

Qualidade do Processo de Software. Qualidade do Produto de Software. Processo de Garantia da Qualidade e Processos Relacionados

Apresentação

Um bom processo de qualidade de *software* e de produto não garante que produtos sejam produzidos com qualidade, mas pode ser um indicativo de que a empresa está preocupada em desenvolver produtos de qualidade, de modo a garantir a satisfação de seus clientes.

Nesta Unidade de Aprendizagem, você aprenderá a analisar a qualidade do processo e do produto de *software*, bem como definir o processo de garantia da qualidade de *software*.

Bons estudos.

Ao final desta Unidade de Aprendizagem, você deve apresentar os seguintes aprendizados:

- Analisar a qualidade do processo de *software* e o produto de *software*.
- Identificar o processo de garantia da qualidade de *software*.
- Definir revisões e inspeções em softwares.

Desafio

O foco principal dos desenvolvedores e gestores da Tecnologia da Informação (TI) é um código de alta qualidade; porém, a urgência nos projetos e nas cobranças de prazos impossibilita que essa tarefa seja executada com êxito. Um item que pode ajudar a alcançar esse objetivo é a revisão de *software*.

Baseado nessa afirmativa, considere o seguinte cenário:

Você é o gestor de TI de uma pequena empresa e está trabalhando com revisão de *software* de projetos em andamento.

Você recebeu um *checklist* iniciado:

ITEM	DESCRIÇÃO
01	Verificar se o código tem comentários desnecessários e retirá-los caso existam.
02	Verificar os <i>warnings</i> de compilação.
03	Verificar se os nomes dos arquivos estão de acordo com as convenções da empresa.

Sua missão é analisar o *checklist* e complementar com outros itens para a realização da inspeção.

Infográfico

Para o desenvolvimento de produtos de qualidade, os processos de *software* são importantes e devem ser seguidos e melhorados conforme a sua necessidade.

Neste Infográfico, você verá as diferenças entre qualidade do processo de *software* e a qualidade do produto de *software*.

QUALIDADE DO PROCESSO DE SOFTWARE X QUALIDADE DO PRODUTO DE SOFTWARE

Qualidade remete a algo operacional que deve estar em conformidade com as especificações que foram pré-definidas. Qualidade do produto diz respeito às características do produto, além da identificação das não conformidades para corrigir o problema assim que possível; é a especificação dos processos que serão realizados para a produção do produto.



QUALIDADE DO PROCESSO DE SOFTWARE	QUALIDADE DO PRODUTO DE SOFTWARE
Especificação dos processos que serão realizados na produção do produto.	Definição das características relevantes do produto.
Norteado pelos modelos de processos, como, por exemplo, CMMI e MPS-BR.	Norteado pelos atributos e as variáveis da qualidade.
Apresenta a certificação conforme modelo escolhido (CMMI, MPS-BR, entre outros).	Apresenta documento com a formalização das definições.
Testes funcionais não são aplicáveis.	Testes funcionais são aplicáveis.
Um processo de qualidade significa uma propensão ao produto de qualidade.	Um produto de qualidade significa a satisfação do cliente.
Precisa ser revisado frequentemente.	Precisa ser testado frequentemente.
Erros precisam ser reparados antes que afetem o produto.	Testes precisam ser feitos para evitar entregar o produto com defeito ao cliente.
O processo precisa ser aceito por todos da equipe.	O produto precisa estar de acordo com o especificado para ser aceito pelo usuário.
Está relacionado à garantia da qualidade.	Está relacionado ao controle de qualidade.



Aponte a câmera para o código e acesse o link do conteúdo ou clique no código para acessar.

Conteúdo do Livro

Processos de *software* são complexos e têm um número muito grande de atividades e características. A qualidade do processo e do produto são itens relevantes para o alcance da satisfação dos clientes.

No capítulo Qualidade do processo de *software*. Qualidade do produto de *software*. Processo de garantia da qualidade e processos relacionados, da obra *Engenharia de software*, base teórica desta Unidade de Aprendizagem, você vai estudar sobre a qualidade do processo de *software* e a qualidade do produto de *software*, além de aprender a definir o processo de garantia da qualidade e processos como revisões e inspeções em softwares.

Boa leitura.

ENGENHARIA DE SOFTWARE

A photograph showing four people in a collaborative workspace. A woman with curly hair is leaning over a table, pointing at a tablet device which displays a colorful interface. Two other individuals are seated at the table, looking at documents and the tablet. The scene is overlaid with a large orange hexagon containing the title text.

Adriana de Souza Vettorazzo

Qualidade do processo de *software*, qualidade do produto de *software*, processo de garantia da qualidade e processos relacionados

Objetivos de aprendizagem

Ao final deste texto, você deve apresentar os seguintes aprendizados:

- Analisar a qualidade do processo de *software* e o produto de *software*.
- Identificar o processo de garantia da qualidade de *software*.
- Definir revisões e inspeções em *softwares*.

Introdução

Não é viável falar de qualidade sem falar de processos. Processos de *software* são complexos e compostos por um número muito grande de atividades e características. Esses processos têm sido aliados no desenvolvimento de *software* de qualidade e, além disso, auxiliam na redução de problemas relacionados a suporte, treinamento e revisões, no aprendizado organizacional e, também, na economia, pois o esforço e o tempo de projetos podem ser reduzidos.

Neste capítulo, você vai estudar sobre a qualidade do processo de *software*, a qualidade do produto de *software* e vai aprender a definir o processo de garantia da qualidade e processos como revisões e inspeções em *softwares*.

Qualidade de processo e qualidade de produto

A qualidade de um processo de desenvolvimento está diretamente ligada à qualidade de um produto. Com o desenvolvimento de *software* não é diferente, mas esse é um processo mais criativo do que mecânico e ainda é necessário contar com as habilidades individuais e fatores externos, como, por exemplo, as novidades tecnológicas. Assim, processos de *software* são complexos e englobam um número muito grande de atividades e características (Quadro 1).

Quadro 1. Características dos processos

Característica de processo	Descrição
Facilidade de compreensão	Em que extensão o processo está explicitamente definido e com que facilidade se comprehende a definição de processo?
Visibilidade	As atividades do processo culminam em resultados claros, de maneira que o progresso do processo seja visível externamente?
Facilidade de apoio	Em que extensão as ferramentas CASE podem ser usadas para apoiar as atividades de processo?
Aceitabilidade	O processo definido é aceitável e usável pelos engenheiros responsáveis pelos produtos de <i>software</i> ?
Confiabilidade	O processo está projetado de maneira que seus erros são evitados ou percebidos antes que resultem em erros de produto?
Robustez	O processo pode continuar apesar do surgimento de problemas inesperados?
Facilidade de manutenção	O processo pode evoluir para refletir requisitos de mudanças organizacionais ou aprimoramentos de processo identificados?
Rapidez	Com que rapidez o processo de entrega de um sistema pode ser concluído com base em determinada especificação?

Fonte: Adaptado de Sommerville (2011).

No entanto, manter processos atualizados e revisados pode ser a garantia de um produto de qualidade e de satisfação do cliente.

Segundo Sommerville (2011), para um gerenciamento de qualidade de processo, é necessário envolver:

- definição de padrões de processo, como “quando” e “como” as revisões devem ser conduzidas;
- monitoramento do processo do desenvolvimento para assegurar que os padrões estão sendo seguidos;
- relato do processo de *software* para a gerência de projeto e para o comprador do *software*.

Para um gerenciamento efetivo, os padrões de qualidade são muito importantes e podem ser internacionais, nacionais, organizacionais ou de projeto.

Os padrões de produto definem as características que os componentes devem ter, como, por exemplo, um estilo de programação. Já os padrões de processos vão definir como o processo de *software* deve ser instituído.

A utilização desses padrões evita a repetição de erros e, também, auxilia na continuidade, caso a produção do produto seja continuada por outros membros da equipe. Veja, no Quadro 2, os padrões de produto e de processo.

Quadro 2. Padrões para produto e processos

Padrões de produto	Padrões de processo
Formulário de revisão de projeto	Conduta de revisão de projeto
Estrutura de documento de requisitos	Envio do documento para o CM
Formato do cabeçalho de método	Processo de liberação de versão
Estilo de programação em Java	Processo de aprovação de plano de projeto
Formato de plano de projeto	Processo de controle de mudança
Formulário de solicitação de mudança	Processo de registro de teste

Fonte: Adaptado de Sommerville (2011).

A qualidade do processo de *software*, além do foco no aumento da qualidade do produto, visa, também, diminuir o retrabalho e aumentar a produtividade.

Para implantar um programa de qualidade, é necessário implantar um processo de *software* que deve ser documentado, compreendido e seguido por todos na organização.

Garantia da qualidade de *software*

A garantia da qualidade de *software* é um padrão sistemático e planejado de ações que são impostas para garantir essa qualidade. Esse padrão busca produzir *softwares* com a menor taxa de defeitos, utilizando procedimentos, modelos, padrões e investimento na equipe — a garantia tem natureza proativa.

A garantia visa, também, o gerenciamento e a melhoria dos processos existentes por meio do aprendizado em cada projeto realizado.

O controle da qualidade tem como objetivo evitar que produtos defeituosos sejam entregues aos clientes, tem ação reativa e objetiva o monitoramento do processo, a detecção e a correção dos defeitos.

Para garantir a qualidade, algumas ações são necessárias:

- **aplicar modelos técnicos** — esses modelos ajudam o analista a adquirir uma especificação de qualidade e ajudam o projetista a desenvolver um projeto de qualidade;
- **aplicar padrões e procedimentos formais** — essa etapa se refere à aplicação de padrões nos processos de engenharia de *software*;
- **anotar e manter registros** — ações de coleta e disseminação de informações de garantia de qualidade de *software*.

O processo de garantia da qualidade contempla:

- a revisão dos produtos intermediários em comparação com os requisitos de qualidade preestabelecidos;
- a revisão das atividades comparando com os planos preestabelecidos, que podem ser, entre outros:
 - plano de projeto;
 - plano de documentação;
 - plano de monitoração de riscos.

Para avaliar os requisitos de qualidade do produto e se eles estão de acordo com os planos, é necessário aplicar as revisões e os testes.

Revisões e inspeções em software

A inspeção de *software* é realizada por pessoas que verificam o *software* com o objetivo de descobrir possíveis defeitos. Essas inspeções podem ser feitas, por exemplo, nos requisitos, no projeto, nos dados de configuração, nos dados de testes e são bastante utilizadas para descobrir erros no programa.

Quando o *software* está em fase de teste, um defeito pode ser encoberto por outro e, para evitar que isso ocorra, vários testes e execuções são necessários. Alguns defeitos são recorrentes em *software*. Nessa hora, o conhecimento em programação faz diferença para processos de detecção e correção mais rápidas.

Durante o processo de validação e verificação, são utilizados as inspeções e os testes, que são técnicas complementares. As inspeções verificam a conformidade com uma especificação do *software*, mas não verificam a não conformidade com os requisitos reais do cliente e nem características não funcionais, como, por exemplo, desempenho, usabilidade e adaptabilidade.

Para inspeções em programas, é necessária uma abordagem mais formalizada, que contenha todas as revisões efetuadas, não só a título de controle, mas também para histórico e comprovação. Essas estão relacionadas à detecção de defeitos que podem ser erros lógicos, erros de sintaxe, erro na lógica de programação e, até mesmo, de não conformidade com os padrões definidos, mas não estão relacionadas com a correção dos mesmos.

Para realizar a inspeção, as condições para que esse processo ocorra precisam estar disponíveis, bem como os padrões organizacionais que são seguidos e utilizados. Conforme os erros são encontrados e reparados, deverão ser inseridos em um *checklist*, embora esse não deva ser usado a título de penalidade aos que cometem o erro e nem para avaliação de competência da equipe. Para a inspeção, Sommerville (2011) indica o processo demonstrado na Figura 1.

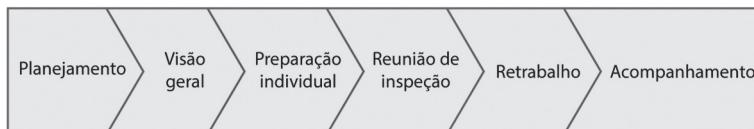


Figura 1. Processo de inspeção.

Fonte: Adaptado de Sommerville (2011).

Nesse processo:

- o planejamento discute como será feita a inspeção;
- a visão geral apresenta o sistema para a equipe que realizará a inspeção;
- na preparação, o código e a respectiva documentação são apresentados para a equipe que realizará a inspeção;
- a reunião de inspeção ocorre quando os erros descobertos são anotados no *checklist*;
- na etapa retrabalho, as modificações são feitas para reparar os erros encontrados;
- na etapa acompanhamento, uma nova inspeção é realizada se for necessário.

Veja, no Quadro 3, os papéis que os participantes da inspeção apresentam (SOMMERVILLE, 2011).

Quadro 3. Papéis na inspeção

Papel	Descrição
Autor e proprietário	Programador ou projetista responsável por produzir o programa ou o documento. Ele é responsável pela correção de defeitos descobertos durante o processo de inspeção.
Inspetor	Encontra erros, omissões e inconsistências nos programas e documentos. Pode, também, identificar questões mais amplas fora do escopo da equipe de inspeção.
Leitor	Apresenta o código ou documento em uma reunião de inspeção.
Relator	Registra os resultados da reunião de inspeção.
Presidente ou moderador	Gerencia o processo e facilita a inspeção. Relata os resultados do processo ao moderador-chefe.
Moderador-chefe	Responsável pelos aprimoramentos do processo de inspeção, pela atualização da lista de verificação, pelo desenvolvimento de padrões, etc.

Fonte: Adaptado de Sommerville (2011).

Revisões técnicas formais

As atividades de controle de qualidade de *software* que são realizadas por engenheiros de *software* são denominadas revisão técnica formal ou RTF. Segundo Pressman e Maxim (2016), os objetivos de uma RTF são:

- descobrir erros na função, na lógica ou na implementação;
- verificar se o projeto que está sendo revisado atende aos requisitos do cliente;
- garantir que o *software* está de acordo com os padrões predefinidos;
- tornar os projetos mais gerenciáveis.

Diretrizes de revisão

As revisões devem ter foco no produto, e não nas pessoas. Apontar erros não deve ser uma prática dentro das revisões formais; caso contrário, as reuniões de revisão podem causar desconforto na equipe, inclusive desmotivando os participantes.

Manter uma agenda para as revisões também é uma boa prática. Além de manter o foco da reunião, a agenda também ajuda a manter os prazos estabelecidos. Para as reuniões de revisão, é interessante limitar o número de participantes, que devem preparar-se com antecedência. Veja o Quadro 4.

Quadro 4. Itens de revisão versus resultado esperado

Revisão	Resultado esperado
Requisitos de sistema	Entendimento sobre o que o sistema deve fazer.
Requisitos do <i>software</i>	Aprovar a especificação de requisitos e iniciar projeto preliminar.
Plano de teste	Aprovar a estratégia de teste.
Projeto preliminar	Estabelecer uma linha base para o projeto.
Projeto detalhado	Aprovar o projeto detalhado e autorizar o início da codificação.
Revisão de módulos	Aprovar a finalização da implementação e o teste das unidades e, após, liberar para testes.
Teste de validação	Determinar o final dos testes de validação.
Aceitação	Aceitar o projeto e aprovar a implementação operacional.

Nesse sentido, é importante destacar a diferença entre **revisão, erro e defeito**: a revisão é o filtro do processo e é utilizada para detectar erros e defeitos; o erro é um problema de qualidade que ocorre antes da entrega ao cliente; e o defeito é um problema de qualidade constatado depois da entrega ao cliente.

Checklist

Conforme comentado, durante a inspeção, um *checklist* deve ser alimentado com os erros encontrados. Uma prática utilizada também é um *checklist* com os erros mais comuns encontrados nos projetos anteriores a fim de direcionar a inspeção atual. Veja, no Quadro 5, um exemplo de verificação de inspeção e, na Figura 2, um modelo de *checklist*.

Quadro 5. Exemplo de verificação de inspeção

Classe de defeitos	Verificação de inspeção
Defeitos de dados	<p>Todas as variáveis de programa são iniciadas antes que seus valores sejam usados?</p> <p>Todas as constantes foram denominadas?</p> <p>O limite superior de vetores deve ser igual ao tamanho do vetor ou tamanho -1?</p> <p>Se são usados <i>strings</i> de caracteres, um delimitador é explicitamente atribuído?</p> <p>Existe alguma possibilidade de <i>overflow</i> de <i>buffer</i>?</p>
Defeitos de controle	<p>Para cada declaração condicional, a condição está correta?</p> <p>Cada <i>loop</i> está terminando corretamente?</p> <p>As declarações compostas estão corretamente delimitadas entre parênteses?</p> <p>Em declarações <i>case</i>, todos os casos possíveis são levados em conta?</p> <p>Se um comando <i>break</i> é necessário após cada caso nas declarações <i>case</i>, ele foi incluído?</p>
Defeitos de entrada/saída	<p>Todas as variáveis de entrada são usadas?</p> <p>Todas as variáveis de saída têm valor atribuído antes de sua saída?</p> <p>Entradas inesperadas podem fazer com que os dados sejam corrompidos?</p>

(Continua)

(Continuação)

Quadro 5. Exemplo de verificação de inspeção

Classe de defeitos	Verificação de inspeção
Defeitos de interface	Todas as chamadas de funções e de métodos têm o número correto de parâmetros? Tipos de parâmetros reais e formais se combinam? Os parâmetros estão na ordem correta? Se os componentes acessam memória compartilhada, eles têm o mesmo modelo de estrutura de memória compartilhada?
Defeitos de gerenciamento de armazenamento	Se uma estrutura ligada é modificada, todas as ligações foram corretamente reatribuídas? Se o armazenamento dinâmico foi usado, o espaço foi corretamente alocado? O espaço de memória é liberado depois de não ser mais necessário?
Defeitos de gerenciamento de exceções	Todas as condições possíveis de erro foram consideradas?

Fonte: Adaptado de Sommerville (2011).

ITEM	ITEM PARA VERIFICAÇÃO	SIM	NÃO
Não ambíguo: É não ambíguo se, e somente se, cada requisito declarado seja suscetível a apenas uma interpretação.			
1	Cada requisito está descrito com clareza, conciso e sem ambiguidade?		
Consistente: É consistente se, e somente se, nenhum dos requisitos do documento, tomado individualmente, está em conflito com qualquer outro requisito do mesmo documento.			
2	Existem requisitos conflitantes?		
Completo: É completo se, e somente se, conter toda e apenas a informação necessária para que o software correspondente seja produzido.			
3	Existem requisitos implícitos?		
4	Os requisitos exibem a distinção clara entre funções, dados e restrições?		
5	As restrições e dependências foram claramente descritas?		
6	Existem requisitos que contêm algum nível desnecessário de detalhe do projeto?		
7	Os requisitos definem todas as informações a serem apresentadas aos usuários?		
8	Os requisitos descrevem as respostas do sistema ao usuário devido às condições de erro?		
9	Existem situações não tratadas pelos requisitos que precisam ser consideradas?		
10	O documento contém realmente toda a informação prometida em sua introdução?		

Figura 2. Modelo de checklist de validação de requisitos.

Fonte: Oliveira (2014, documento on-line).



Referências

OLIVEIRA, C. *Utilização de checklist para validação de requisitos de software*. 30 abr. 2014. Disponível em: <<https://imasters.com.br/devsecops/utilizacao-de-checklist-para-validacao-de-requisitos-de-software>>. Acesso em: 01 nov. 2018.

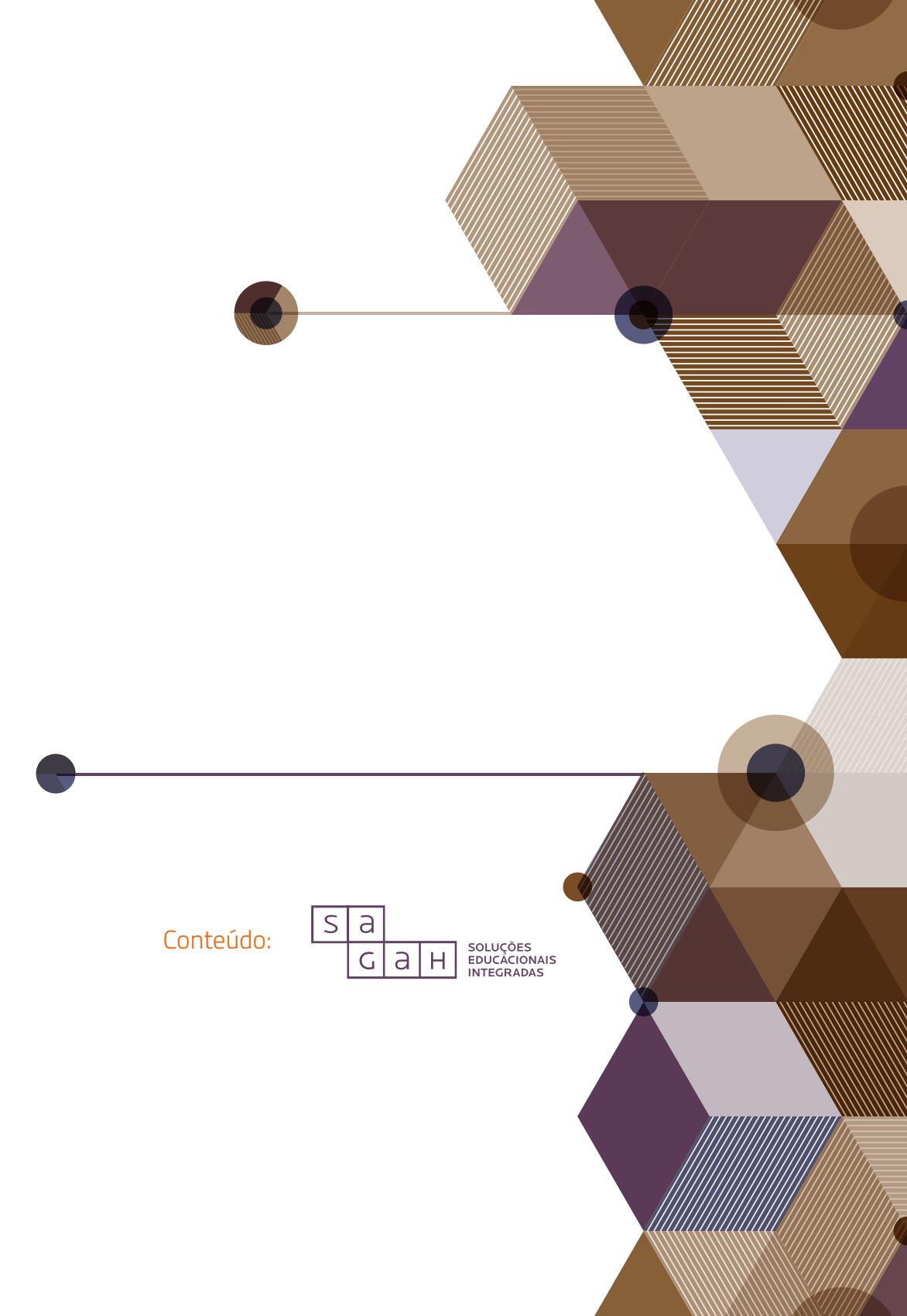
PRESSMAN, R. S.; MAXIM, B. R. *Engenharia de software: uma abordagem profissional*. 8. ed. Porto Alegre: Bookman, 2016.

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

Leitura recomendada

STEFFEN, J. B. *O que são essas tais de metodologias ágeis?* 2012. Disponível em: <https://www.ibm.com/developerworks/community/blogs/rationalbrasil/entry/mas_o_que_s_c3_a3o_essas_tais_de_metodologias__c3_a1geis?lang=en>. Acesso em: 01 nov. 2018.

Encerra aqui o trecho do livro disponibilizado para esta Unidade de Aprendizagem. Na Biblioteca Virtual da Instituição, você encontra a obra na íntegra.



Conteúdo:



SOLUÇÕES
EDUCACIONAIS
INTEGRADAS

Dica do Professor

O controle da qualidade é um item importante para a entrega de um bom produto. Já um *software* de qualidade é aquele fácil de usar, o qual não apresenta falhas em seu funcionamento.

Nesta Dica do Professor, você verá dicas sobre o controle de qualidade do *software*.



Aponte a câmera para o código e acesse o link do conteúdo ou clique no código para acessar.

Exercícios

- 1) Analise as alternativas a seguir e assinale a que corresponde a um dos objetivos das revisões técnicas formais de *software*.**
 - A) Realizar uma única reunião ao final do projeto para avaliar se o *software* foi bem construído.
 - B) Realizar reuniões com os clientes para descobrir o que deve ser feito.
 - C) Documentar os requisitos solicitados.
 - D) Garantir que o *software* não tenha erros.
 - E) Garantir que o *software* atenda aos requisitos especificados.
- 2) Analise as alternativas a seguir e assinale a que corresponde a um objetivo das inspeções de *software*.**
 - A) Revisões de progresso.
 - B) Avaliação de metas organizacionais.
 - C) Detecção de defeitos.
 - D) Revisões de cronograma.
 - E) Revisões de custo.
- 3) Para Pressman (2016), no que se refere à qualidade de *software*, as revisões, as inspeções e os testes realizados ao longo do processo de *software* para garantir que o produto satisfaça os requisitos estabelecidos, são conhecidos como:**
 - A) garantia de qualidade.
 - B) custo da qualidade.
 - C) controle de qualidade.
 - D) reengenharia de processos.

- E) *gold plate.*
- 4) No gerenciamento da qualidade, como é conhecido quando são estabelecidos padrões organizacionais e uma estrutura de procedimentos para condução de um *software* de qualidade?
- A) Planejamento da qualidade.
- B) Garantia da qualidade.
- C) Controle da qualidade.
- D) Gerenciamento da configuração.
- E) Revisões de qualidade.
- 5) As técnicas de prototipação e de revisão de requisito são as mais utilizadas para:
- A) o gerenciamento de requisitos.
- B) a validação de requisitos.
- C) o levantamento e a análise de requisitos.
- D) o estudo de viabilidade e o desenvolvimento do sistema.
- E) a especificação de requisitos.

Na prática

As revisões e inspeções de *software* fazem parte da rotina da organização que tem a qualidade implantada. Inicialmente, a revisão era apenas uma ferramenta para controle gerencial e o seu foco era a revisão de progresso do projeto; porém, havia também a necessidade de avaliar a qualidade do trabalho executado e, dessa forma, surgiram as revisões com foco nos aspectos técnicos.

Neste Na Prática, você verá como conduzir uma reunião de revisão de *software*, além de um exemplo de relatório.

Conteúdo interativo disponível na plataforma de ensino!

Saiba mais

Para ampliar o seu conhecimento a respeito desse assunto, veja abaixo as sugestões do professor:

Processo de *software* e a qualidade de produto

Leia o artigo indicado e veja como o processo de *software* influencia na qualidade do produto.



Aponte a câmera para o código e accese o link do conteúdo ou clique no código para accesar.

Qualidade, qualidade de *software* e garantia da qualidade de *software* são as mesmas coisas?

Você sabe a diferença entre os termos qualidade, qualidade de *software* e garantia de qualidade de *software*? Veja a seguir.



Aponte a câmera para o código e accese o link do conteúdo ou clique no código para accesar.

Introdução à garantia de qualidade de *software* e ferramentas para teste

Conheça uma introdução à garantia de qualidade e ferramentas de testes de *software*.



Aponte a câmera para o código e accese o link do conteúdo ou clique no código para accesar.