notificando reciprocamente sobre as tarefas necessárias e as tarefas concluídas;
- As alterações são propagadas por meio do trabalho dos outros mesclando arquivos;
- Existem mecanismos que asseguram que, para componentes submetidos a alterações simultâneas;
- Há uma maneira de resolver conflitos e mesclar alterações;
- É mantido um histórico da evolução de todos os componentes do sistema, num registro (log) com os motivos das alterações e um registro do que realmente foi alterado;

Os engenheiros têm seu próprio espaço de trabalho para criar, alterar, testar e integrar o código;

Em certo ponto, o código é transformado em referencial, com base em qual o desenvolvimento que continua e por meio do qual são criadas variações para as máquinas de destino.

Estudante, você percebe que esses mecanismos são essenciais para obter certa segurança de controle sobre os itens de softwares, todos que fazem, ou farão, parte da aplicação que está sendo ou será útil ao time de operações. Assim, todas essas operações realizadas pelo engenheiro de software ou desenvolvedor podem ser úteis se identificadas pela rastreabilidade.

Siga em Frente...

Processos de integração contínua

O avanço em tecnologia e metodologia no processo de desenvolvimento de software tende, cada vez mais, ao uso da construção de código-fonte integrado, a integração contínua (CI - *continuous integration*), importante para os desenvolvedores ágeis que seguem o fluxo de trabalho do DevOps.

Sendo assim, o CI agrega valor ao SCM, fazendo valer as melhores práticas: (1) minimizar o número de variantes de código; (2) testar desde o início e com frequência; (3) integrar desde o início e com frequência; e (4) usar ferramentas para automatizar o teste, a construção e a integração do código.

Porquanto, o SCM garante que todas as alterações sejam imediatamente integradas ao código-fonte do projeto, compiladas e testadas automaticamente. A CI oferece às equipes de desenvolvimento vantagens evidentes:

- **Feedback acelerado:** notificar os desenvolvedores imediatamente quando acontecem problemas na integração, permitindo que os consertos ocorram enquanto o número de alterações realizadas é pequeno.
- **Maior qualidade:** criar e integrar software sempre que necessário gerar confiança sobre a qualidade do produto desenvolvido.
- **Menor risco:** integrar os componentes com antecedência evita o risco de uma fase de integração longa, pois as falhas de projeto são descobertas e consertadas antes.
- **Melhores relatórios:** fornecer informações adicionais (p. ex., métricas de análise de código) permite uma contabilidade de *status* de configuração mais precisa.

Assim, os processos de CI colaboram imensamente, permitindo que gerentes de projeto, gerentes de garantia da qualidade e engenheiros de software melhorem a qualidade do software, reduzindo a probabilidade dos defeitos.

Estudante, você já percebe que para os gestores de projeto de software é uma conquista conseguir capturar defeitos nas fases iniciais, pois isso diminuirá consideravelmente os custos de desenvolvimento.

Vamos Exercitar?

Estudante, agora que você está preparado para implementar a solução em SCM no projeto ComponenteRastre, sabe que o seu time confiará na ferramenta de construção de código.

Você consegue perceber o uso de políticas e operações que afetam um SCI no projeto podem ser contempladas na gestão de configuração, tais como:

- Para manter o controle sobre as mudanças de código, utiliza-se de mecanismo para assegurar que alterações simultâneas ao mesmo componente sejam adequadamente

rastreadas, gerenciadas e executadas.

- As solicitações oficiais de alterações serão controladas por mecanismos de comunicação e coordenação das atividades.
- Será possível avaliar as alterações propostas junto à equipe de desenvolvimento, por intermédio de mecanismos que asseguram que, para componentes submetidos a alterações simultâneas, há uma maneira de resolver conflitos e mesclar alterações.
- Garantir que as mudanças sejam aceitáveis para o Product Owner, em função da comunicação e coordenação com maturidade, notificando reciprocamente sobre as tarefas necessárias e as tarefas concluídas; e que as alterações são propagadas por meio do trabalho dos outros mesclando arquivos.
- Coletar dados estatísticos sobre os componentes do sistema de software, garantido pela manutenção do histórico da evolução de todos os componentes do sistema um registro (log) com os motivos das alterações e um registro do que realmente foi alterado.

A rastreabilidade e a política do controle oferecidos pela gestão de configuração (SCM) torna o processo de desenvolvimento de software um ambiente seguro e gerenciado, como você pode perceber.

Bons estudos!

Saiba mais

A rastreabilidade é atributo essencial em revisões formais

Leia o item “Revisões formais”, Inspeção, página 85, sobre a verificação por conta da inspeção, em que é satisfatório o atributo da rastreabilidade.

FILHO, W. P. P. [Engenharia de Software - Projetos e Processos](#), v. 2, 4 ed. Rio de Janeiro: LTC, 2019.

1. Leitura básica do GIT

Leia artigo sobre o uso do Git, de Ariane G. [GIT – Tutorial Para Iniciantes. Hostinger](#), 2023.

2. Demonstração do GIT

[Confira uma demonstração do Git](#) e conheça as opções do gerenciamento de configurações GIT por meio de uma demonstração das principais funcionalidades da ferramenta.

3. Matriz de rastreabilidade

Leia o item 7.2.6 Rastreabilidade, página 109, que descreve a matriz de rastreabilidade como um recurso fundamental na identificação de um SCI.

PROJETO DE SOFTWARE

PRESSMAN, R. S.; MAXIM, B. R. [Engenharia de software](#). Porto Alegre: AMGH, 2021.

Referências

CAMARGO, R.; RIBAS, T. **Gestão ágil de projetos**. São Paulo: Saraiva Educação, 2019.

CAVALCANTI, F. R. P. **Fundamentos de gestão de projetos**: gestão de riscos. São Paulo: Atlas, 2016.

MAXIMIANO, A. C. A.; VERONEZE, F. **Gestão de projetos**: preditiva, ágil e estratégica. Barueri: Atlas, 2022.

PRESSMAN, R. S.; MAXIM, B. R. [Engenharia de software](#). Porto Alegre: AMGH, 2021.

Aula 2

Evolução e Mudanças de Software

Evolução e Mudanças de Software

Este conteúdo é um vídeo!



Para assistir este conteúdo é necessário que você acesse o AVA pelo computador ou pelo aplicativo. Você pode baixar os vídeos direto no aplicativo para assistir mesmo sem conexão à internet.

Dica para você

Aproveite o acesso para baixar os slides do vídeo, isso pode deixar sua aprendizagem ainda mais completa.

Estudante, esta videoaula foi preparada especialmente para você. Nela, você irá aprender conteúdos importantes para a sua formação profissional. Vamos assisti-la? Bons estudos!

Ponto de Partida

Voltemos o olhar à agilidade e aos muitos detalhes em desenvolvimento de software, tanto por parte da entrega ao usuário, quanto em procedimentos e ferramentas no processo todo.

PROJETO DE SOFTWARE

Estudante, se coloque no lugar de um dirigente que toma decisões importantes semanalmente. A velocidade das mudanças aumenta significativamente na fábrica CosmeTech, *startup* no ramo de cosméticos, a cada temporada. Se antes uma regra de negócio ficava intacta por vários meses, atualmente essas regras se alteraram muitas vezes em 12 meses; sem contar ainda que algumas que até desaparecem. Isso se dá porque a sociedade, os empreendimentos e os indivíduos exigem um software com o comportamento inovador e atualizado.

Diante de um cenário altamente dinâmico, estratégico e ao mesmo tempo muito imediato, tudo se pensa estrategicamente, ao mesmo devem ser praticadas no curto prazo. Portanto, as metas táticas e decisões de mudanças, de processos e de resultados, devem ocorrer na semana ou até diariamente. Sendo assim, o aprendizado e habilidade de execução, na CosmeTech, para executar novas tarefas devem ser absorvidas e praticadas quase instantaneamente.

Você é o dirigente do projeto de software na CosmeTech, deve estar pronto para selecionar processos que promovem a dinâmica segura, implementar novos labores que proporcionarão resultados vertiginosos para a indústria. Conheceremos técnicas da gestão da mudança e a observabilidade, sob o método *Continuous Integration/Continuous Deployment* (CI/CD).

Vamos Começar!

Boas-vindas ao cenário de tecnologias emergentes, uso de ferramentas configuráveis e adaptáveis ao panorama de mudanças frequentes, nos negócios e na Tecnologia da Informação (TI). Se você planeja ser um desenvolvedor inovador e eficiente, se prepare para a constante necessidade de recriar soluções em software.

Manutenção e evolução do software

Primeiramente, você deve aprender a fazer uma leitura adequada do mercado e das organizações concorrentes. Se o escopo (requisitos, histórias de usuário) mudam com muita frequência, se o prazo de desenvolvimento está abaixo do que de costume e se o orçamento do projeto é muito baixo exigem-se expertises apuradas, disciplina e rigor na implementação de processos otimizados.

Esta realidade é comum em quase todos os setores da economia, ou seja, a evolução e as modificações dos procedimentos operacionais na indústria, no comércio, no serviço, ou nas instituições do conhecimento ocorre com muita frequência. A mesma evolução que acontece no ambiente operacional também deve ocorrer nas aplicações de software, de maneira similar em velocidade e precisão.

É importante você perceber que o ambiente de desenvolvimento de software necessita dar respostas rápidas às modificações das regras de negócio, que a manutenção do software deve ocorrer concomitantemente à evolução do software; o construtor de código-fonte deve estar inserido num processo contínuo e formal para uso em curtíssimo prazo.

Gestão de mudanças e observabilidade

Ao mesmo tempo que as mudanças acontecem no ambiente operacional, o time de TI deverá “correr” para implementar e adaptar a versão atual, para a nova realidade na próxima iteração do desenvolvimento.

A revisão constante dos recursos e das necessidades é prática fundamental para o sucesso da gestão de mudanças. Lembre-se que as restrições de recurso, escopo, tempo e qualidade é imprescindível em novos projetos! Da mesma forma, o gestor do projeto deve ter o mesmo rigor a cada iteração, da manutenção/evolução, para garantir que as mudanças sejam empreendidas com sucesso, como ilustrado na Figura 1.



Figura 1 | Restrições do sucesso em projetos. Fonte: adaptado de Camargo (2019, p. 100).

As práticas da gestão de mudanças aparecem claramente na fase de controle do projeto, em que acompanhar, rever e atuar no progresso e desempenho do projeto, conforme Cavalcanti (2016) garantirá que o sistema estará aderente aos negócios, em que se deve:

- Gerenciar as tarefas do projeto, considerando pessoas envolvidas e os demais recursos computacionais e organizacionais.
- Controlar as mudanças do projeto, de requisitos funcionais e não funcionais, e demais elementos da TI.
- Providenciar, antecipadamente, ações para evitar problemas ou minimizar prejuízos.

PROJETO DE SOFTWARE

- Garantir a implementação apenas as mudanças aprovadas, cientes entre dono do projeto, time de operações e time de TI.

Ações que vão ampliar, tecnicamente, na prática, a prevenção da propagação de possíveis problemas e consequências negativas a curto e longo prazo são iniciativas da observabilidade.

Estudante, saiba que a observabilidade é empregada em ambiente operacional complexo, por meio da detecção de anomalias. São definidas como comportamentos ou eventos que são bem diferentes do que é normalmente esperado ou planejado. A detecção de anomalias envolve a identificação de padrões de comportamento anormais, ou eventos que possam indicar problemas ou falhas no sistema (REINHOLDS, 2021), vejamos algumas práticas-chave:

- a) Coletar dados relevantes:** é importante fazer coleta de dados relevantes para detecção de anomalias; pode incluir dados de desempenho, logs, métricas e outros dados relevantes para o comportamento do sistema (SRIDHARAN, 2018).
- b) Definir limites e padrões de comportamento normais:** definir limites e padrões de comportamento normais para cada componente do sistema; os parâmetros definidos são usados como base para identificar anomalias e alertar as equipes quando os padrões normais são ultrapassados (SRIDHARAN, 2018).
- c) Utilizar algoritmos de detecção de anomalias:** utilizar algoritmos de detecção de anomalias para identificar anomalias automaticamente; podem ser usados para analisar dados de desempenho e outros dados relevantes para identificar padrões de comportamento anormais (SRIDHARAN, 2018).
- d) Desenvolver uma mentalidade voltada para a detecção de anomalias:** as equipes devem estar atentas a eventos e comportamentos inesperados, investigar suas causas e tomar medidas corretivas quando necessário. Ter essa mentalidade, combinada com o uso de ferramentas avançadas, como o Amazon QuickSight, pode ajudar a garantir a estabilidade e a eficiência dos sistemas em ambientes operacionais complexos (AMAZON WEB SERVICES, 2023).

Siga em Frente...

Continuous Integration/Continuous Deployment

Relembremos que uma das metas do projeto de software é entregar a aplicação funcionando corretamente com as funcionalidades aprovadas em planejamento da iteração. Alcançar esse sucesso depende dos códigos-fonte. A integração contínua, para Pressman (2021) é a prática de fundir componentes com o incremento de software em evolução uma ou mais vezes ao dia. Essa prática é facilmente verificada em Extreme Programming (XP), ou DevOps.

Diante da competitividade no mercado de serviços, produtos e inovações, a comunidade de engenharia de software desenvolve novas abordagens para a representação de requisitos de

PROJETO DE SOFTWARE

sistema e de projeto, na qual as seguintes características podem ser tratadas pelo CI/CD:

- **Multifuncionalidade:** os dispositivos digitais evoluíram, começaram a apresentar um amplo conjunto de funções, às vezes não relacionadas no início do projeto.
- **Reatividade e temporização (*timeliness*):** os dispositivos digitais interagem com o mundo real e devem reagir aos estímulos externos de maneira pontual. Eles devem oferecer interface com um amplo conjunto de sensores e devem responder em um intervalo de tempo apropriado para a tarefa em questão.
- **Novos modos de interação do usuário:** tendências abertas para software significam que novos modos de interação devem ser modelados e implementados; podendo ser interfaces multitoque, reconhecimento de voz, ou interfaces mentais diretas.
- **Arquiteturas complexas:** sistemas mais complexos estão no horizonte próximo, apresentando desafios significativos para os projetistas de software.
- **Sistemas heterogêneos distribuídos:** os componentes de tempo real de qualquer sistema embarcado moderno podem ser conectados por meio de um barramento interno, de uma rede sem fio, ou da Internet (ou pelas três coisas).
- **Criticidade:** o software tornou-se o componente central em todos os sistemas críticos nos negócios. Contudo, a comunidade de engenharia de software apenas começou a aplicar os princípios mais básicos de segurança de software.
- **Variabilidade de manutenção:** a vida útil do software é variável, conforme sua natureza, dispositivo digital, softwares de automóveis, ou sistemas da aviação.

Diante das inúmeras realidades da TI, um ambiente inteiramente integrador se faz necessário. Juntamente com iterações ágeis e conceitos do “pronto”, as organizações e seus times de operações necessitam de aplicações mais aderentes às realidades, em tempo real.

Portanto, o CI/CD, é a solução nesse cenário tão exigente pela complexidade e manutenção. Você precisa reconhecer as situações observadas em seu ambiente e auxiliar na configuração de ferramentas que entregarão sistemas em funcionamento com o menor tempo possível. Estudante, continue atualizado em CI/CD para o seu sucesso em projeto de software.

Vamos Exercitar?

Estudante, vamos te mostrar como a CosmeTech procederá para controlar as mudanças com o máximo de rigor e entrega cada vez mais segura e rápida.

Você pode iniciar a solução por meio da CI/CD, preparando todo ambiente de desenvolvimento em integração contínua e configurar consequentemente o ambiente operacional usando a implantação contínua.

A solução para a gestão de mudanças será incrementada com a aplicação da observabilidade, configurada para indicar o comportamento do software ao longo processo de desenvolvimento, principalmente quanto a resultados de defeitos encontrados, para cada requisito construído.

PROJETO DE SOFTWARE

Assim, o resultado da observabilidade ficará sob a responsabilidade do gestor de TI decidir por modificar o processo de desenvolvimento, aplicar melhorias em técnicas ou ferramentas.

Da mesma forma, a CosmeTech terá acesso aos registros e indicadores comparativos entre um produto e outro, mantendo um histórico seguro; sendo analisável para tomada de decisões; tanto para otimizar a linha de produção, ou para interromper o produto que se apresenta aquém das expectativas.

Estudante, com a modificação do processo de desenvolvimento, incrementando a observabilidade, muitos indicadores farão parte do histórico para o time de TI com dados técnicos; e à disposição do time de operações com dados de negócios. Tudo isso será insumo para melhorar a gestão de mudanças da organização e, também, de projeto de software.

Bons estudos!

Saiba mais

1. Conceitos para detecção de anomalias

Leia o guia “Conceitos para detecção de anomalias”, conforme a seguir:

AMAZON WEB SERVICES (AWS). (s.d.). Detecção de anomalias, outliers e principais drivers. [Documentação do Amazon QuickSight](#).

2. Processos de escopo nas fases de execução e controle

Leia “Processos de escopo nas fases de execução e controle”, página 97, sobre tarefas na fase de execução e controle do projeto.

CAVALCANTI, F. R. P. [Fundamentos de gestão de projetos: gestão de riscos](#). São Paulo: Atlas, 2016. [Minha Biblioteca]

3. Uma estratégia de integração contínua

Leia o item 20.2.3 Integração contínua, página 400, e entenda sobre a integração contínua na prática de testes.

PRESSMAN, R. S.; MAXIM, B. R. [Engenharia de software](#). Porto Alegre: AMGH, 2021. [Minha Biblioteca]

Referências

PROJETO DE SOFTWARE

AMAZON WEB SERVICES (AWS). (s.d.). Detecção de anomalias, outliers e principais drivers.

Documentação do Amazon QuickSight. Disponível em:

https://docs.aws.amazon.com/pt_br/quicksight/latest/user/anomaly-detection-outliers-and-key-drivers.html. Acesso em: 11 nov. 2023.

CAVALCANTI, F. R. P. **Fundamentos de gestão de projetos**: gestão de riscos. São Paulo: Atlas, 2016.

PRESSMAN, R. S.; MAXIM, B. R. **Engenharia de software**. Porto Alegre: AMGH, 2021.

REINHOLDS, A. (2021). What is observability? **Newrelic**. Disponível em:

<https://newrelic.com/blog/best-practices/what-is-observability/>. Acesso em: 11 nov. 2023.

SRIDHARAN, C. (2018). **Monitoring and Observability**. Disponível em:

<https://www.oreilly.com/library/view/distributed-systems-observability/9781492033431/ch01.html>. Acesso em: 10 nov. 2023.

Aula 3

Projeto da experiência do usuário

Projeto da experiência do usuário

Este conteúdo é um vídeo!



Para assistir este conteúdo é necessário que você acesse o AVA pelo computador ou pelo aplicativo. Você pode baixar os vídeos direto no aplicativo para assistir mesmo sem conexão à internet.

Dica para você

Aproveite o acesso para baixar os slides do vídeo, isso pode deixar sua aprendizagem ainda mais completa.

Estudante, esta videoaula foi preparada especialmente para você. Nela, você irá aprender conteúdos importantes para a sua formação profissional. Vamos assisti-la?
Bons estudos!

Ponto de Partida

PROJETO DE SOFTWARE

Boas-vindas ao tema que agrada a todos os usuários de aplicativos! Para esse assunto, te colocaremos na PersonaDesign, um grupo de profissionais altamente capacitado para manter a saúde das pessoas em dia. Você será o responsável pelo diálogo entre o time de TI e as pessoas que utilizarão os aplicativos, para acompanhamento das atividades físicas com os clientes e para o monitoramento cardiovascular deles.

As principais funcionalidades que necessitam de uma interface com profissionais da PersonaDesign são: cadastro e perfis dos clientes; plataforma de acompanhamento e evolução da saúde e agendamento das sessões. Por outro lado, os clientes interagem com o sistema para receber as orientações na plataforma de acompanhamento e a evolução da sua saúde, interface de quitação das mensalidades e aplicativo móvel para o monitoramento em tempo integral, em casos especiais.

Primeiramente, se é necessário um projeto de interface com o usuário, quais elementos são fundamentais nesse projeto? Tecnicamente, veremos quais são os elementos associados à experiência do usuário (UX). Para garantir o funcionamento adequado, explanaremos como será avaliada a interface construída.

Estudante, faça uma reflexão em rotinas do seu dia a dia que você possa comparar com as situações apresentadas, isso facilita a compreensão de cada conceito abordado nesse tema.

Bons estudos!

Vamos Começar!

Veremos uma sequência lógica dos processos relacionados ao design, de como você executará estratégias para obter a melhor experiência do usuário no desenvolvimento de aplicações.

Projeto de interação de usuário

Bem, a interação do software com o usuário é construída a partir da vivência e experiência do usuário. Portanto, estar junto à rotina dele auxiliará na produção de protótipos ricos em detalhes, semelhantes ao que usuários de todos os níveis desejam e, também, como eles aproveitarão o potencial de um recurso da TI. Você deve mergulhar em situações reais, praticar a empatia com os profissionais, especialistas, a sociedade, ou qualquer classe de envolvido com o sistema em desenvolvimento.

Diante da lista das histórias dos usuários, dentre as prioritárias, classificadas durante a fase de elicitação de requisitos, agora, deve prosseguir interagindo continuamente com os envolvidos para a construção do protótipo, visando produzir um artefato muito próximo do aplicativo real.

Os passos fundamentais do projeto de interação do usuário, como sugerido por Pressman (2021) e ilustrado na Figura 1 são:

- **Rascunhe a interface do usuário:** pense na funcionalidade (e rabisque), na visão do envolvido que realmente utilizará o software.
- **Crie o protótipo virtual:** materialize em formato digital com alguns elementos de interação (visual, toque, visão, audição, entre outros).
- **Adicione entrada e saída em código:** construa determinado comportamento, considerando a funcionalidade da história prioritária.
- **Teste o protótipo com o usuário:** momento de aguçar a sensibilidade para coletar o máximo de *feedback*, aproveite a oportunidade para construir a lista de dados e comportamentos da prática do usuário.
- **Atualize o protótipo sempre:** adequações podem ocorrer por mudanças de requisitos, pela escolha da arquitetura de software, ou escolhas do recurso de TI.

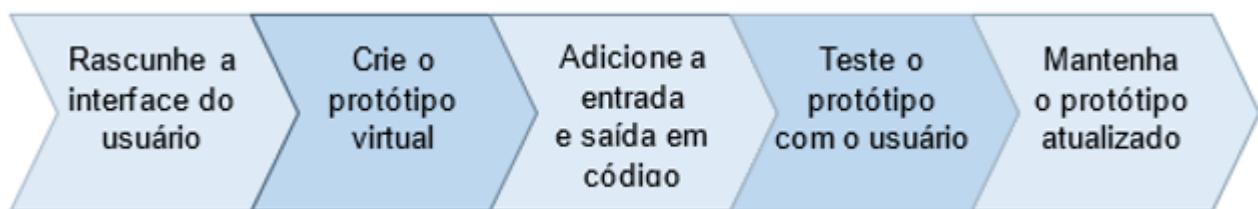


Figura 1 | Prototipação (projeto de interação do usuário). Fonte: elaborada pelo autor.

O projeto de interface, no projeto de software, é uma das etapas cruciais, pois essa interação que identificará visualmente e consolidará o comportamento do software.

Elementos do projeto da experiência do usuário

Para considerar os elementos da experiência do usuário (UX), no projeto de software, você deve ter a sensibilidade para escolher e trabalhar com elementos fundamentais, conforme cita Pressman (2021):

- **Envolvidos no projeto e no produto:** pessoas que dialogarão com o aplicativo em algum momento na operação de suas atividades.
- **Experiências:** pesquisar informações diversas, conforme o objetivo do projeto, ações, sentimentos, finalidade, ideias, motivações, situações críticas, barreiras, questionamentos, tudo que pode influenciar no comportamento da aplicação.
- **Da interface:** construção do protótipo visual, de contato, áudio, movimento e sensações.

Como não poderia ser diferente, o principal elemento no projeto de interação é a modelagem da persona. Como o usuário reagirá ao estímulo do sistema, ou qual será a reação do sistema quando o usuário interagir com o sistema.

Na modelagem do persona, ilustrada na Figura 2, são sintetizadas características pessoais, a partir dos dados coletados durante entrevistas com usuários, devendo representar os objetivos e comportamentos de um perfil de usuário.

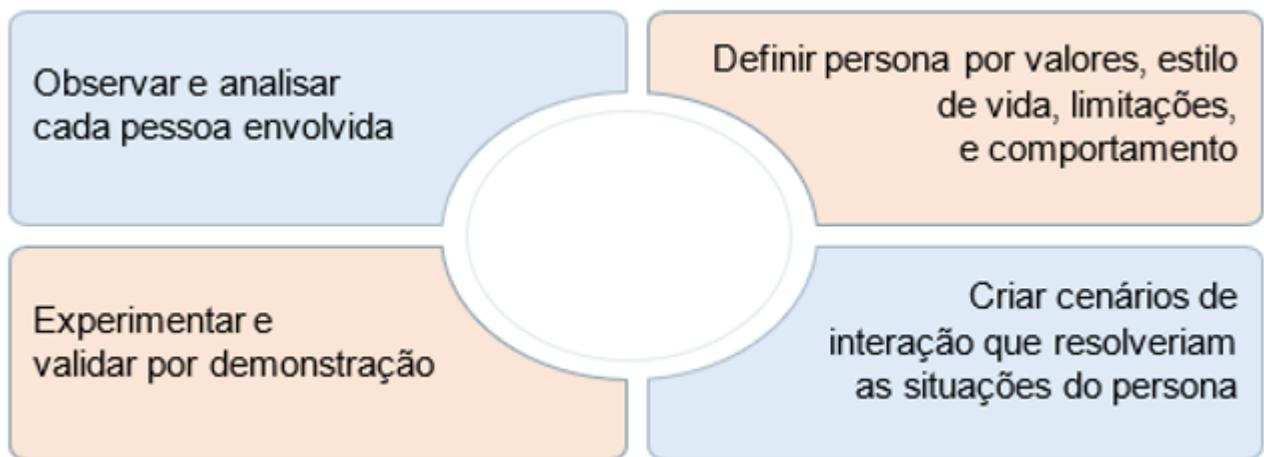


Figura 2 | Persona UX. Fonte: elaborada pelo autor.

Você será o principal responsável na projeção do persona (ou de várias) que influenciará no desenvolvimento e uso do sistema. Por isso, exerce o seu melhor na compreensão das necessidades, dos desejos e da satisfação do usuário.

Siga em Frente...

Análise e projeto de interfaces

Para concluir a fase de interação do usuário, vamos compreender a depuração do projeto de interfaces que deve se manter em constante melhoria, conforme o modelo do processo de desenvolvimento. O fluxo que se iniciou com o primeiro protótipo deve seguir por adaptações em todas (ou quase todas) as iterações do projeto de software, ilustrada na Figura 3.



Figura 3 | Avaliação do projeto de interfaces. Fonte: elaborada pelo autor.

Para analisar e validar a interface, consideremos os critérios relacionados à tecnologia, natureza do software, segmento do mercado da aplicação; porém, é relevante considerar aspectos específicos de UX, segundo Pressman (2021):

- O volume de aprendizado pelos usuários depende da duração e da complexidade de especificação de requisitos.
- O tempo de interação com o usuário varia conforme a quantidade de processos e o número médio de ações de cada processos.

- Os recursos computacionais são dimensionados conforme a quantidade de ações, processos e estados que o sistema demanda para realizar a funcionalidade.
- O protocolo de tratamento de erros e o estilo da interface são reflexos da complexidade e do grau de aceitação do usuário.

Como você pode perceber, exige-se atenção em detalhes e características para conceber e manter uma interface aceitável pelo usuário e os demais perfis de usuário. Lembre-se que todas as preocupações apresentadas são relevantes e desejáveis, em testes de sistemas, nas revisões da iteração e em *feedback* dos envolvidos.

Estudante, bons estudos! Continuamente, acesse informações acerca desse tema tão importante para o profissional de TI, em projetos de software.

Vamos Exercitar?

Percebemos como você poderia propor o projeto de interação e a interface no projeto PersonaDesign.

Quanto maior tempo você conseguir se dedicar na construção da interação (do protótipo de interface) com o usuário, maior será o aprendizado dele. Neste ponto é importante ser eficiente, portanto, escolher perfis específicos de usuários (envolvidos) é uma grande missão: qual perfil profissional *personal trainer* ou qual perfil cliente da PersonaDesign?

- a) Um processo dos clientes da PersonaDesign depende da acessibilidade, ou seja, considere o desenvolvimento de um aplicativo móvel para uma experiência mais conveniente e móvel, no monitoramento integral da pessoa.
- b) Experimente quais e como as notificações push serão úteis, para lembrar as próximas sessões, enviar dicas de saúde e manter o engajamento, ou alarmar por indicadores.

Todas essas características devem estar nos protótipos e ser validadas, antes do início da codificação da aplicação.

Ainda, você pode trabalhar com experiências de outras organizações que vivenciaram situações adversas, como a detecção de anomalias ou valores atípicos, conforme consta no guia de usuários da AWS.

Bem, as novas iterações do projeto podem trazer outras situações mais desafiadoras. Imagine que o *personal trainer* está em pleno atendimento e explanando as situações, os resultados (até o momento) da evolução do cliente, no próprio celular do cliente. De repente, uma notificação de chamada telefônica se exibe na tela.

Como se pode resolver essa situação? Pode não ser problema, mas obtenha do usuário como ele gostaria de configurar essas notificações para a sua aplicação.

PROJETO DE SOFTWARE

Bons estudos!

Saiba mais

1. Arquiteturas que favorecem a interface de usuários

Leia o tópico, na página 189 a 191, a Arquitetura Modelo-Visão-Controlador, no qual a visão contém todas as funções específicas à interface e possibilita a apresentação do conteúdo e lógica de processamento exigido pelo usuário.

PRESSMAN, R. S.; MAXIM, B. R. [Engenharia de software](#). Porto Alegre: AMGH, 2021.

2. Experiência de organização global

Leia o que a AWS compartilhou sobre Interfaces, [Conceitos para detecção de anomalias ou valores atípicos](#).

3. Exemplos de testes de interação do usuário

Leia o item 21.3.4, Alertas e condições extraordinárias, na página 417, em que você vai conhecer algumas situações de interação que podem ser tratadas na concepção do sistema.

PRESSMAN, R. S.; MAXIM, B. R. [Engenharia de software](#). Porto Alegre: AMGH, 2021. [Minha Biblioteca]

4. UX – Métodos para colher insights do seu público-alvo

Leia a publicação no site [“Muito além do teste de usabilidade: os vários tipos de pesquisas com usuários em UX”](#), e vai compreender como uma pesquisa e a validação da interface é detalhada.

Referências

AWS – Amazon. **Conceitos para detecção de anomalias ou valores atípicos**. Disponível em: https://docs.aws.amazon.com/pt_br/quicksight/latest/user/anomaly-detection-outliers-and-key-drivers.html. Acesso em: 15 nov. 2023.

CAMARGO, R.; RIBAS, T. **Gestão ágil de projetos**. São Paulo: Saraiva Educação, 2019.

CAVALCANTI, F. R. P. **Fundamentos de gestão de projetos: gestão de riscos**. São Paulo: Atlas, 2016.

PROJETO DE SOFTWARE

MAXIMIANO, A. C. A.; VERONEZE, F. **Gestão de projetos**: preditiva, ágil e estratégica. Barueri: Atlas, 2022.

PRESSMAN, R. S.; MAXIM, B. R. **Engenharia de software**. Porto Alegre: AMGH, 2021.

Aula 4

Paradigmas em Projeto de Software estratégico

Paradigmas em Projeto de Software estratégico

Este conteúdo é um vídeo!



Para assistir este conteúdo é necessário que você acesse o AVA pelo computador ou pelo aplicativo. Você pode baixar os vídeos direto no aplicativo para assistir mesmo sem conexão à internet.

Dica para você

Aproveite o acesso para baixar os slides do vídeo, isso pode deixar sua aprendizagem ainda mais completa.

Estudante, esta videoaula foi preparada especialmente para você. Nela, você irá aprender conteúdos importantes para a sua formação profissional. Vamos assisti-la?
Bons estudos!

Ponto de Partida

Agora estamos no tema extremo em inovação e competitividade, sejam bem-vindos à governança em projeto de software, reunindo do tradicional e atual até o desafiador paradigma ESG.

Fundada há mais um século por um chocolatier, a ChocoRock, fábrica renomada de chocolate, deseja melhorar a transparência em sua cadeia de suprimentos e garantir práticas comerciais sustentáveis. A organização segue firme e engajada, com os princípios ESG e visa desenvolver uma plataforma digital de rastreabilidade para monitorar todo o ciclo de produção, desde a origem dos ingredientes até o produto acabado, inclusive a sua distribuição.

O time de desenvolvimento de software tem o domínio e prática da metodologia ágil, portanto está seguro em aprender novas necessidades, técnicas e frameworks.

PROJETO DE SOFTWARE

Proponho partirmos para extrair o que é essencial em projeto de software com ITIL, COBIT, PMI e PMBOK, para que a ChocoRock aproveite as melhores oportunidades em construção de software. Simultaneamente, a nova plataforma deverá ser desenvolvida aderente ao ESG (*Environmental, Social and Governance*) e às diretrizes da Governança de TI.

Estudante, mergulhe no vasto conhecimento em gestão de projetos, identifique as oportunidades de uso em seu projeto e aproveite para desenvolver o seu melhor projeto de software. Além de reconhecer a ampliação das competências em gestão, você poderá demonstrar a melhoria da qualidade em processos de desenvolvimento de software.

Bons estudos!

Vamos Começar!

A Governança é o tema do momento! Se fala muito em todos os setores, inclusive é assunto obrigatório em TI. Compreendamos as diversidades das metodologias, esteja preparado como profissional do gerenciamento de projeto de software, ou em uma equipe de TI.

Projeto de Software com ITIL, COBIT, PMI e PMBoK

Relembremos sobre a gestão de recursos e das melhores práticas em gerenciamento da qualidade sendo conduzido por pessoas engajadas com o projeto de software. O triângulo de talentos do guia PMBoK/PMI, em que o foco deixou de ser apenas na entrega do que foi encomendado, mas atender ao que for melhor para o negócio, ou seja: gestão estratégica de negócios complementada com as técnicas em gestão de projetos e pela liderança.

As melhores práticas reunidas no PMBoK foram condensadas a partir de experiências vivenciadas por gestores de projetos em diversos negócios, mundo afora, portanto, é naturalmente compreensível adotar algumas delas ou parcialmente uma dessas práticas.

A nova gestão, um modelo ágil com ferramentas do preditivo, com base no PMBoK, segundo Maximiano (2022, p. 178) percebem-se vantagens das fases do preditivo e a dinâmica do ágil, uma combinação para otimizar as entregas e proporcionar benefícios aos stakeholders, como ilustrado na figura 1.

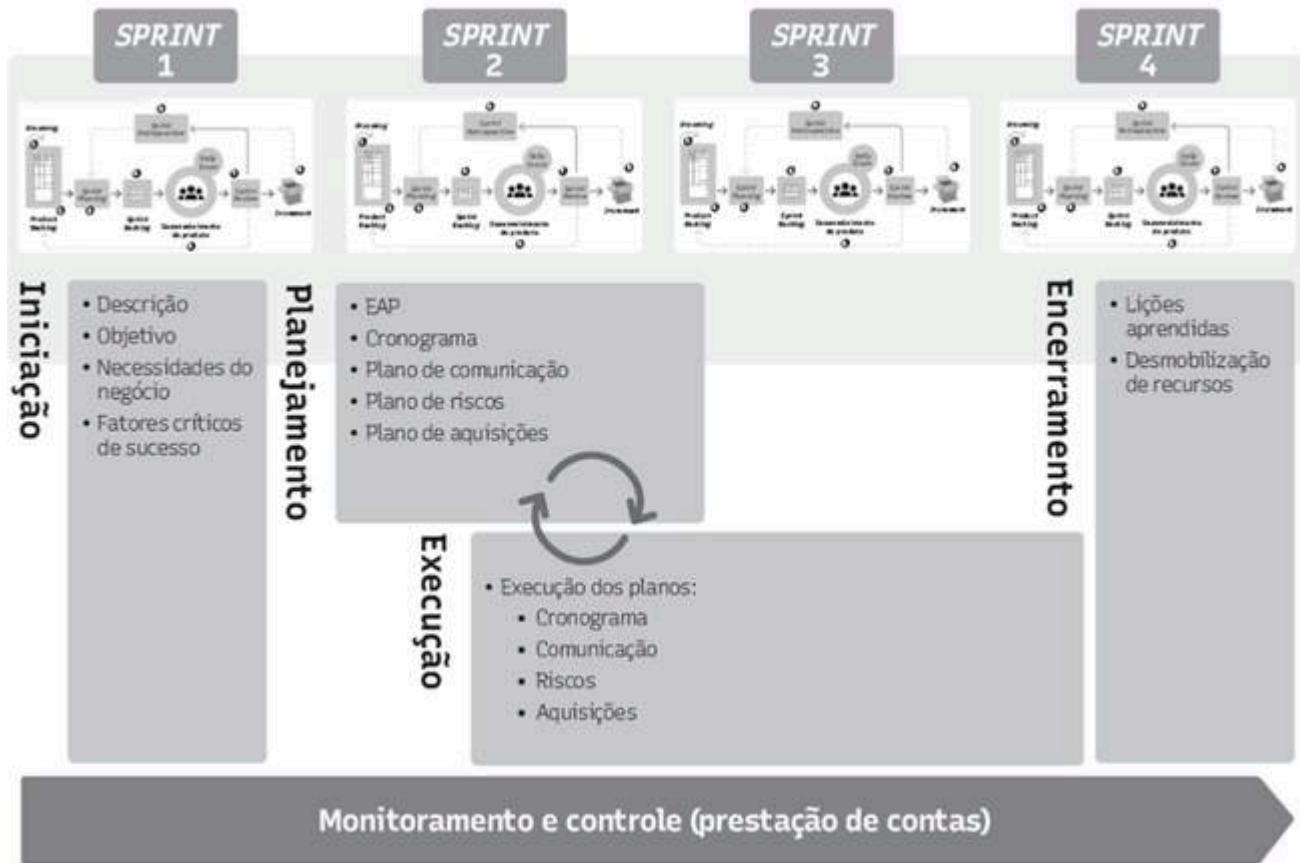
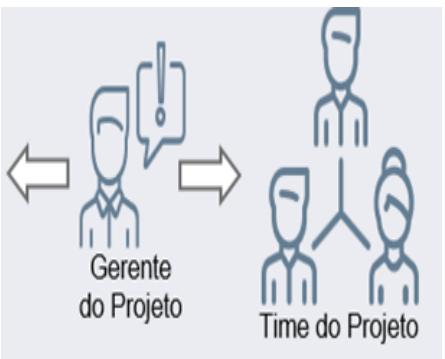
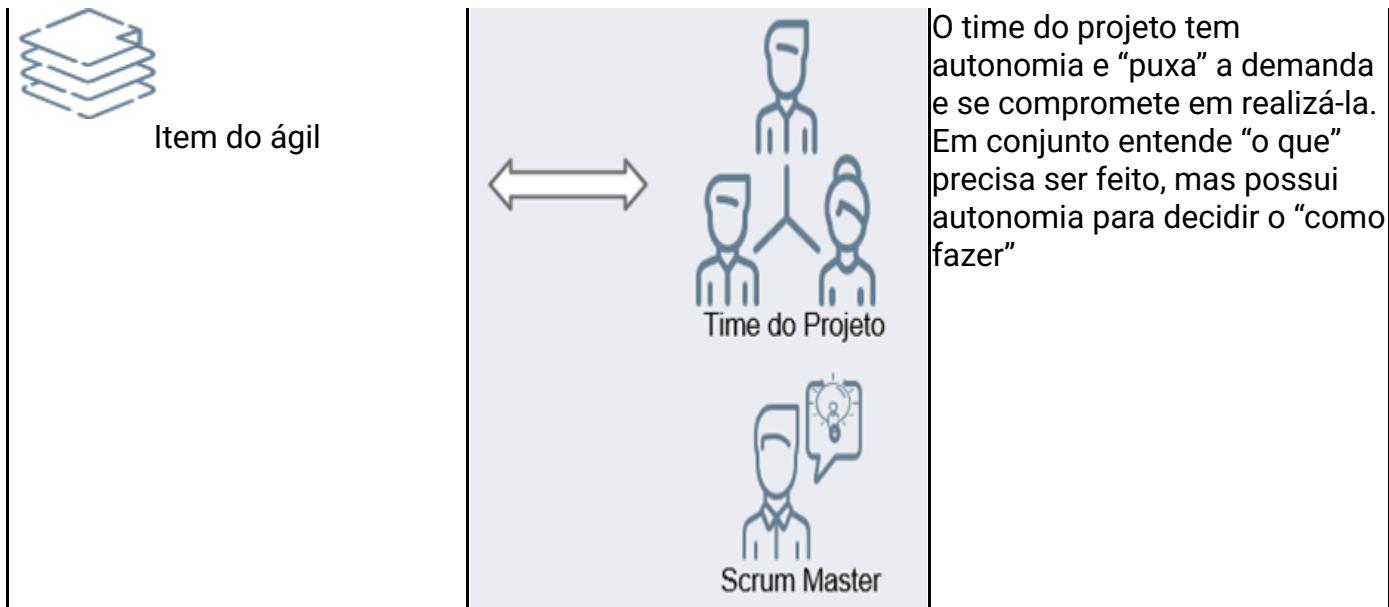


Figura 1 | Nova gestão, ágil com ferramentas do preditivo. Fonte: Maximiano (2022, p. 178).

Perceba que o modelo da nova gestão deve seguir o conceito timeboxing, base da metodologia ágil, como principal imposição do fluxo de atividades; nela acrescentar as ferramentas do processo preditivo. Um detalhe importante que se deve cuidar é quanto ao esclarecimento dos papéis do Gerente do Projeto e do Scrum Master. Vejamos, no Quadro 1, a diferença existente entre eles.

Backlog	Estrutura do projeto	Comportamento
 Item do preditivo		Processo Empurrado O gerente do projeto frequentemente “empurra” a demanda para a equipe, pressionando e direcionando o “o que” e “como fazer”
		Processo Puxado



Quadro 1 | Princípios ágil, papel colaborativo no lugar do comando e controle. Fonte: adaptado de Maximiano (2022, p.177).

Os frameworks cooperam com gestão de projetos, apresentam técnicas que darão maior segurança na administração dos recursos, de pessoas, do tempo, da qualidade e todos os aspectos que um projeto envolve.

O ITIL é largamente aplicável no gerenciamento da tecnologia por apresentar uma biblioteca e modelos apropriados para atender às demandas de ampliação ou diagnósticos em recursos da TI. Aproveite por apresentar uma estrutura prática e flexível como suporte na transformação digital, alinhando os recursos humanos, digitais e físicos.

Já o COBIT é um framework mantido pela ISACA, uma associação global de profissionais de auditoria e controle em sistemas de informação. Portanto, é um manual com reconhecimento internacional e formado pela união de recomendações e boas práticas de governança de TI.

Paradigmas ESG - Environmental, Social and Governance

Comecemos relembrando do conceito de riscos. Se utilizamos modelos de ciclo de vida, mesmo que adaptados à nossa realidade e experiência, ou seguimos recomendações de especialistas em qualidade é porque escolhemos caminho menos arriscado. Ah! Não esqueçamos do grau de dificuldade, em função da inovação desafiadora que nos move para alcançar um determinado sucesso, em projeto de software.

Muito bem! As medidas de governança e ESG recomendadas permitem anteciparmos os riscos, para aumentar as chances de evitá-los, também para mitigá-los, até mesmo para preparar possíveis respostas às crises.

As métricas em ESG são diretamente relacionadas a investidores que defendem equilibrar os interesses financeiros de curto prazo com preocupações mais amplas relacionadas ao meio ambiente, à sociedade e à ética nos negócios, conforme Filho (2022, p. 26).

Sendo o principal foco fortalecer a reputação da empresa, para manter a confiança do público e evitar controvérsias e boicotes, as prioridades estão baseadas em três pilares: *Environmental, Social e Governance*.

- **Environmental:** ações relacionadas aos recursos naturais do planeta, como: emissão de carbono, aquecimento global, eficiência energética, gestão de resíduos, poluição do ar e da água são bem-vindas quando a organização gerencia essas iniciativas.
- **Social:** fator humano, prezando pelo bem mental e físico dos colaboradores. A relação com inclusão e diversidade, e o relacionamento com as demais entidades da sociedade.
- **Governance:** considerando política de remuneração dos altos cargos, transparências e ética da instituição; responsabilidade pela proteção de dados e privacidade das informações dos colaboradores.

O fator investimento nas organizações tem impacto direto para os negócios, mas qual seria a relação com a TI? Compreendemos, segundo DTT, sob o aspecto da inovação, quando a TI é um fator impulsionador; aderência a sistemas e processos de controles, a cada dia mais relacionadas a TI, é imprescindível na administração moderna, em ambiente regulatório atual e as mudanças futuras; ainda, por conta da segurança e eficiência.

Continua Filho (2022), na linha estratégica de sustentabilidade corporativa e governança, foi verificado o nítido aumento de empresas listadas com as melhores práticas de governança e ESG, conforme o índice IGC (Índice de Governança Corporativa) obtido nos últimos dez anos.

Dentre os processos levantados pela DTT (2023), para 23% das corporações é desafiador a incorporação de processos tecnológicos às atividades. Ou seja, apesar do avanço do uso da TI, ainda exige esforços no âmbito das implementações da tecnologia.

Siga em Frente...

Projeto de software e a governança de TI

Estudante, entendamos como você pode alinhar o projeto de software à governança, visando processos mais eficientes. Inúmeras ferramentas apoiam no dia a dia das organizações, como: análise de dados, automação de tarefas e atendimento aos investidores; todas baseadas em ferramentas de IA (inteligência artificial).

A análise de dados está diretamente associada aos processos otimizadores por indicadores presentes em programas de qualidade, tanto no desenvolvimento de software quanto em produção de bens e serviços das empresas. Quanto à automação de tarefas, se percebe no uso de ferramentas CASE em construção de software, e largamente utilizada na alta performance

dos processos industriais. E, por fim, no atendimento aos investidores, é evidente que muitos relatórios são

preparados a um dos principais Stakeholders de projetos, o monitoramento das entregas e a continuidade do projeto são processos interdependentes.

Atualmente, segundo verificada na pesquisa da DTT (2023), sob a perspectiva da regulamentação de influenciadores digitais, 27% das empresas esperam cumprir com o compromisso com a transparéncia e com a evolução da agenda ESG. Tendo como possibilidade, a análise de riscos a partir do uso das mídias sociais, por investidores e do impacto em tomadas de decisões em investimentos.

Dentre os fatores que mostram relevantes, relacionados a projeto de software, segundo DTT (2023), estão: padronização de relatórios e indicadores ESG, novos meios de comunicação com investidores, monitoramento de sinais de alerta precoce e maiores exigências em governança, riscos e controles.

Ainda, percebem-se vários aspectos que desafiam o monitoramento de indicadores ESG, conforme DTT (2023): falta de padronização dos dados, falta de ferramentas para coleta de informações e informações pulverizadas.

Os estudos em governança de TI continuam em plena evolução, portanto, estudante, sugerimos ficar atento para atender às organizações em processos relacionados a investimentos e às práticas da governança em seus projetos de software.

Vamos Exercitar?

Se o objetivo da ChocoRock, fábrica de chocolate, é melhorar a transparéncia em sua cadeia de suprimentos para obter investimentos em seus projetos de software, chegou a hora de colocar ações diretamente relacionadas à gestão e à Governança.

Você sabe que as informações de um sistema devem estar armazenadas adequadamente para recuperá-la no momento correto para tomada de decisões, certo? E que pessoas devem proceder seus processos rotineiros e, em sistemas, conforme as necessidades estratégicas da organização.

Pois bem, chegou a hora de colocar em prática, aproveitando seu conhecimento de valores ágil, processos PMBoK, ITIL e de Governança de TI e de ESG para o seu projeto ser inovador e principal ferramenta de sucesso.

Do PMBoK, aproveite a gestão de riscos e gestão de comunicação para implementar a base de um projeto detalhista necessário aos processos da cadeia de suprimentos da ChocoRock, sempre praticando o senso ágil timeboxing e da auto-organização, disseminando a prática do processo puxado, interagindo constantemente com os Stakeholders.

PROJETO DE SOFTWARE

Procure se colocar à disposição do time de operação, ou ao Product Owner, soluções para implementação de ferramentas que incorporem processos tecnológicos às atividades, visando a geração de informações essenciais para a tomada de decisões em governança ESG.

Seja o protagonista de soluções digitais em processos do desenvolvimento de sistemas e no apoio às organizações com seus desafios da inovação e na implantação de ferramentas da gestão corporativa.

Bons estudos!

Saiba mais

1. Ferramentas ágeis para diminuir riscos

Leia o item 19.3 Projeto Preditivo com Utilização das Ferramentas do Ágil, na página 169, as ferramentas do ágil que podem ser incorporadas no projeto para diminuir os riscos.

MAXIMIANO, A. C. A.; VERONEZE, F. [Gestão de projetos: preditiva, ágil e estratégica](#). Barueri: Atlas, 2022.

2. Paradigma e evolução da ESG

[Evolução da agenda ESG - O impacto desta transformação na atividade dos Ris](#)

3. Decisões consideradas importantes em ESG

Leia o tópico, 1.4 O contexto atual da Governança e ESG – caso brasileiro, na página 24 a 37, sobre equilíbrio entre pessoas, planeta e lucro.

FILHO, R. I.; CIERCO, A. A. [Governança, ESG e estrutura organizacional](#). São Paulo: Actual, 2022.

Referências

CAVALCANTI, F. R. P. **Fundamentos de gestão de projetos:** gestão de riscos. São Paulo: Atlas, 2016.

FILHO, R. I.; CIERCO, A. A. **Governança, ESG e estrutura organizacional**. São Paulo: Actual, 2022.

MAXIMIANO, A. C. A.; VERONEZE, F. **Gestão de projetos:** preditiva, ágil e estratégica. Barueri: Atlas, 2022.

PRESSMAN, R. S.; MAXIM, B. R. **Engenharia de software**. Porto Alegre: AMGH, 2021.

Aula 5

Encerramento da Unidade

Videoaula de Encerramento

Este conteúdo é um vídeo!



Para assistir este conteúdo é necessário que você acesse o AVA pelo computador ou pelo aplicativo. Você pode baixar os vídeos direto no aplicativo para assistir mesmo sem conexão à internet.

Dica para você

Aproveite o acesso para baixar os slides do vídeo, isso pode deixar sua aprendizagem ainda mais completa.

Estudante, esta videoaula foi preparada especialmente para você. Nela, você irá aprender conteúdos importantes para a sua formação profissional. Vamos assisti-la?
Bons estudos!

Ponto de Chegada

Olá, estudante!

Boas-vindas aos conceitos fundamentais de projeto de software avançado, em que você deve compreender o uso de projeto de software para desenvolver aplicações alinhadas à governança de TI, contemplando técnicas modernas de construção da interface do usuário e, também, como atender a alta demanda de mudanças e evoluções.

As evoluções e mudanças em software são inevitáveis. Portanto, ser competente em gestão de mudanças é uma experiência que você deve colocar como meta profissional. Estar atento às novas regras de negócio, ou alteradas a curto prazo, é extremamente importante para adaptar os processos do desenvolvimento via ferramentas que dão suporte ao grande volume de modificações dos códigos-fonte.

Sempre que falamos de interface do sistema, o projeto da interação do usuário com o software deve estar muito alinhado às operações e aos negócios. Já se pegou pesquisando e compreendendo as tendências e conhecendo os perfis dos usuários? Então, convide o usuário para uma dinâmica da experiência!

PROJETO DE SOFTWARE

Muitas vezes nem ele sabe como deve ser a interface. Basta colocá-lo frente a um protótipo com os comportamentos e ele passará a experienciar com o protótipo de maneira natural, aflorando os movimentos que serão devidamente registrados, para ser a base da construção do projeto de interface.

A criação de diferenciais em negócios, sob os recursos da tecnologia da informação, está proporcionando grande oportunidade para qual você estará preparado em breve. Ao compreender a governança de TI e o ESG, você estará capacitado a implementar processos para gerar valor com os sistemas. Esses valores são imprescindíveis para projetos inovadores. São valores que os patrocinadores apoiam, por saberem que as equipes competentes em governança conseguirão oferecer inovação em seus produtos.

Outro ponto fundamental na gestão avançado de projeto é o uso das diretrizes da observabilidade. Por meio da observabilidade, o processo de desenvolvimento poderá ser monitorado digitalmente. Porém, a grande vantagem quando se está implementado nos sistemas, é porque esse processo gerará informações essenciais na análise para preparar as empresas estrategicamente.

É Hora de Praticar!



Este conteúdo é um vídeo!

Para assistir este conteúdo é necessário que você acesse o AVA pelo computador ou pelo aplicativo. Você pode baixar os vídeos direto no aplicativo para assistir mesmo sem conexão à internet.

Boas-vindas ao CoffeeX, projeto de software, inserido no berço do norte paranaense, local sagrado nas décadas de 1960 a 1970, quando o mundo conheceu o sabor incrível do café brasileiro. A nova geração utiliza o produto, na criação de bebidas gourmet, respeitando as causas ambientais, sociais e da governança.

A proposta do sistema CoffeeX é abraçar o ciclo de vida dessa bebida, desde a seleção de sementes do café, manejo, colheita, transformação, entrega, até a fidelização do consumidor. Aproveitando que o setor agrícola se encontra numa fase avançada, atualmente é possível identificar e obter informações acerca do terreno plantado (GPS), nessa área quais produtos (veneno e adubo) foram utilizados e os recursos pessoais ou máquinas foram empregados. É também praticável o rastreamento desde a manufatura da transformação do grão do café *in natura* até chegar à xícara.

Percebida a extensão do ciclo do café, você é um dos responsáveis para implementar soluções sistematizadas para operacionalizar todo ciclo, passo a passo, com todos os detalhes e variações, possibilitando o controle e monitoramento das operações.

Encaremos esse desafio!

PROJETO DE SOFTWARE

- Seria realmente necessário configurar um ambiente SCM para o desenvolvimento de software? Em caso afirmativo, qual vantagem tenho com esse recurso?
- A experiência do usuário é muito importante na construção da aplicação, apenas para o design e a aparência da tela?
- Como reunir as necessidades de evolução e mudanças e os elementos da governança de TI, com todas as regras de negócio do software?

A primeira atitude como gestor ou desenvolvedor de software, em um projeto tão minucioso, é se colocar à disposição dos *stakeholders* para implementar processos adicionais no software quanto ao impacto ambiental que uma lavoura proporcionará, sem o devido cuidado. Simultaneamente, reconhecer que está preparado para conhecer os procedimentos que podem afetar a sociedade, como também estar em compliance nas esferas municipais, estaduais, federais e mundiais.

Para o melhor aproveitamento e produtividade no software, para a elaboração do projeto de interface, planeje com o time de operações as ações para estudar, levantar e catalogar o máximo da experiência do usuário, deixando-os à vontade para novas adaptações que o sistema venha necessitar, em *design*, à medida que o projeto avance.

Na concepção do software, você deve implementar os processos de SCM de maneira sistemática e integral. Desde o começo, adicionar todos os itens configuráveis: projetos, modelos, documentos, protótipos, código-fonte, configuração, enfim, artefatos relacionados ao software, ou aqueles que serão relacionados ou incluídos em novas versões. Todos os procedimentos viáveis de registro, com métricas definidas, farão parte das regras da observabilidade, visando um software controlado e um processo de desenvolvimento monitorado.

Estudante, compartilhe e estimule seu time no cumprimento dos prazos de projeto e na auto-organização, visando a execução das tarefas do desenvolvimento de software com eficiência e transparência.

O infográfico apresenta os tópicos essenciais que você estudou, ilustra resumidamente um cenário no qual o projeto de software poderá ser suportado pelas técnicas modernas e avançadas de gestão.

Projeto de Software Avançado



● Ásia ● Europa ● América ● Demais

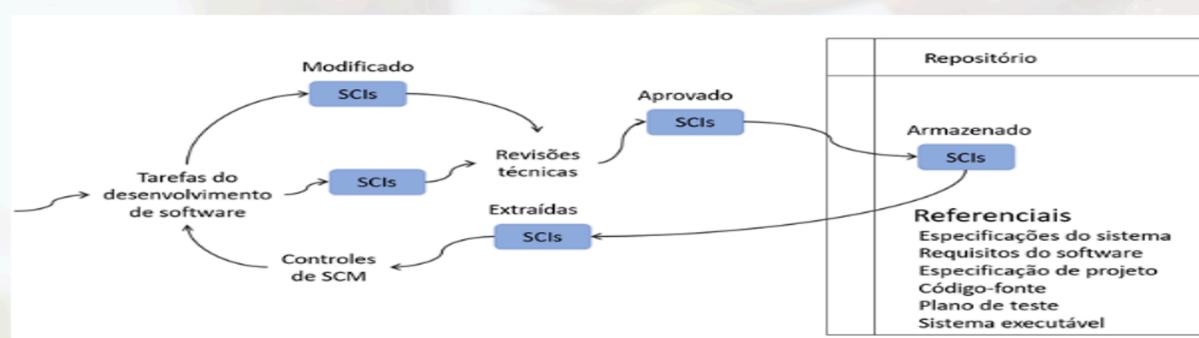
Controle Versão Evolução ESG Gestão Mudanças Interação CI/CD UX Observabilidade

Protótipo da interface UX → UX2 → UX-n

O time de TI e time de operações combinam com o designer para reunir as melhores experiências dos usuários, especialistas, clientes, produtores, consumidores, patrocinadores, para definir padrões, com refinamento iterativo. Os clientes e consumidores podem possuir perfis específicos para acompanhar a sua experiência ao longo do tempo, inclusive com os registros em visitações na propriedade agrícola, fonte do café que ele está consumindo.

Controle de versão com CI/CD

Cada SCI sendo controlado no projeto de software, garante a rastreabilidade dos artefatos, durante o desenvolvimento e após a entrega. O time de operações visualizam os itens sendo definidos, versionados, finalizados e entregues, a cada iteração (Sprint), aumentando a confiabilidade do processo. Inclusive o feedback registrado em registros do log permitido pela **Observabilidade** são integrados na versão do software.



Governança de TI e ESG

A premissa básica da perspectiva da Dependência de Recursos é que as decisões são tomadas dentro das organizações, mas atentas ao ambiente externo, pelos aspectos Ambiental, Social e Governança. Os níveis de compliance devem acontecer em todos os níveis, equilibrando as ações gerencial e corporativa.



Figura 1 | Projeto de Software Avançado
Bons estudos!

PROJETO DE SOFTWARE

AWS – Amazon. **Conceitos para detecção de anomalias ou valores atípicos.** Disponível em: https://docs.aws.amazon.com/pt_br/quicksight/latest/user/anomaly-detection-outliers-and-key-drivers.html. Acesso em: 15 nov. 2023.

CAMARGO, R.; RIBAS, T. **Gestão ágil de projetos.** São Paulo: Saraiva Educação, 2019.

CAVALCANTI, F. R. P. **Fundamentos de gestão de projetos:** gestão de riscos. São Paulo: Atlas, 2016.

MAXIMIANO, A. C. A.; VERONEZE, F. **Gestão de projetos:** preditiva, ágil e estratégica. Barueri: Atlas, 2022.

PRESSMAN, R. S.; MAXIM, B. R. **Engenharia de software.** Porto Alegre: AMGH, 2021.