

NÃO PODE FALTAR

DOCUMENTAÇÃO DO PROJETO

Daniela Teresa Rossignoli Uebele



Fonte: Shutterstock.

Deseja ouvir este material?

Áudio disponível no material digital.

PRATICAR PARA APRENDER

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

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

Para isso, temos, hoje, o recurso tecnológico a nosso favor, não é verdade? Podemos documentar e acessar as documentações em tempo real e compartilhá-las, o que traz a você, futuro desenvolvedor de uma equipe criadora de software, grandes facilidades, no entanto, agora, é preciso entender os motivos que tornam a documentação um elemento que define e melhora a qualidade do seu trabalho e de sua equipe, perceber o quanto a sua falta ou a má documentação gera grandes riscos de se ter projetos inadequados ou mesmo projetos encerrados sem a sua conclusão.

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

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

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

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

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

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

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

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

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

CONCEITO-CHAVE

RISCOS PELA AUSÊNCIA DE DOCUMENTAÇÃO

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

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

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

necessidades do cliente, havendo, por consequência, o desperdício de tempo, devido ao retrabalho ou ao fato das informações estarem restritas e/ou concentradas no código fonte desenvolvido, com acesso exclusivo ao desenvolvedor, levando a equipe a refazer códigos que poderiam ser reaproveitados (SOMMERVILE, 2018).

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

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

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

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

informações em tempo integral. Essas ferramentas de integração, conhecidas também como repositório, geram o versionamento do sistema em desenvolvimento, além de um excelente controle de alterações.

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

A documentação se torna ainda mais relevante para manter as atividades da equipe alinhadas aos objetivos da organização. Portanto, o estudo sobre a questão da governança e da estratégia das organizações utilizando-se ITIL e COBIT para o processo de gestão dos recursos da área de TI em relação ao alinhamento com todos os outros setores da empresa cita também a necessidade de haver um processo ágil, ainda sugerindo a criação de documentos de formalização.

Ver anotações

ASSIMILE

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

Quadro 3.12 | Agendamento

Termo de Abertura do Projeto
Nome do Projeto – tudo que estive grifado substitui AUTOMATIZANDO AGENDAMENTO DE CONSULTAS LOGO

Controle de Versões			
Versão	Data	Autor	Notas da Revisão
1.0	01/12/2020	Izabel Santos	Daniela Uebele
1.1	07/12/2020	Daniela Uebele	Izabel Santos

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

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

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

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

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

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

(PMI, 2017).

Ver anotações

Considerando esses fatos, diga-me, então, por que não documentar um projeto de desenvolvimento de sistemas? Quanto custa um sistema bem documentado?

Quanto você valoriza documentar e organizar o seu trabalho? Se você for continuar o trabalho de alguém, imagina como seria mais simples a compreensão de tudo que já foi feito, discutido, pensado, o que deu certo e o que já deu errado!. A economia de tempo é um ponto relevante para criarmos o hábito de documentar (SOMMERVILLE, 2018).

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

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

REFLITA

Voltando a alguns assuntos já trabalhados, a gestão ágil veio como uma alternativa à gestão tradicional, mas nunca foi dito que se descartaria todo o aprendizado anterior! O objetivo maior dessa metodologia que surgiu no início do século XX, a partir do sistema de produção Toyota, tem como foco eliminar desperdícios de toda natureza e fazer entregas rápidas, contínuas e com qualidade total (TQM – total *quality management*) (SHIMOKAWA; FUJIMOTO, 2009). Baseou-se no fato de que imprevistos, problemas e riscos

são inerentes a todo e qualquer projeto, independentemente de suas proporções de complexidade e tamanho, certo? O que importa mesmo é absorver as inevitáveis e constantes mudanças? Fazer as integrais de forma constante, permitindo que o seu cliente participe do processo? Que o produto a ser entregue seja realmente útil para o que foi proposto? A documentação faz parte de todo esse processo, a qualidade e os riscos estão na falta da documentação? Muitos questionamentos, não é? Vale a busca por mais experiências (COSTA; PEREIRA, 2019).

Ver anotações

■ QUANDO DOCUMENTAR

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

Segundo Newton (2011), criar documentos padronizados, a exemplo o “termo de abertura” sugerido por PMI ou usar um “documento de iniciação de projeto”. É um raciocínio que embasa a criação destes documentos, sendo este o ponto importante, e não os templates prontos que trazem apenas lacunas a serem preenchidas (Figura 3.7).

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

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

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

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

✓ Estrutura Analítica do Projeto – Fases e principais entregas (Principais entregas, versão inicial.)

- 1 Análise
 - 1.1 Prototipagem – esboço das telas com a definição de navegação e campos necessários para o agendamento.
 - 1.2 Documentação – TAP; cronograma; análise de risco; planejamento de comunicação; análise do sistema UML.
- 2 Desenvolvimento
 - 2.1 Layout – criação dos layouts de tela com configuração de identidade visual.
 - 2.2 Banco de dados - criação do CRUD do sistema e testes com pesquisas e relatórios.
 - 2.3 Sistema de segurança - controle de log dos usuários da clínica.
- 3 Gestão de recursos – definir e controlar os recursos necessários para o desenvolvimento do sistema.

✓ Principais requisitos das principais entregas/produtos (Principais requisitos dos produtos/entregas a serem atendidos identificados na EAP acima.)

- 1 Visualização e agendamento, conforme a liberação de horário pelo médico;
- 2 Confirmação da consulta automática um dia antes;
- 3 Criar relatório da agenda do dia;
- 4 Gerar relatório no final da semana com o resumo dos atendimentos realizados por médico e plano de saúde.

Termo de Abertura do Projeto.docx Página 1 de 2
 Nome da EMPRESA Link da empresa

Fonte: adaptada de PMI (2017).

Ver anotações

Além desses itens, temos a utilização de algumas ferramentas disponíveis para análise de risco e definição das ações para os classificados como prioritários; para a definição do planejamento de qualidade, decidindo-se como serão feitas as avaliações da qualidade e quais os métodos e as ferramentas a serem utilizadas. Portanto é necessário definir como serão feitos os testes no sistema, e após todo o processo de planejamento e geração das documentações pertinentes, vale ressaltar que a utilização desses documentos e das ferramentas para esses processos será definida pela equipe conforme a necessidade de cada projeto e as características da organização (CARVALHO, 2018).

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

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

Figura 3.8 | Exemplo de solicitação de mudanças

DOCUMENTO DE SOLICITAÇÃO E CONTROLE DE MUDANÇAS			
Controle de Versões			
Versão	Data	Autor	Notas da Revisão
1.0	10/12/2020	Izabel Santos	Elaboração Inicial – primeiro rascunho
1.1	15/12/2020	Daniela Uebele	Revisão pós reunião com a funcionária
Solicitante	Izabel Santos	Prioridade [0-Maior ↔ 4-Menor]	2
Status	Pendente	Classificação	Fora do escopo

✓ Descrição sumária
 [Descreva os requisitos e as características dos produtos com as mudanças que devem ser entregues.]
 Inserção de um relatório semestral por plano de saúde contendo levantamento de atendimentos, procedimentos efetuados e valor recebido.

✓ Justificativa
 [Porque a necessidade da mudança.]
 Verificação do número de pacientes, os valores que estão sendo oferecidos por cada plano de saúde, e se a permanência no plano de saúde é pertinente.

✓ Classificação de impacto no projeto
 [Preenchido pelo solicitante ou GP conforme definido no plano de gerenciamento de projetos.]

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

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

Fonte: adaptada de PMI (2017).

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

Todos os processos do planejamento que geram documentações visam a uma forma de apoio para tomada de decisões acertadas. Verificando-se a qualidade e as modificações que o projeto venha a sofrer, criam-se parâmetros para que a equipe e os sponsor consigam determinar se os requisitos do sistema foram atendidos, sendo esse um momento oportuno para se utilizar a documentação no grupo de processo de encerramento, pois é por meio da documentação inicial que auxiliou o planejamento que é possível obter a definição de pronto para os membros da equipe, a aceitação do cliente, revisando o final de cada fase e do projeto total, ter as lições aprendidas documentadas é um meio de criar um histórico para os projetos futuros (CARVALHO, 2018; COSTA; PEREIRA, 2019; SOMMERVILLE, 2018).

Observe na Figura 3.9 as restrições, as premissas e os riscos iniciais analisados. O exemplo está bem simples, porém é suficiente para entender que o cliente tem que ter ciência do que foi acordado e que ele possui responsabilidade também

o

Ver anotações

pelo sucesso do projeto. As premissas deixam claras as questões de comunicação e de infraestrutura necessárias, bem como os gastos para se manter o sistema em funcionamento.

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

✓ Restrições (limitação aplicável ao projeto, que afetará o seu desempenho. Limitações reais: orçamento, recursos, tempo de alocação)									
Valor do projeto de R\$ 5.000,00 (cinco mil reais) Servidor do sistema, página web, segurança, banco de dados, sistema desenvolvido, toda a responsabilidade da equipe. Atendimento à questão de tempo de resposta do sistema, disponibilidade para os usuários, manutenção dentro do tempo acordado com o cliente.									
✓ Premissas (considera-se como fatos verdadeiros sem prova para fins de planejamento.)									
Os usuários terão disponibilidade para atender à equipe e fazer testes 50% do tempo do projeto. Já possui infraestrutura e conhecimento da utilização do sistema solicitado. A comunicação se dará por meio de e-mail ou sistema de conversa instantânea e reuniões, e se houver necessidade de alterações, com documentação pertinente.									
✓ Riscos (Descreva os principais riscos do projeto.)									
Mudança no cenário financeiro inicial. Usuários perderem o interesse no sistema. Surgimento de sistema comercial equivalente.									
✓ Orçamento do Projeto (Estimativa preliminar dos custos do projeto representada pelo orçamento ou pelo fluxo de caixa com suas principais entradas e saídas financeiras. Base para a aprovação financeira do projeto e da formação da linha de base dos custos)									
Aprovações									
<table border="1"> <thead> <tr> <th>Participante</th> <th>Assinatura</th> <th>Data</th> </tr> </thead> <tbody> <tr> <td>Patrocinador do Projeto</td> <td>Consultório ??????????</td> <td></td> </tr> <tr> <td>Gerente do Projeto</td> <td>Daniela Teresa Rossignoli Uebel</td> <td>14/12/2020</td> </tr> </tbody> </table>	Participante	Assinatura	Data	Patrocinador do Projeto	Consultório ??????????		Gerente do Projeto	Daniela Teresa Rossignoli Uebel	14/12/2020
Participante	Assinatura	Data							
Patrocinador do Projeto	Consultório ??????????								
Gerente do Projeto	Daniela Teresa Rossignoli Uebel	14/12/2020							
Início do Projeto: xx/xx/20xx Fim Previsto do Projeto: xx/xx/20xx									

Fonte: adaptada de PMI (2017).

POR QUE DOCUMENTAR?

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

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

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

• Ver anotações

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

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

ASSIMILE

Vamos reforçar apenas dois dos doze princípios ágeis:

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

Portanto, são 2 momentos distintos para se gerar a documentação necessária, o de **incremento**, gerado durante o desenvolvimento, interessante para que se adicione o critério de definição de pronto (*Definition of done*), garantindo que nada estará pronto sem a devida documentação; ou a documentação de **backlog**, em que o *Product Owner* fará análise de prioridade, decidindo o momento de disponibilizar para a equipe de desenvolvimento. O que é possível notar é que a documentação passou a ser um resultado do desenvolvimento e não mais um insumo.

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

o

Ver anotações

```
1 <?php
2 require_once ("connection-config.php");
3 /*
4 Compatibilidade: PHP 5.2, 5.3, 5.4, 5.5, 5.6, 7.0, 7.1, 7.2
5 */
6 try {
7 // $stringConexao = $driver. ":host=" . $server . ";dbname=" .
$database;
8 // $conn = new PDO($stringConexao, $user, $password);
9 // Padrão sem se preocupar com acentuação
10 // $conn = new PDO("$driver:host=$server;dbname=$database", $user,
$password);
11 // Novo e obrigatório formato para que possamos usar os acentos!
12 //-----
13 //-----$conn = new PDO("$driver:host=$server;dbname=$database;charset=utf8",
$user,
14 $password,
15 array(PDO::MYSQL_ATTR_INIT_COMMAND => "SET NAMES utf8"));
16 //-----
17 // PONTE ENTRE O PHP E O BANCO DE DADOS
18 //-----
19 // Instancia um novo objeto do tipo PDO
20 // Indicando que deverá este objeto conectar ao banco, utilizando o
driver do sgbd mysql,
21 // Acessando a base de dados chamada dbphp, no servidor indicado como
localhost e utilizando as credenciais de acesso (user=root/password =
usbw)
22 //-
23 // Modifico atributos /propriedades do objeto de conexão informando que
os erros devem ser tratados utilizando/enviando EXCEPTION
24 // Outros possíveis tipos de Atributos de ERRO!!
25 // PDO::ERRMODE_SILENT -- Fica quietinho e não fala que deu erro -- MAS
GERA LOG!
26 // PDO::ERRMODE_WARNING -- ENVIA uma mensagem de ALERTA -- MAS GERA LOG!
27 // PDO::ERRMODE_EXCEPTION -- Envia uma EXCEPTION que será tratada dentro
de um try catch! -- MAS GERA LOG!
28 //-
29 }
30 catch( PDOException $Exception ) {
31 echo "Erro: " . $Exception->getMessage() . " - Código: " .
$Exception-
32 >getCode();
33 die;
34 }
35 ?>
36 CÓDIGO FONTE PROFISSIONAL
37 //Daniela Uebele - cel.:(99)99999-9999 - data da conclusão 13/02/2018
38 //Muito mais indicado para conexão por ser o mais seguro.
39 //http://php.net/manual/pt_BR/class pdo.php
40 //Alterado por Andre Silva em cel (99)99999-9999 - data da conclusão
```

20/12/2020

Ver anotações

```

41 //para funcionar nas versões atuais do PHP
42
43 <?php
44 require_once ("connection-config.php");
45 //Compatibilidade: PHP 5.2, 5.3, 5.4, 5.5, 5.6, 7.0, 7.1, 7.2
46 try {
47 //Novo e obrigatório formato para que possamos usar os acentos!
48 $conn = new PDO("$driver:host=$server;dbname=$database;charset=utf8",
49 $user, $password,
50 array(PDO::MYSQL_ATTR_INIT_COMMAND => "SET NAMES utf8"));
51 /*PONTE ENTRE O PHP E O BANCO DE DADOS - Instancia objeto PDO, driver
52 do sgbd mysql, bdiif dbphp, credenciais de acesso (user=root/password =
53 usbw)
54 */
55 $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
56 }
57 catch( PDOException $Exception ) {
58 echo "Erro: " . $Exception->getMessage() . " - Código: " .
59 $Exception-
60 >getCode();
61 die;
62 }
63 ?>
```

Ver anotações

Fonte: Sommerville (2018).

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

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

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

MOTIVOS PARA CRIAR DOCUMENTOS IMPRETERIVELMENTE

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

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

Dentro do assunto documentação, existe uma preocupação que é bem pouco citada, já que, em geral, em sala de aula, falamos de pequenos projetos, em que o desenvolvimento ocorre de forma individual ou dentro de um grupo de alunos, simulando uma pequena empresa, onde os desenvolvedores estão no mesmo espaço, trabalhando lado a lado, mas é mais do que notório que a cada dia o *home office* acontece com maior frequência, então, levantamos outra questão!

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

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

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

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

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

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

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

- Astah UML: antigo JUDE, versões gratuitas e pagas desenvolvidas pela Change Vision;
- Enterprise UML: desenvolvido pela Sparx Systems; compatibilidade com linguagens como Java e C#; possui a função de criar código e engenharia reversa.
- Visio: está integrado ao pacote Office da Microsoft; possui outras representações gráficas suportadas.

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

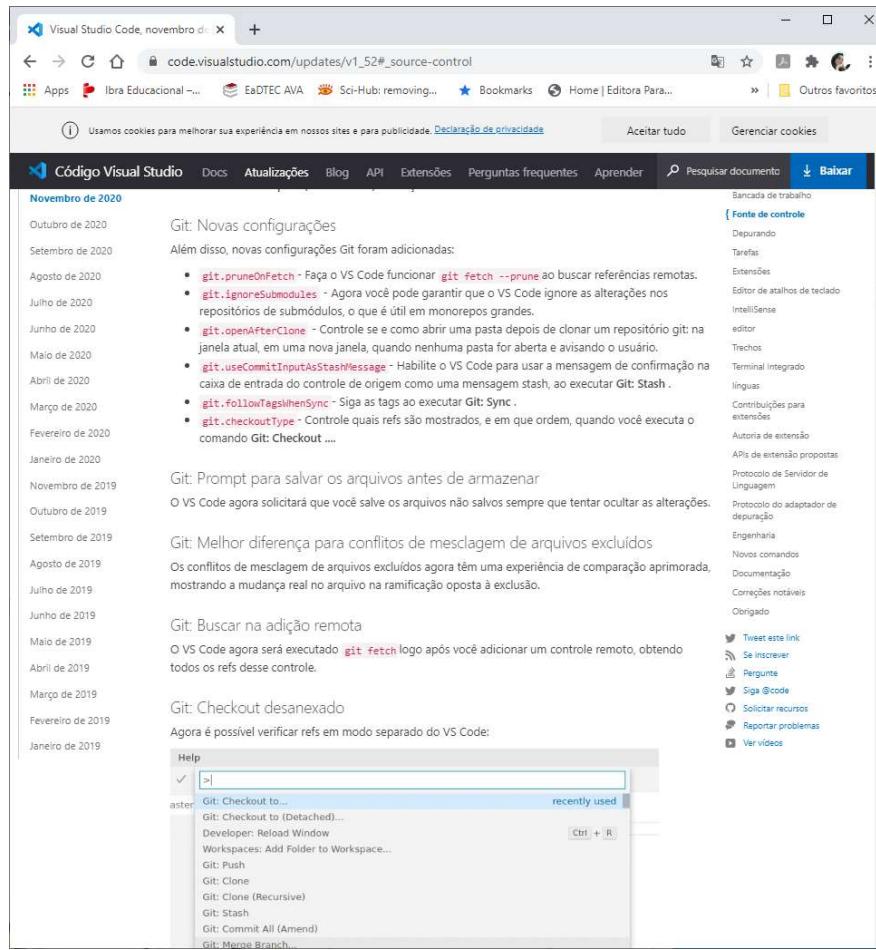
Após a criação da documentação técnica e análise, passa-se à utilização de editores para o desenvolvimento de software. Verifique que as ferramentas são constantemente tratadas como documentos independentes, e esse fato não procede; muitos deles oferecem recursos que são ignorados, como o de versionamento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

Figura 3.10 | Visual Studio Code trazendo as últimas atualizações



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

Figura 3.11 Atualização com novas funcionalidades para o uso do Git, detalhando e oferecendo pequenos vídeos demonstrativos, observe no final da imagem



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

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

A questão não é apenas documentar, mas como e onde documentar. São utilizadas em sala de aula algumas ferramentas desenvolvidas para um Ambiente Virtual de Aprendizado (AVA), como o *Teams*, da Microsoft, integrado com os aplicativos Office 365, além de recursos para a gestão de projetos, chat, entre outros recursos, bem como o Google Classroom integrado com o Google Drive, o Docs, o Sheets, o Slides e o Hangout, trazendo uma agenda comum à equipe. Tratam-se de plataformas que oferecem recursos de reuniões *online*, compartilhamento de arquivos, programas web de organização, editores de textos, planilhas eletrônicas, entre inúmeros recursos que atende à questão de trabalho colaborativo entre os membros de uma equipe (SOMMERVILLE, 2018).

O que essas ferramentas trazem é um repositório, no entanto, existem alguns recursos que são integrados e ele, e esse é um ponto importante a ser trabalhado. Além dessas ferramentas voltadas para as instituições de ensino, existem os softwares profissionais que atuam também com o versionamento, como o conhecido Git, o Mercurial, TFS (*Team Foundation Server*), Subversion e CVS, que

foi a primeira ferramenta de controle de software, desenvolvida em 1968, para o controle de versionamento. Mas por que fazer um versionamento de sistema? Quais os benefícios? (WAZLAWICK, 2013).

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

Ver anotações

ASSIMILE

Como fazer um versionamento no software?

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

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

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

Ver anotações

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

DOCUMENTAÇÃO COMO ELEMENTO DE QUALIDADE

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

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

O documento de controle integrado de mudanças, como é citado no PMBOK, oportuniza a verificação de um ciclo para o controle das alterações, iniciando com a solicitação da mudança, passando para uma análise do impacto no projeto e tendo o *feedback* de aprovado ou de reprovado. No caso de aprovado, passa para

o replanejamento, sendo revisto os riscos, o cronograma, os custos, a forma de se analisar se o requisito será atendido por completo (DoD) e a qualidade esperada para essa nova funcionalidade ou para a alteração. Após esse processo de atualização de todos os documentos, deve-se informar os *stakeholders* e executá-lo (NEWTON, 2011).

Por fim, voltamos a falar sobre o registro das lições aprendidas, que trará à equipe foco no desenvolvimento embasado em melhores práticas, otimizando o tempo e os recursos a cada nova atividade a ser desenvolvida, o que trará produtividade e eficácia. Cabe ressaltarmos que a documentação das lições aprendidas inclui absolutamente tudo que deu certo e o que deu errado (PMBOK, 2017).

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

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

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

ASSIMILE

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

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

• Ver anotações

FAÇA VALER A PENA

Questão 1

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

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

Considerando o contexto apresentado, assinale a alternativa correta.

a. Apenas as afirmativas I e II estão corretas.

b. Apenas a afirmativa I está correta.

c. Apenas a afirmativa II está correta.

d. Apenas as afirmativas I e III estão corretas.

e. As afirmativas I, II e III estão corretas.

Questão 2

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

- () O desenvolvimento ágil reduz alguns riscos, como os de mudanças de requisitos.
- () O desenvolvimento ágil independe de pessoas, portanto, a rotatividade de desenvolvedores não trará efeitos significativos para o software a ser desenvolvido e para o projeto.

() A falta de documentos formais e a comunicação informal dificultam a continuidade e o ritmo no caso de desenvolvedores-chave deixarem o projeto.

Assinale a alternativa que apresenta a sequência CORRETA.

a. V – V – F.

b. F – V – V.

c. V – F – V.

d. F – V – E.

e. V – F – E.

o
Ver anotações

Questão 3

“O saber adquirido guia o profissional, direcionando-o a soluções para problemas que já ocorreram, e que podem ser evitados com algum tratamento específico. Os acertos do passado podem ser replicados no presente, e os do presente, no futuro” (COSTA; PEREIRA, 2019, p. 214)

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

PORTANTO

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

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

a. As asserções I e II são proposições verdadeiras e a II não é uma justificava da I.

b. As asserções I e II são proposições verdadeiras e a II é uma justificava da I.

c. A asserção I é uma proposição falsa e a II é uma proposição verdadeira.

d. A asserção I é uma proposição verdadeira e a II é uma proposição falsa.

e. As asserções I e II são proposição falsas.

REFERÊNCIAS

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

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

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

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

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

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

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

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

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

o

Ver anotações