



DEFINIÇÃO

O planejamento, a garantia e o controle da qualidade como parte do processo de desenvolvimento de software.

PROPÓSITO

Compreender a importância do planejamento, garantia e controle da qualidade, através de técnicas adequadas de medições, seleção de métricas e realização de diferentes tipos de testes.

PREPARAÇÃO

Não existem pré-requisitos operacionais para o tema.

OBJETIVOS

MÓDULO 1

Descrever a qualidade de processo e de produto de software

MÓDULO 2

Explicar o processo da garantia de qualidade de software

MÓDULO 3

Explicar o planejamento da qualidade e o controle da qualidade de software

MÓDULO 4

Descrever as medições e métricas do software

INTRODUÇÃO

O software tem um destaque cada vez maior na vida moderna. Podemos imaginar a complexidade de um software embarcado em uma aeronave com 500 passageiros ou um software que gerencia o tráfego aéreo de uma grande cidade.

A construção de um software desta magnitude exige o envolvimento de vários especialistas, tais como engenheiros de software, web designers, administradores de banco de dados,

arquitetos de software e outros que trabalham arduamente para o sucesso do produto software.

A referida equipe tem uma composição multidisciplinar, refletindo uma característica do software: A complexidade. O trato desta complexidade requer a aplicação da Engenharia de Software, que por sua vez tem como foco a camada qualidade. A referida camada garante que os requisitos que atendem às expectativas do usuário serão cumpridos.

Nesse contexto, o presente tema descreve o planejamento, a garantia e o controle da qualidade de software.

MÓDULO 1

🕒 Descrever a qualidade de processo e de produto de software

ENGENHARIA DE SOFTWARE VS. QUALIDADE

Vamos iniciar nosso estudo sobre qualidade no contexto da área de conhecimento Engenharia de Software, que tem como principal produto o software.

A cada dia, o produto software está mais presente nas nossas vidas, quer por meio das redes sociais, em nossos veículos ou nas nossas residências.

No contexto da qualidade de software, podemos destacar a denominada “Crise do software” sintetizada pela afirmativa a seguir:

A MAIOR CAUSA DA CRISE DO SOFTWARE É QUE AS MÁQUINAS SE TORNARAM VÁRIAS ORDENS DE MAGNITUDE MAIS POTENTES! EM

TERMOS DIRETOS, ENQUANTO NÃO HAVIA MÁQUINAS, PROGRAMAR NÃO ERA UM PROBLEMA; QUANDO TIVEMOS COMPUTADORES FRACOS, ISSO SE TORNOU UM PROBLEMA PEQUENO, E AGORA QUE TEMOS COMPUTADORES GIGANTESCOS, PROGRAMAR TORNOU-SE UM PROBLEMA GIGANTESCO.

DIJKSTRA, 1972

Importante destacar que o aprimoramento tecnológico do hardware nos últimos anos permitiu o desenvolvimento de softwares cada vez mais complexos, tendo um forte impacto na indústria de software.

★ EXEMPLO

Podemos apresentar a substituição do paradigma estruturado pelo paradigma orientado a objetos, que permite o reuso intensivo de especificação, bem como uma melhor manutenibilidade e, como consequência, o desenvolvimento de softwares mais complexos.

Podemos imaginar as sérias consequências na falha de um software embarcado em uma aeronave com 500 passageiros ou em um sistema de controle de tráfego aéreo de uma grande cidade. Uma falha seria algo bem trágico!

Ambos os exemplos possuem uma característica comum: A complexidade. A melhor tratativa para a complexidade é a aplicação de metodologia que permita a decomposição do problema em problemas menores de forma sistemática, cabendo à engenharia de software essa sistematização.

A engenharia de software é uma tecnologia em camadas . Veja:



📷 Figura 1 – Camadas da Engenharia de Software.

QUALIDADE

A camada qualidade garante que os requisitos que atendem às expectativas do usuário serão cumpridos.

PROCESSO

A camada de processo determina as etapas de desenvolvimento do software.

MÉTODOS

A camada de métodos define os artefatos gerados em função da técnica de modelagem adotada.

FERRAMENTAS

A camada ferramentas estimula a utilização de ferramentas **CASE**.

CASE

É uma classificação que abrange todas as ferramentas baseadas em computadores que auxiliam atividades de Engenharia de software, desde análise de requisitos e modelagem até programação e testes.

Retornando aos nossos exemplos, qual camada evitaria uma falha?

Vejamos os seguintes argumentos e depois tire a sua conclusão.

ATENÇÃO

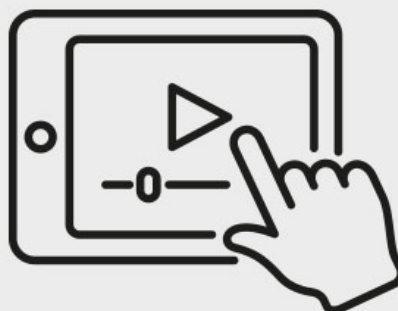
A camada base é a camada de processo, sendo a camada qualidade dependente da existência dessa, ou seja, a definição de um processo de desenvolvimento de software é condição para a implantação de um sistema de qualidade em um projeto de software.

Os métodos e ferramentas da engenharia de software servem, entre outras coisas, ao propósito de garantir, ou pelo menos facilitar, a obtenção do objetivo de garantir a qualidade no software.



Assista ao vídeo abaixo para saber um pouco mais sobre qualidade de software e as atividades de gerenciamento da qualidade no processo de desenvolvimento de software:

Para assistir a um vídeo sobre o assunto, acesse a versão online deste conteúdo.



QUALIDADE DE SOFTWARE

De acordo com Pressman (2016), a qualidade de software é definida como uma gestão de qualidade efetiva, aplicada de modo a criar um produto útil que forneça valor mensurável para aqueles que o produzem e para aqueles que o utilizam.



📷 Roger Pressman

Vamos a uma análise da citada definição:

GESTÃO DE QUALIDADE

Permite estabelecer uma infraestrutura de suporte ao desenvolvimento de software com qualidade.

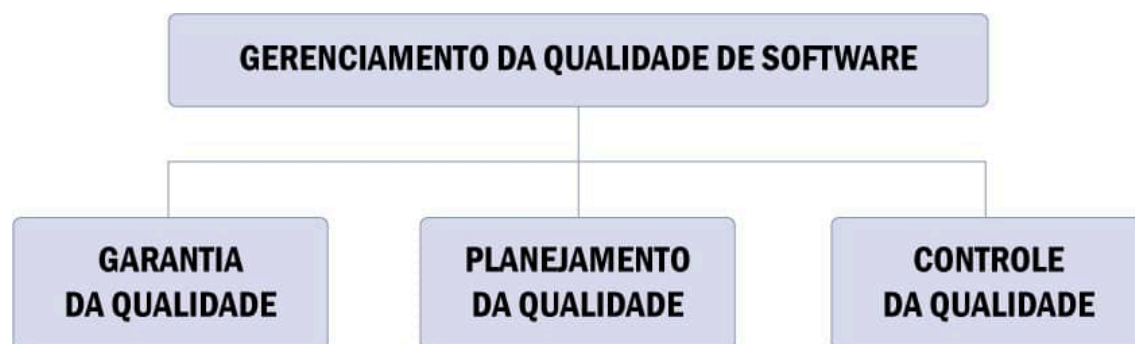
PRODUTO ÚTIL

Inclui o atendimento às necessidades dos usuários, ou seja, atendimento aos requisitos, sendo esse produto isento de defeitos.

VALOR MENSURÁVEL

Permite a agregação de valor aos usuários e ao fabricante, sendo que nesse último caso, os engenheiros de software terão menos trabalho de manutenção e mais tempo disponível na construção de novas aplicações.

O gerenciamento de qualidade de software usualmente é estruturado em três atividades principais:



Fonte:Shutterstock

📷 Figura 2 – Gerenciamento da qualidade.



Garantia de qualidade: Estabelece um conjunto de procedimentos e padrões organizacionais que permitem a geração de software com qualidade.

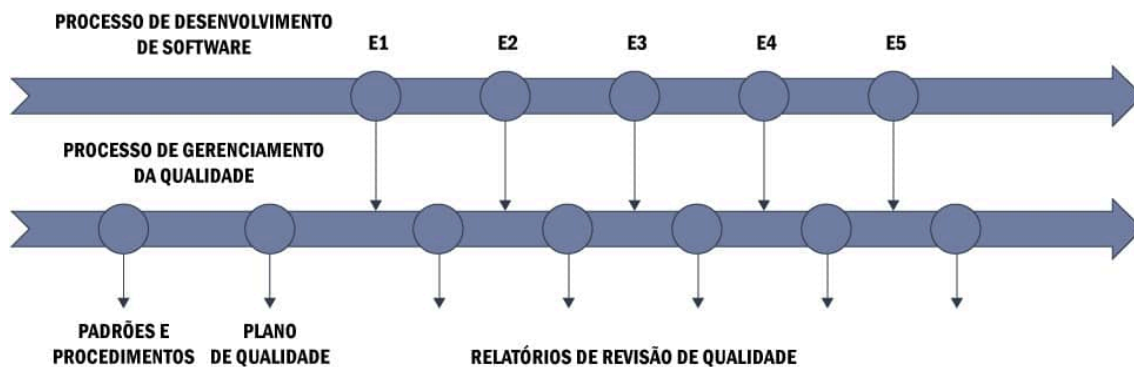
Planejamento de qualidade: Permite o ajuste dos procedimentos e padrões a um determinado projeto de software.



Controle de qualidade: Inclui a definição e aprovação de processos que assegurem que a equipe de desenvolvimento de software tenha seguido os procedimentos e os padrões de

qualidade de projeto.

A Figura 3 ilustra o relacionamento entre o processo de gerenciamento da qualidade e o processo de desenvolvimento de software proposta por Sommerville (2019). O primeiro permite uma verificação independente no segundo, verificando as entregas de artefatos do projeto. Uma melhor prática é que a equipe de garantia de qualidade seja independente da equipe de desenvolvimento, pois no caso de surgirem problemas, tal como atraso no projeto, poderia ocorrer um comprometimento da qualidade de modo a cumprir o cronograma.



📷 Figura 3 – Gerenciamento de qualidade versus processo de desenvolvimento.

Fonte: O Autor.

Podemos visualizar a qualidade de software sob duas dimensões:

PROCESSO

Podemos destacar que o processo de desenvolvimento do software exige que diversas decisões sejam tomadas ao longo do mesmo. Portanto, se desejamos produzir software com qualidade, é necessário garantir a qualidade em todas as etapas do processo.

PRODUTO

Destacamos que softwares mal testados provocam prejuízos enormes às empresas, pois um simples erro poderá, por exemplo, gerar requisições de compras desnecessárias, produzir indicadores falsos de produtividade, entre outros; podem, também, afetar o processo de tomada de decisão de gerentes e executivos que apoiam-se nas informações para minimizar riscos, direcionar esforços, promover investimentos, sempre na busca da excelência operacional, pois a qualidade das decisões está intimamente ligada à qualidade das informações disponibilizadas pelos softwares organizacionais.

QUALIDADE DO PROCESSO

Processo é uma sequência de etapas que permitem a geração de um produto, no nosso caso, o software.

ATENÇÃO

O processo permite uma melhor tratativa em relação à complexidade de obtenção de um determinado produto, pois na maioria das vezes é um trabalho multidisciplinar realizado por analistas, programadores, gerentes de projeto, gerentes de teste e outros.

A Engenharia de Software possui uma diversidade de modelos para processos de desenvolvimento de software que atendem a diferentes problemas, sendo um requisito importante na seleção de um processo a complexidade.

Basicamente, quanto maior a complexidade de um sistema, mais formal deverá ser o processo adotado.

Uma metodologia de processo genérica, segundo Pressman (2016), compreende cinco atividades:



COMUNICAÇÃO

A comunicação permite o entendimento do problema, a definição de objetivos para o projeto, bem como a identificação de requisitos.

PLANEJAMENTO

Inclui o gerenciamento de projeto.



MODELAGEM

Abrange a geração de modelos, tal como o modelo de casos de uso.

CONSTRUÇÃO

Inclui, a partir dos modelos gerados, a codificação e os testes de software de acordo com o planejado.



ENTREGA

Disponibiliza o produto em produção de acordo com o planejado.

A garantia da qualidade de um software inclui o estabelecimento de uma cultura de não tolerância a erros.

Os processos devem estar estruturados com mecanismos de detecção de defeitos nos diversos artefatos gerados durante o processo, com o objetivo de evitar a propagação de defeitos nas fases seguintes, refletindo na inserção de defeitos durante a codificação. Os artefatos incluem qualquer entrega gerada durante o processo de software, incluindo, entre outros, os documentos de requisitos, os modelos, bem como o produto software.

COMENTÁRIO

Usualmente, os testes estão relacionados ao produto software, mas podemos aplicar testes em todas as tarefas do processo de desenvolvimento de software, sendo esses denominados de testes de verificação, conhecidos também como testes estáticos. Os objetivos desses testes são os artefatos gerados durante todo o processo de desenvolvimento do software, tal como o diagrama de classes ou o diagrama de casos de uso.

ATENÇÃO

A produção de software com qualidade exige o estabelecimento de um processo de desenvolvimento de software, pois não poderemos garantir a qualidade de algo que não existe.

QUALIDADE DO PRODUTO

Pressman (2016) descreve os fatores de qualidade do produto software de McCall, sendo uma proposta de categorização dos fatores que afetam a qualidade de software. Esses fatores de

qualidade de software, apresentados na Figura 4, focam em três etapas de um produto de software: revisão do produto, transição e operação.



📷 Figura 4 – Fatores de qualidade de software de McCall.

Fonte: O Autor.

Vejamos uma descrição dos referidos fatores, cabendo destacar que a avaliação da qualidade de uma aplicação usando esses fatores possibilitará uma sólida indicação da qualidade de um software:

Correção - O quanto um programa satisfaz a sua especificação e atende às necessidades do cliente.

Confiabilidade - O quanto se pode esperar que um programa realize a função pretendida com a precisão exigida.

Eficiência - A quantidade de recursos computacionais e código exigidos por um programa para desempenhar sua função.

Integridade - O quanto o acesso ao software ou dados por pessoas não autorizadas pode ser controlado.

Usabilidade - Esforço necessário para aprender, operar, preparar a entrada de dados e interpretar a saída de um programa.



Fonte: Shutterstock

Facilidade de manutenção - Esforço necessário para localizar e corrigir um erro em um programa.

Flexibilidade - Esforço necessário para modificar um programa em operação.

Testabilidade - Esforço necessário para testar um programa de modo a garantir que ele desempenhe a função destinada.

Portabilidade - Esforço necessário para transferir o programa de um ambiente de hardware e/ou software para outro.

Reusabilidade - O quanto um programa, ou partes do mesmo, pode ser reutilizado em outras aplicações.

Interoperabilidade - Esforço necessário para integrar um sistema a outro.



A qualidade do produto pode ser garantida por meio de um sistemático plano de testes executado durante a etapa de codificação do processo de software, sendo esses testes denominados de testes de validação.

Inicialmente, ocorre a validação da estrutura interna da unidade de software, bem como sua aderência aos requisitos estabelecidos.



Em seguida, a validação da sua integração com as unidades de software desenvolvidas anteriormente, com destaque para as validações das interfaces dos componentes de software.

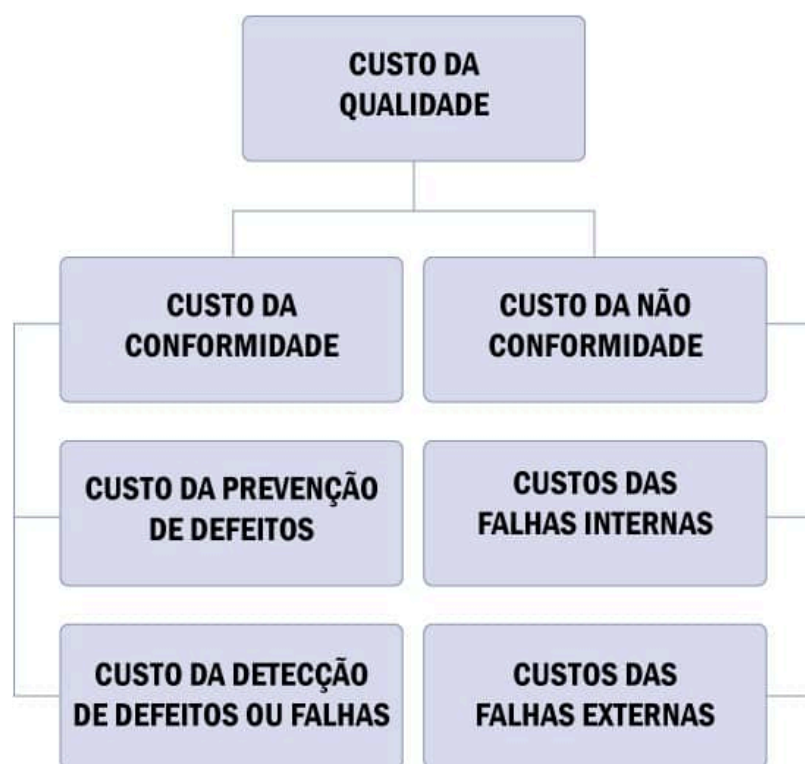


Finalmente, ocorre a validação funcional e não funcional do software como um todo, sendo os usuários envolvidos nessa etapa.

Os procedimentos de testes aplicados diretamente em softwares são também denominados de testes de software ou testes dinâmicos.

Os testes de software podem sofrer um alto nível de automação, o que possibilita a criação de complexos ambientes de testes que simulam diversos cenários de utilização. Quanto mais cenários simulados, maior o nível de validação que obtemos do produto, caracterizando maior nível de qualidade do software desenvolvido.

CUSTOS DA QUALIDADE



Fonte: Shutterstock

📷 Figura 5 – O custo da qualidade.

O custo da qualidade é composto pelo custo da conformidade e custo da não conformidade.

CUSTO DA CONFORMIDADE

Agrega o custo da prevenção de defeitos e o custo da detecção de defeitos ou falhas. O custo de prevenção inclui os esforços do sistema de qualidade em evitar a ocorrência de defeitos, tais como detalhamento de metodologias, treinamentos, utilização de ferramentas, estabelecimento de procedimentos, entre outros. O custo de detecção de defeitos, também denominado de custo de avaliação, é composto pelos custos relacionados à atividade de avaliação, detecção ou inspeção da qualidade do processo ou do produto, tais como revisões

de artefatos (requisitos, modelos, planos de testes etc), inspeções de código, testes de software, auditorias, entre outros.

CUSTO DA NÃO CONFORMIDADE

O custo proveniente da falta de qualidade, que desapareceria caso não surgisse nenhum erro antes ou depois da entrega do produto ao cliente, soma-se aos custos relacionados às correções de defeitos originados no processo de desenvolvimento de software, tais como retrabalhos, ações corretivas, atrasos nos cronogramas, perdas financeiras e operacionais, entre outros. Tais custos podem ser denominados de custos de falhas internas, ou seja, erros identificados antes da entrega do software ao cliente, e custos de falhas externas, associados aos erros identificados após a referida entrega.

Quanto mais cedo ocorrer a identificação de um defeito, menor o custo.

RESUMINDO

Neste módulo, destacamos a importância da qualidade no contexto da engenharia de software.

Vimos que a qualidade de software possui duas dimensões: processo e produto.

A **qualidade do processo** exige a definição prévia de um processo de desenvolvimento de software que viabilize a sua aplicação.

Nessa dimensão, são aplicados os testes de verificação nos artefatos gerados nas diversas etapas do processo de software, com o objetivo de impedir a propagação de defeitos nas etapas posteriores do referido processo.

A **qualidade do produto** é aplicada ao software na etapa de codificação do processo de software. A partir de um plano de testes, são aplicados os testes de validação ou testes de software, que validam desde a estrutura interna de uma unidade de software até as funcionalidades por parte dos usuários.

ATENÇÃO

Lembre-se de que um sistema de qualidade tem um custo associado à conformidade, ou seja, à obtenção da garantia da qualidade do software, tendo, também, um custo associado à não

conformidade, ou seja, à falta de qualidade do software.

Agora é com você! Vamos verificar se atingimos o objetivo deste módulo? Você está convidado a realizar as atividades a seguir.

VERIFICANDO O APRENDIZADO

1. CONSIDERANDO OS CONCEITOS RELACIONADOS COM A QUALIDADE DE SOFTWARE, PREENCHA AS LACUNAS NAS AFIRMAÇÕES ABAIXO.

- 1) _____ É UM LAPSO HUMANO QUE RESULTA EM UM SOFTWARE INCORRETO.**
- 2) _____ É UMA ANOMALIA NO PRODUTO.**
- 3) _____ OCORRE QUANDO UMA UNIDADE FUNCIONAL DE UM SISTEMA RELACIONADO A UM SOFTWARE NÃO MAIS CONSEGUE DESEMPENHAR AS FUNÇÕES NECESSÁRIAS OU DEIXA DE OPERAR DENTRO DOS LIMITES ESPECIFICADOS.**

AS LACUNAS ESTÃO CORRETA E RESPECTIVAMENTE PREENCHIDAS EM:

- A) Erro - Falha - Defeito**
- B) Erro - Defeito - Falha**
- C) Defeito - Erro - Falha**
- D) Defeito - Falha - Erro**
- E) Falha - Erro - Defeito**

2. (PETROBRAS TRANSPORTE SA –TRANSPETRO – ANALISTA DE SISTEMAS – NEGÓCIOS – CESGRANRIO – 2018). O CUSTO DA QUALIDADE INCLUI TODOS OS CUSTOS FEITOS NA BUSCA DA QUALIDADE, DIVIDINDO-SE EM CUSTOS DE PREVENÇÃO, DE

AVALIAÇÃO E DE FALHA, INTERNA E EXTERNA. ENTRE OS CUSTOS DE PREVENÇÃO ESTÁ O DAS ATIVIDADES DE:

- A) Testes e depuração**
- B) Coleta de dados e métricas de avaliação**
- C) Retrabalho necessárias para corrigir o erro**
- D) Condução de revisões técnicas para os produtos de engenharia de software**
- E) Gerência para planejar e coordenar todas as atividades de controle e garantia de qualidade**

GABARITO

1. Considerando os conceitos relacionados com a qualidade de software, preencha as lacunas nas afirmações abaixo.

- 1) _____ é um lapso humano que resulta em um software incorreto.**
- 2) _____ é uma anomalia no produto.**
- 3) _____ ocorre quando uma unidade funcional de um sistema relacionado a um software não mais consegue desempenhar as funções necessárias ou deixa de operar dentro dos limites especificados.**

As lacunas estão correta e respectivamente preenchidas em:

A alternativa **"B "** está correta.

Um erro é uma diferença detectada entre o resultado de uma computação e o resultado correto ou esperado, ou seja, um problema introduzido no software pelo programador. Defeito é uma manifestação de um erro no software e pode acarretar uma falha. Uma falha é o resultado errado provocado por um defeito ou condição inesperada, causando a suspensão da execução normal do software.

2. (Petrobras Transporte SA –Transpetro – Analista de Sistemas – Negócios – CESGRANRIO – 2018). O custo da qualidade inclui todos os custos feitos na busca da qualidade, dividindo-se em custos de prevenção, de avaliação e de falha, interna e externa. Entre os custos de prevenção está o das atividades de:

A alternativa **"E "** está correta.

Os custos de prevenção são caracterizados pelos esforços que a equipe de qualidade aplica na tentativa de evitar que o processo de desenvolvimento de software ou o software não apresentem defeitos, incluindo os custos relacionados com o planejamento da qualidade.

MÓDULO 2

⦿ Explicar o processo da garantia de qualidade de software

PROCESSO DE GARANTIA DA QUALIDADE

Um sistema de garantia da qualidade de software, também denominado SQA (Software Quality Assurance), pode ser definido como a estrutura organizacional com responsabilidades, procedimentos, processos e recursos para implementação da gestão da qualidade, garantindo que os produtos e serviços de uma determinada organização satisfaçam às expectativas do cliente por meio do atendimento às suas especificações.

O processo de garantia de qualidade de software, parte do referido sistema, inclui a definição e seleção de padrões que devem ser aplicados ao processo de desenvolvimento de software ou ao produto de software, podendo ocorrer a seleção de ferramentas e métodos, como apoio a esses padrões.



Fonte: Song_about_summer/Shutterstock

ATENÇÃO

A equipe de qualidade deve examinar o software sob o ponto de vista do cliente, assertiva alinhada com a máxima de que “A qualidade é conformidade aos requisitos”, ou seja, o software deve atender às necessidades do cliente.

A garantia da qualidade de software é composta pelas atividades que se concentram na gestão da qualidade de software. Vejamos algumas:

PADRÕES

Os padrões, tal como os estabelecidos pela ISO, podem ser adotados pela equipe de Engenharia de Software ou determinados pelo cliente, cabendo à equipe de qualidade garantir que esses padrões sejam aplicados e que todas as entregas estejam em conformidade aos mesmos.

REVISÕES E AUDITORIAS

As revisões técnicas permitem a identificação de erros nos diversos artefatos gerados, sendo as auditorias usualmente aplicadas na verificação da conformidade de processos com os

respectivos procedimentos, tal como a verificação do processo de revisão, a fim de garantir que as revisões estejam sendo eficazes na descoberta de erros.

TESTES

Os testes de software têm como finalidade a descoberta de erros, cabendo à equipe de qualidade garantir que os planos de testes sejam implementados e executados na consecução do citado objetivo básico.

COLETA E ANÁLISE DE ERROS/DEFEITOS

Considerando a máxima de que “Não se controla o que não se mede”, a equipe de qualidade deverá realizar a análise de erros e defeitos, propondo melhorias de procedimentos.

GERENCIAMENTO DE MUDANÇAS

Considerando a alta volatilidade de requisitos e as evoluções tecnológicas constantes, cabe à equipe de qualidade o estabelecimento de procedimentos relacionados com a gestão de mudanças.

EDUCAÇÃO

A equipe de qualidade deverá realizar a gestão de programas educacionais a fim de permitir o aperfeiçoamento da equipe de desenvolvimento.

GERÊNCIA DOS FORNECEDORES

Cabe à equipe de qualidade, no caso de aquisição de software, incorporar exigências de qualidade como parte de qualquer contrato com um fornecedor externo.

ADMINISTRAÇÃO DA SEGURANÇA

Inclui, entre outros, a proteção de dados em todos os níveis, a implantação de firewalls e a garantia de que o software não tenha sido alterado internamente, sem autorização, cabendo à equipe de qualidade o emprego de processos e tecnologias apropriadas.

PROTEÇÃO

O impacto de defeitos ocultos pode ser catastrófico, tal como no nosso exemplo da aeronave com 500 pessoas, cabendo à equipe de qualidade avaliar o impacto de falhas de software e as respectivas tratativas em relação à redução de riscos.

ADMINISTRAÇÃO DE RISCOS

A gestão de riscos é responsabilidade da equipe de desenvolvimento, cabendo à equipe de qualidade garantir que essa gestão seja conduzida de forma apropriada e que planos de

contingência estejam estabelecidos.

PADRÕES DE SOFTWARE

Vamos entender os tipos de padrões de software que podem ser estabelecidos pela equipe de qualidade:

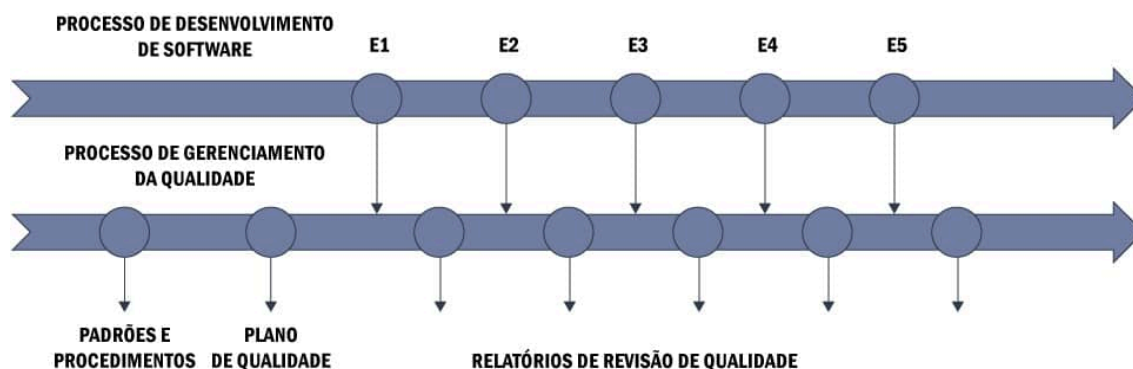


Inclui padrões de documentação, tal como um cabeçalho de comentário padronizado para uma definição de uma classe que permitirá instanciar objetos, e padrões de codificação, que definem como uma linguagem de programação deve ser usada.

PADRÕES DE PROCESSO

Definem os processos que devem ser seguidos durante o desenvolvimento de software, tal como o processo da engenharia de requisitos, e uma descrição dos documentos que devem ser escritos ao longo desses processos.

Você se lembra da Figura 3, apresentada no módulo 1? Reveja:



📷 Figura 3 novamente – Gerenciamento de qualidade versus processo de desenvolvimento.

Fonte: O Autor.

Ela ilustra o relacionamento entre os padrões de produto e de processo. Os padrões de produto aplicam-se à saída do processo de software e, em muitos casos, os padrões de processo incluem atividades de processo específicas que asseguram que os padrões de produto sejam seguidos.

Podemos destacar alguns aspectos relacionados aos padrões de software:

Incluem melhores práticas, evitando a repetição de erros

Provêm um framework para a implementação do processo de garantia de qualidade, assegurando que essas melhores práticas sejam cumpridas

Permitem a continuidade quando da ocorrência de substituições na equipe

A equipe de qualidade deve gerar, podendo ter como base modelos de padrões nacionais e internacionais, os padrões necessários para sua organização. Recomenda-se que esse trabalho inclua o envolvimento da equipe de desenvolvimento, uma sistemática revisão dos padrões e a disponibilização de ferramentas CASE de apoio aos padrões.

A aplicação de um determinado processo de desenvolvimento de software depende fortemente da complexidade do escopo do problema, de modo que poderão ser utilizados diferentes tipos de processos de software em uma organização, impactando o processo de garantia da qualidade. Do exposto, a equipe de desenvolvimento deve ter a autoridade de modificar os padrões de processo de acordo com as circunstâncias de cada projeto em comum acordo com a equipe de qualidade; importante ressaltar a possibilidade de alterações nos padrões existentes, ou até mesmo a inclusão de novos padrões.



Fonte: Jirsak/Shutterstock

PADRÕES ISO 9000

Vejamos agora um exemplo de padrão de processo internacional e vamos entender de forma simplificada como pode ocorrer a sua adoção por parte de uma empresa.

ISO 9000 é um conjunto internacional de padrões aplicados a um sistema de gerenciamento de qualidade genérico, podendo ser aplicado na indústria de manufatura, na indústria de serviço e outros.

O padrão ISO 9001 não é especificamente voltado para desenvolvimento de software, mas possui uma interpretação denominada ISO 9000-3 que estabelece princípios gerais que podem ser aplicados ao software.

As necessidades delineadas pelos tópicos da ISO 9001 são:

Responsabilidade administrativa.

Um sistema de qualidade.

Revisão contratada.

Controle de projeto.

Controle de dados e documentos.

Identificação e rastreabilidade de produtos.

Controle de processos.

Inspeções e testes.

Ações preventivas e corretivas.

Registros de controle de qualidade.

Auditorias de qualidade internas.

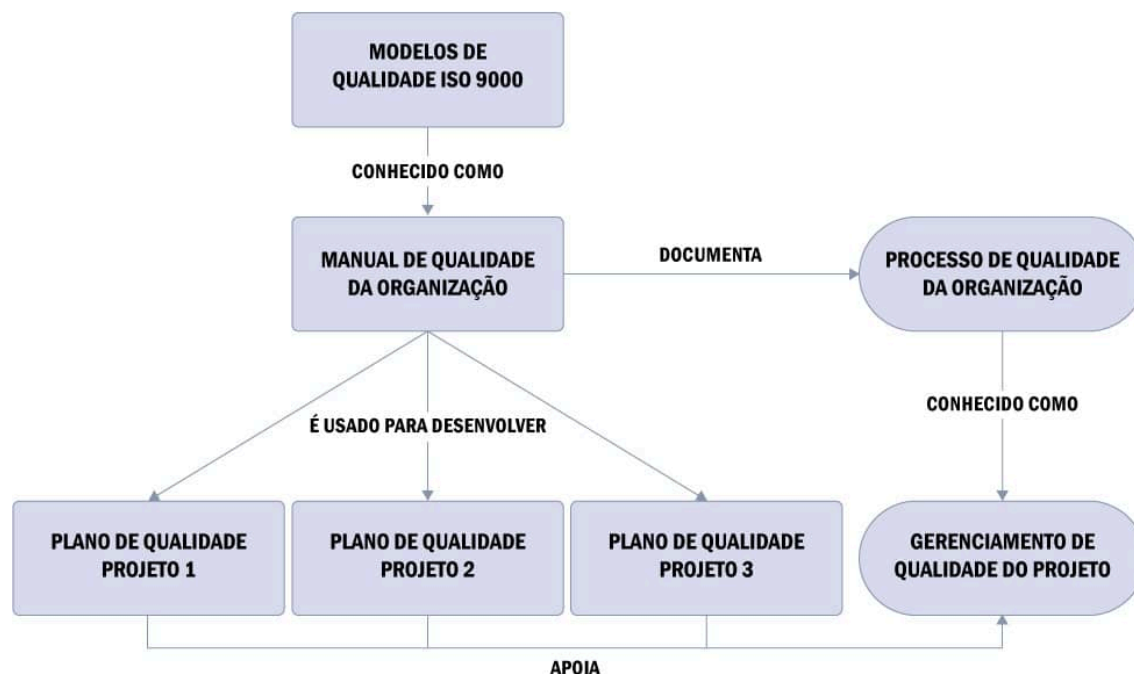
Treinamento.


Manutenção.

ATENÇÃO

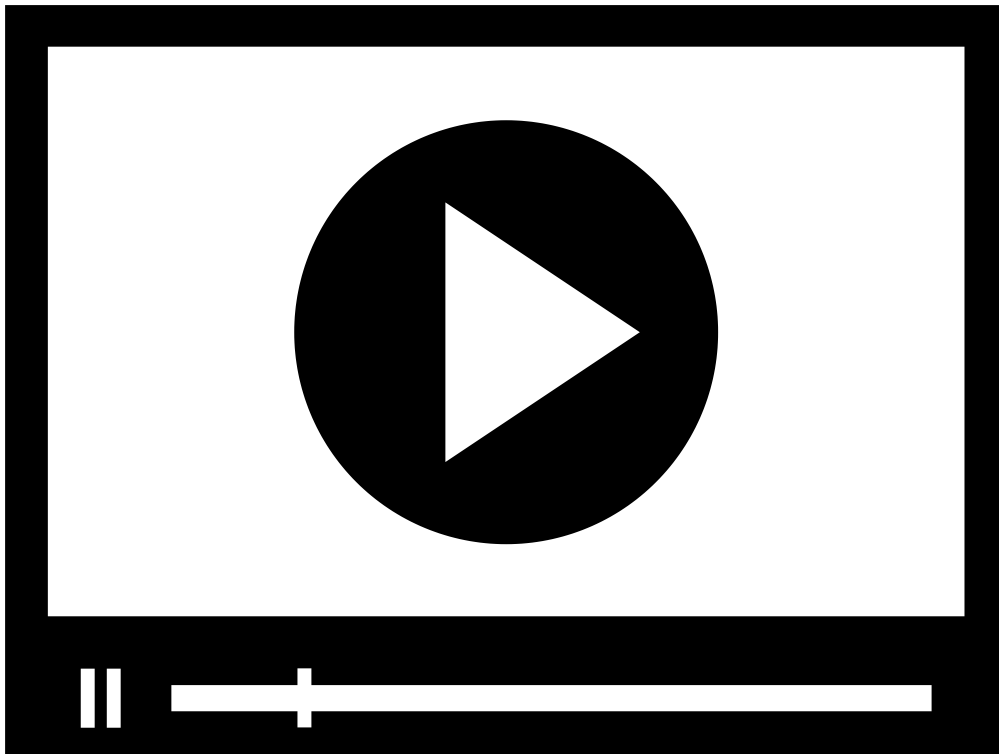
Uma organização de software que busque uma certificação ISO 9001 tem de estabelecer políticas e procedimentos para atender a cada uma das necessidades citadas, bem como ser capaz de demonstrar que tais políticas e procedimentos estão sendo seguidos. A certificação é um indicador da seriedade de como uma organização trata a qualidade.

Os relacionamentos entre o padrão ISO 9000, o manual de qualidade e os planos de qualidade de projeto individual, de acordo com Sommerville (2019), são mostrados na Figura 6. O **manual de qualidade**, ou Plano SQA, aplica os elementos básicos do padrão ISO 9000. Cada projeto instancia um plano de qualidade a partir do referido manual e de acordo com o processo de desenvolvimento de software adotado. O manual determina o processo de qualidade da organização, que devidamente adaptado a cada novo projeto, permite a implantação do gerenciamento de qualidade de cada projeto instanciado.



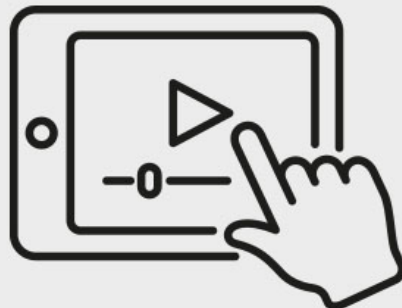
 Figura 6 – ISO 9000 versus gerenciamento da qualidade.

Fonte: O Autor.



Agora, assista ao vídeo sobre o padrão ISO 9000 e sua aplicabilidade à qualidade de software:

Para assistir a um vídeo sobre o assunto, acesse a versão online deste conteúdo.



PADRÃO ISO 9126

Vejamos agora um exemplo de padrão de produto disponível para adoção em um projeto de software. O padrão ISO 9126 foi desenvolvido como uma tentativa de identificar os atributos fundamentais de qualidade para software. O padrão identifica seis atributos fundamentais de qualidade:

FUNCIONALIDADE

O grau com que o software satisfaz às necessidades declaradas conforme indicado pelos subatributos adequabilidade, exatidão, interoperabilidade, conformidade e segurança.

CONFIABILIDADE

Quantidade de tempo que o software fica disponível para uso conforme indicado pelos subatributos maturidade, tolerância a falhas, facilidade de recuperação.

USABILIDADE

Grau de facilidade de utilização do software conforme indicado pelos subatributos facilidade de compreensão, facilidade de aprendizagem, operabilidade.

EFICIÊNCIA

O grau de otimização do uso, pelo software, dos recursos do sistema conforme indicado pelos subatributos comportamento em relação ao tempo, comportamento em relação aos recursos.

FACILIDADE DE MANUTENÇÃO

A facilidade com a qual uma correção pode ser realizada no software conforme indicado pelos subatributos facilidade de análise, facilidade de realização de mudanças, estabilidade, testabilidade.

PORTABILIDADE

A facilidade com a qual um software pode ser transposto de um ambiente a outro conforme indicado pelos subatributos adaptabilidade, facilidade de instalação, conformidade, facilidade de substituição.

GERENCIAMENTO DA QUALIDADE DE ACORDO COM O *PMBOK*

Vamos agora entender o gerenciamento da qualidade segundo o ***PMBOK*** (2017).

Uma das dez áreas de conhecimento do *PMBOK* é o Gerenciamento da Qualidade do Projeto que inclui os processos para incorporação da política de qualidade da organização com relação ao planejamento, gerenciamento e controle dos requisitos de qualidade do projeto e do produto para atender às expectativas dos clientes. A referida área possui três processos que serão descritos a seguir:

PMBOK

O guia **Project Management Body of Knowledge (PMBOK)** é um conjunto de práticas na gestão de projetos organizado pelo instituto PMI e é considerado a base do conhecimento sobre gestão de projetos por profissionais da área.

Planejamento	Execução	Monitoramento e Controle
Gerenciar a Qualidade	Controlar a Qualidade	Planejar o Gerenciamento da Qualidade
Processo do grupo de processo planejamento que permite identificar os requisitos e/ou padrões da qualidade do projeto e suas entregas, e documentar como o projeto demonstrará a conformidade com os requisitos e/ou padrões de qualidade, tendo como principal entrega o plano de gerenciamento da qualidade.	Processo do grupo de processo execução que transforma o plano de gerenciamento da qualidade em atividades da qualidade executáveis que incorporam no projeto as políticas de qualidade da organização.	Processo do grupo de processo planejamento que monitora e registra resultados da execução de atividades de gerenciamento da qualidade para avaliar o desempenho e garantir que as saídas do projeto sejam completas, corretas e atendam às expectativas do cliente.

❑ **Atenção!** Para visualizaçãocompleta da tabela utilize a rolagem horizontal

Tabela 1 – Processos do Gerenciamento da Qualidade do Projeto.

ATENÇÃO

O Gerenciamento da Qualidade do Projeto trata do gerenciamento do projeto e das entregas do projeto, adotando, entre outras, a premissa de que a prevenção é preferível à inspeção, pois é melhor obter a qualidade nas entregas, em vez de identificar não conformidades em uma determinada inspeção. Nesse caso, o custo de prevenção é geralmente muito menor do que o custo de corrigir os erros identificados.

Existem cinco níveis de gerenciamento da qualidade cada vez mais eficaz, conforme a seguir:

1. Deixar que o cliente encontre os defeitos é uma abordagem que pode gerar problemas de garantia, recalls, retrabalho e outros, sendo em geral uma abordagem mais cara.
2. Detectar e corrigir os defeitos antes que as entregas sejam enviadas para o cliente como parte do processo de controlar a qualidade, pois esse processo tem custos relacionados, que são principalmente os custos de avaliação e os custos internos de falhas.
3. Usar a garantia da qualidade para examinar e corrigir o processo em si e não apenas defeitos especiais.
4. Incorporar a qualidade no planejamento e design do projeto e do produto.
5. Criar uma cultura na organização que esteja ciente e comprometida com a qualidade em processos e produtos.

RESUMINDO

Neste módulo, avaliamos a importância da garantia da qualidade no desenvolvimento de software.

Inicialmente, destacamos que um sistema de garantia da qualidade define a estrutura organizacional responsável pela gestão da qualidade, garantindo o atendimento às expectativas do cliente.

Vimos que o processo de garantia de qualidade de software, parte do referido sistema, enfatiza a adoção de padrões que devem ser aplicados ao processo de desenvolvimento de software, bem como garantir que esses sejam cumpridos; os padrões devem ser aplicados nos processos e produtos.

A adoção de um padrão, tal como os especificados pela ISO, permite a geração de manuais de qualidade da organização, que por sua vez define o processo de qualidade da organização, sendo o mesmo customizado para cada projeto em função de sua complexidade e processo de desenvolvimento de software adotado.

Vimos, também, que o *PMBOK* possui três processos na área de conhecimento e gerenciamento da qualidade de projetos em diferentes grupos de processos.

Agora é com você! Vamos verificar se atingimos o objetivo deste módulo? Você está convidado a realizar as atividades a seguir.

VERIFICANDO O APRENDIZADO

1. (DEFENSORIA PÚBLICA DO ESTADO DO RIO DE JANEIRO – DPE-RJ – ANALISTA – TECNOLOGIA DA INFORMAÇÃO – FGV-2019) A EMPRESA “ARMAZÉNS DO JOÃO”, COM O PROPÓSITO DE ADQUIRIR UM SOFTWARE DE CONTROLE DE ESTOQUE, SOLICITOU A UM ANALISTA DE SUA EQUIPE DE INFORMÁTICA QUE VERIFICASSE A QUALIDADE DO SOFTWARE. A AVALIAÇÃO CONSTATOU QUE O SOFTWARE NÃO POSSUÍA DOCUMENTAÇÃO TÉCNICA, NÃO HAVIA COMENTÁRIOS NO CÓDIGO, E SUAS CLASSES E MÉTODOS POSSUÍAM NOMES POUCO SIGNIFICATIVOS. ALÉM DISSO, O SOFTWARE NÃO GARANTIA O ACESSO RESTRITO A INFORMAÇÕES CONFIDENCIAIS DE FORMA CONSISTENTE.

COM BASE NESSAS INFORMAÇÕES, O SOFTWARE NÃO FOI CONSIDERADO DE QUALIDADE, POIS NÃO ATENDIA ÀS CARACTERÍSTICAS DESEJÁVEIS PARA:

- A) Manutenibilidade e confiabilidade**
- B) Confiabilidade e usabilidade**
- C) Usabilidade e segurança**
- D) Manutenibilidade e segurança**

2. (FUNDAÇÃO PAPA JOÃO XXIII – FUNPAPA – ANALISTA DE SISTEMAS – AOCP – 2018) O GERENCIAMENTO DE QUALIDADE DE SOFTWARE PARA SISTEMAS DE SOFTWARE COMPREENDE MELHORIAS SIGNIFICATIVAS NO NÍVEL ORGANIZACIONAL E DE PROJETO. COM RELAÇÃO AOS CONCEITOS DE QUALIDADE DE SOFTWARE, É CORRETO AFIRMAR QUE:

A) No nível de projeto, o gerenciamento de qualidade está preocupado com o estabelecimento de um framework de processos organizacionais e padrões que levem a softwares de alta qualidade. Isso significa que a equipe de gerenciamento de qualidade deve assumir a responsabilidade de definir os processos de desenvolvimento do software que serão usados e os padrões que devem ser usados no software, bem como a documentação relacionada, incluindo os requisitos de sistema, projeto e código.

B) No nível de projeto, o gerenciamento de qualidade envolve a aplicação de processos específicos de codificação, verificando se os processos planejados foram seguidos, e a garantia de que as saídas de projeto estejam em conformidade com os padrões aplicáveis ao projeto.

C) No nível organizacional, o gerenciamento de qualidade está preocupado com o estabelecimento de um plano de qualidade. O plano de qualidade deve definir as metas de qualidade para o projeto e quais processos e padrões devem ser usados.

D) No nível organizacional, o gerenciamento de qualidade envolve a aplicação de processos específicos de qualidade, verificando se os processos planejados foram seguidos, e a garantia de que as saídas de projeto estejam em conformidade com os padrões aplicáveis ao projeto.

E) No nível organizacional, o gerenciamento de qualidade está preocupado com o estabelecimento de um framework de processos organizacionais e padrões que levem a softwares de alta qualidade. Isso significa que a equipe de gerenciamento de qualidade deve assumir a responsabilidade de definir os processos de desenvolvimento do software que serão usados e os padrões que devem ser usados no software, bem como a documentação relacionada, incluindo os requisitos de sistema, projeto e código.

GABARITO

1. (Defensoria Pública do Estado do Rio de Janeiro – DPE-RJ – Analista – Tecnologia da Informação – FGV-2019) A empresa “Armazéns do João”, com o propósito de adquirir um software de controle de estoque, solicitou a um analista de sua equipe de informática que verificasse a qualidade do software. A avaliação constatou que o software não possuía documentação técnica, não havia comentários no código, e suas classes e métodos possuíam nomes pouco significativos. Além disso, o software não garantia o acesso restrito a informações confidenciais de forma consistente.

Com base nessas informações, o software não foi considerado de qualidade, pois não atendia às características desejáveis para:

A alternativa **"D "** está correta.

A manutenibilidade é um aspecto da qualidade de software que se refere à facilidade de um software de ser modificado a fim de corrigir defeitos, adequar-se a novos requisitos, aumentar a suportabilidade ou se adequar a um ambiente novo. O desenvolvimento de software seguro exige o cumprimento das necessidades de segurança do cliente.

2. (Fundação Papa João XXIII – FUNPAPA – Analista de Sistemas – AOCP – 2018) O gerenciamento de qualidade de software para sistemas de software compreende melhorias significativas no nível organizacional e de projeto. Com relação aos conceitos de qualidade de software, é correto afirmar que:

A alternativa **"E "** está correta.

De acordo com Sommerville (2011), o gerenciamento de qualidade de software para sistemas de software tem três principais preocupações: 1. No nível organizacional, o gerenciamento de qualidade está preocupado com o estabelecimento de um framework de processos organizacionais e padrões que levem a softwares de alta qualidade. Isso significa que a equipe de gerenciamento de qualidade deve assumir a responsabilidade de definir os processos de desenvolvimento do software que serão usados e os padrões que devem ser usados no software, bem como a documentação relacionada, incluindo os requisitos de sistema, projeto e código. 2. No nível de projeto, o gerenciamento de qualidade envolve a aplicação de processos específicos de qualidade, verificando que os processos planejados foram seguidos, e a garantia de que as saídas de projeto estejam em conformidade com os padrões aplicáveis ao

projeto. 3. No nível de projeto, o gerenciamento de qualidade também está preocupado com o estabelecimento de um plano de qualidade. O plano de qualidade deve definir as metas de qualidade para o projeto e quais processos e padrões devem ser usados.

MÓDULO 3

- ⦿ **Explicar o planejamento da qualidade e o controle da qualidade de software**

PLANEJAMENTO DA QUALIDADE

A equipe de SQA, ou equipe de garantia da qualidade, tem como atribuição auxiliar a equipe de software a obter um produto final de alta qualidade.

O **SEI** recomenda um conjunto de ações de garantia da qualidade que tratam do planejamento, da supervisão, da manutenção de registros, da análise e de relatórios relativos à referida garantia.

SEI

O **Software Engineering Institute** (SEI) é um centro de pesquisa e desenvolvimento patrocinado pelo Departamento de Defesa dos Estados Unidos que provê uma prática avançada de engenharia de software qualificando graus de qualidade de software.

💬 COMENTÁRIO

Essas ações deverão ser realizadas pela equipe de qualidade, sempre que possível independente, que, entre outras atividades, prepara um plano de qualidade de software para

um projeto. O plano é desenvolvido como parte do planejamento de projeto, sendo revisado por todos os interessados.

As ações de garantia da qualidade realizadas pela equipe de Engenharia de Software e equipe de qualidade são orientadas pelo plano, que identifica as avaliações, auditorias e revisões a serem realizadas, padrões que serão aplicados no projeto, procedimentos para relatório e acompanhamento de erros, produtos resultantes produzidos pelo grupo de SQA e o feedback que será fornecido à equipe de software.

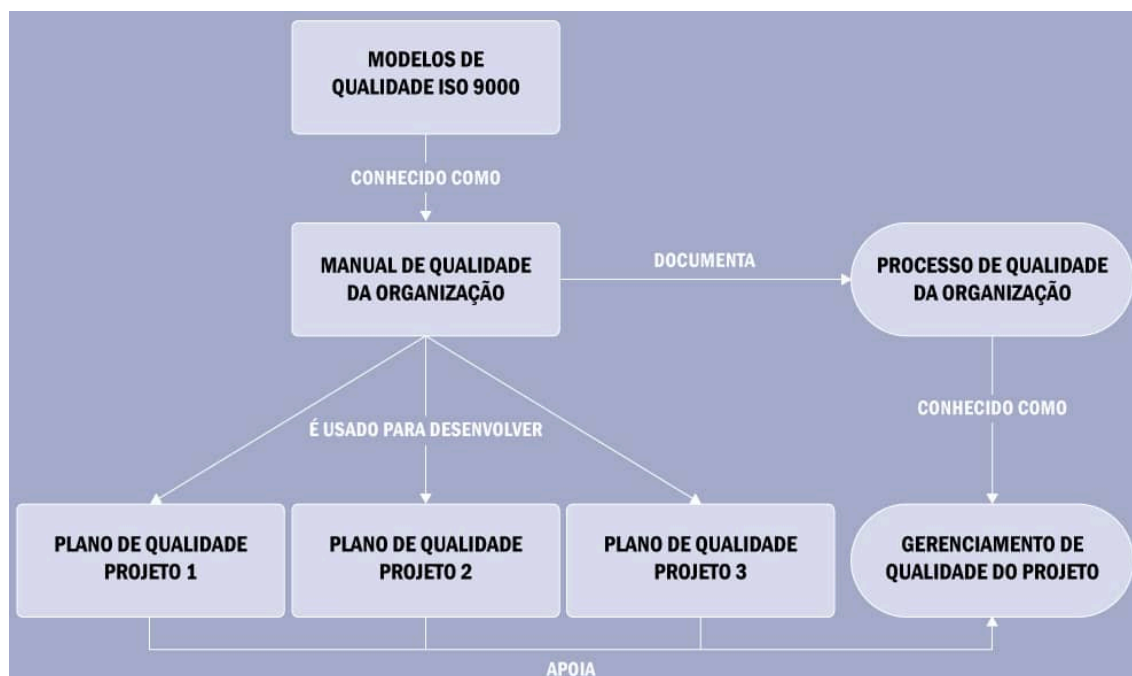
O plano de qualidade de software fornece um roteiro para instituir a garantia da qualidade de software, devendo ser desenvolvido pela equipe SQA ou pela equipe de desenvolvimento, no caso da inexistência da equipe SQA. O plano serve como um gabarito para atividades de garantia da qualidade de software que serão instituídas para cada projeto de software.

Imagino um questionamento: Cada projeto de software tem um plano?

RESPOSTA

Respondemos ao questionamento nos reportando à Figura 6, na qual podemos identificar que a organização elabora um manual de qualidade, ou uma política com procedimentos padrões, sendo instanciado para cada projeto, a partir desse manual, um plano de qualidade de software primariamente em função da complexidade e do processo de desenvolvimento do software em questão.

Reveja:



📷 Figura 6 novamente – ISO 9000 versus gerenciamento da qualidade.

Fonte: O Autor.

Foi publicado pelo **IEEE** um padrão para planos de qualidade de software. O padrão recomenda uma estrutura que identifique:

O propósito e o escopo do plano.

Uma descrição de todos os artefatos resultantes de engenharia de software (por exemplo, modelos, documentos, código-fonte).

Todos os padrões e práticas que são aplicados durante a gestão de qualidade.

As ações e tarefas da garantia da qualidade (incluindo revisões e auditorias) e sua aplicação na gestão de qualidade.

As ferramentas e os métodos que dão suporte às ações e tarefas da garantia da qualidade.

Procedimentos para administração de configurações de software.

Métodos para montagem, salvaguarda e manutenção de todos os registros relativos à garantia da qualidade.

Papéis e responsabilidades dentro da organização relacionados com a qualidade do software.

IEEE

O Instituto de Engenheiros Eletricistas e Eletrônicos é a maior organização profissional do mundo dedicada ao avanço da tecnologia em benefício da humanidade. O IEEE tem filiais em muitas partes do mundo, sendo seus sócios engenheiros eletricistas, engenheiros da computação, cientistas da computação, profissionais de telecomunicações etc. Sua meta é promover conhecimento no campo da Engenharia Elétrica, Eletrônica e Computação, entre outros.



Fonte: Blue Planet Studio/Shutterstock

PLANEJAMENTO DA QUALIDADE

SEGUNDO O *PMBOK*

O *PMBOK* demonstra a importância da qualidade em todos os projetos por meio da definição de uma área de conhecimento denominada de Gerenciamento da Qualidade do Projeto, sendo essa área composta de três processos em diferentes grupos de processos (Tabela 1), ou etapas do ciclo de vida, sendo o processo Planejar o Gerenciamento da Qualidade do grupo de processo planejamento; esse processo permite gerar o Plano de Gestão de Qualidade do Projeto como um todo.

O processo Planejar o Gerenciamento da Qualidade, ilustrado na Tabela 2, é o processo de identificação dos requisitos e/ou padrões de qualidade do projeto e suas entregas, e de documentação de como o projeto demonstrará conformidade com os requisitos e/ou padrões de qualidade.

O principal benefício desse processo é o fornecimento de orientação e direcionamento sobre como a qualidade será gerenciada e verificada ao longo de todo o projeto.

De acordo com a Tabela 2, as entradas são convertidas em saídas pela aplicação das técnicas e ferramentas listadas, cabendo destaque à geração do Plano de Gerenciamento da Qualidade.

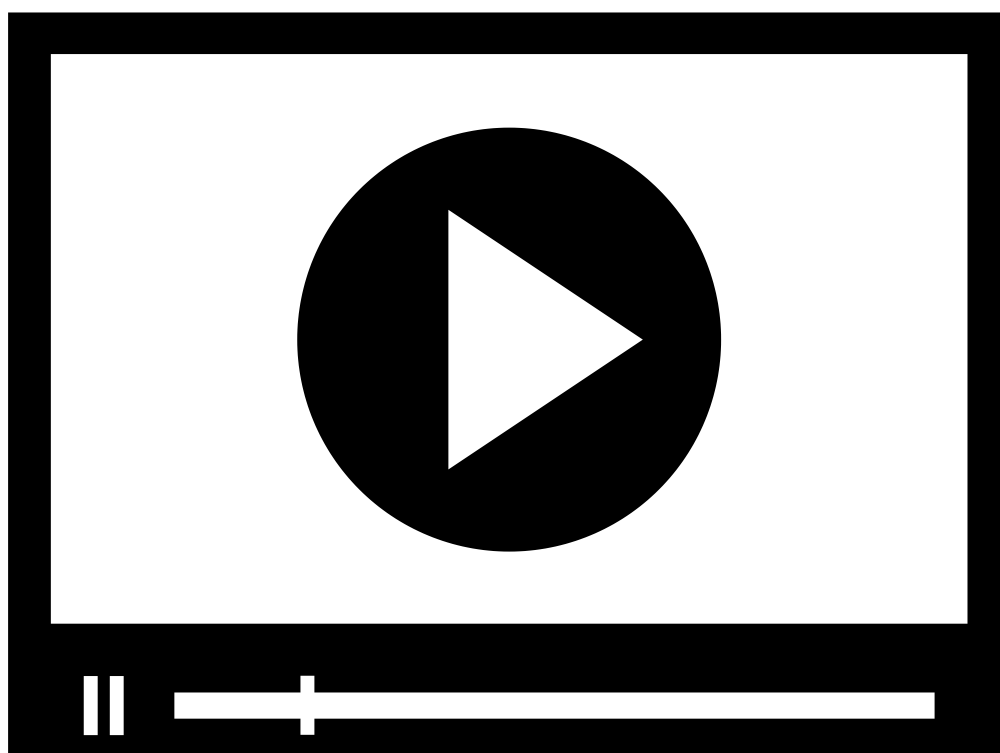
Entradas	Técnicas e Ferramentas	Saídas
Termo de abertura do projeto	Opinião especializada	Plano de gerenciamento da qualidade
Plano de gerenciamento do projeto	Coleta de dados	Métricas da qualidade
Documentos do projeto	Análise de dados	Atualizações do plano de gerenciamento do projeto

Fatores ambientais da empresa	Tomada de decisões	Atualizações de documentos do projeto
Ativos de processos organizacionais	Representação de dados	
	Planejamento de testes e inspeções	
	Reuniões	

☐ **Atenção!** Para visualizaçãocompleta da tabela utilize a rolagem horizontal

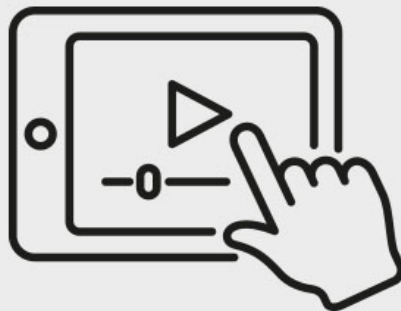
Tabela 2 – Planejar o Gerenciamento da Qualidade.

Fonte: O Autor.



Assista ao vídeo abaixo e conheça a importância dos diversos tipos de testes na garantia da qualidade de software.

Para assistir a um vídeo sobre o assunto, acesse a versão online deste conteúdo.



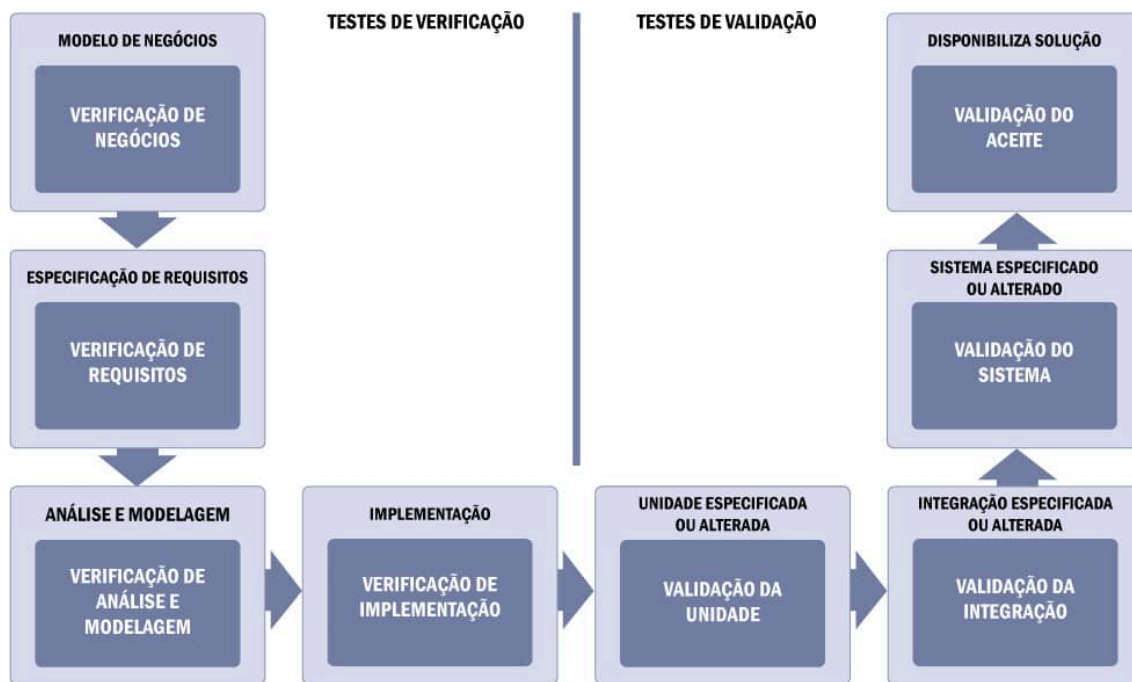
PLANEJAMENTO DE TESTES E REVISÕES

RELEMBRANDO

Lembramos que teste é um processo sistemático e planejado que tem por finalidade única a identificação de erros.

Vamos apresentar uma visão prática do planejamento de testes e revisões, item 6 do módulo “Técnicas e Ferramentas” da Tabela 2, que permite o planejamento da garantia da qualidade do processo e do produto, segundo Bartié (2002). Podemos observar na Figura 7 que o Processo de Qualidade de Software proposto está decomposto em fases que se organizam em um formato de "U".

Na Figura 7, as etapas à esquerda estão relacionadas com a qualidade do processo de desenvolvimento de software, sendo aplicados os denominados testes de verificação. As etapas à direita estão relacionadas com a qualidade do produto software, sendo aplicados os denominados testes de validação.



📷 Figura 7 – Processo de Garantia da Qualidade em “U”.

Fonte: O Autor.

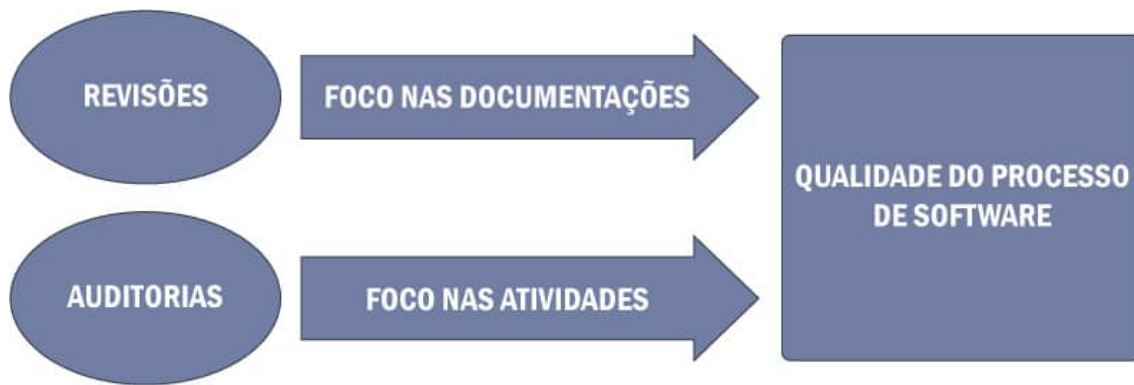
O objetivo de ambos os grupos de testes é garantir que durante o ciclo de desenvolvimento do software se produza efetivamente todos os produtos previstos na metodologia empregada e que o software em construção atenda aos requisitos.

Os testes de verificação visam a garantir o processo de engenharia do software, enquanto os testes de validação estão focados na garantia da qualidade do produto de software, cabendo destacar que ambos os testes se complementam.

TESTES DE VERIFICAÇÃO

Avaliam a qualidade de todo o processo de desenvolvimento de software, validando se todas as documentações e atividades geradas estão de acordo com os padrões estabelecidos, reduzindo os riscos de propagação de falhas ao longo do processo.

Os testes de verificação devem concentrar-se em dois aspectos bem distintos, de acordo com a Figura 8: As atividades de verificação com foco nas documentações são denominadas de revisões; e as que avaliam as atividades, são chamadas de auditorias.



📷 Figura 8 – Métodos Estruturados de Verificação.

Fonte: O Autor.

As duas situações são consideradas métodos estruturados, pois possuem planejamento, regras de condução e necessitam de profissionais qualificados para desempenhar essas funções e acompanhar seu progresso. A revisão é um processo de análise de determinado documento. O principal objetivo das auditorias de qualidade é avaliar se em determinado projeto a equipe de desenvolvimento segue o processo de desenvolvimento de software, ou seja, registrando os defeitos encontrados, produzindo os documentos de análise e projeto etc.

TESTES DE VALIDAÇÃO

Observando a Figura 7 novamente, iniciamos os testes de validação que focam na garantia do produto software. Os procedimentos de testes aplicados diretamente em softwares são também conhecidos como testes de software ou testes dinâmicos, podendo, em sua maioria, sofrer um alto nível de automação, o que possibilita a criação de complexos ambientes de testes que simulam diversos cenários de utilização.

A validação da unidade de software tem como objetivo executar o software de forma a exercitar adequadamente toda a estrutura interna de um componente, como os desvios condicionais, os laços de processamento etc.

**UNIDADE ESPECIFICADA
OU ALTERADA**

**VALIDAÇÃO DA
UNIDADE**

**INTEGRAÇÃO ESPECIFICADA
OU ALTERADA**

**VALIDAÇÃO DA
INTEGRAÇÃO**

A validação da integração objetiva a manutenção de compatibilidade das novas unidades construídas, ou modificadas, com os componentes previamente existentes.

A validação do sistema conta com uma infraestrutura mais complexa de hardware, visando a estar o mais próximo do ambiente real de produção, sendo realizados os testes funcionais, que visam a verificar se a versão corrente do sistema permite executar processos ou casos de uso completos do ponto de vista do usuário, de modo a obter os resultados esperados, e os testes não funcionais, que visam a testar o software em relação às características não funcionais, como desempenho, segurança, confidencialidade, recuperação de falhas etc.

**SISTEMA ESPECIFICADO
OU ALTERADO**

**VALIDAÇÃO DO
SISTEMA**

DISPONIBILIZA SOLUÇÃO

**VALIDAÇÃO DO
ACEITE**

Como essa é a última oportunidade efetiva de detectar falhas no software, poderemos empregar o aceite como uma estratégia para reduzir riscos de uma implantação na qual um erro vital não detectado pode comprometer a imagem da solução como um todo.

Podemos dividir o aceite em três momentos distintos:

TESTE ALPHA

Os usuários finais são convidados a operar o software dentro de um ambiente controlado, sendo realizado na instalação do desenvolvedor.

TESTE BETA

Os usuários finais são convidados a operar o produto utilizando suas próprias instalações físicas, ou seja, o software é testado em um ambiente não controlado pelo desenvolvedor.

ACEITE FORMAL

É uma variação do Teste Beta, cabendo aos próprios clientes e usuários determinarem o que deverá ser testado e validar se os requisitos foram adequadamente implementados.

TESTE DE SISTEMA

? VOCÊ SABIA

De acordo com Pressman (2016), os testes de sistema, não inclusos na Figura 7, extrapolam os limites da engenharia de software, ou seja, consideram o contexto mais amplo da engenharia de sistemas de computadores.

Após ser validado, o software deve ser combinado com outros elementos do sistema, tais como hardware, base de dados etc. Vejamos alguns testes sugeridos nesta etapa:

TESTE DE RECUPERAÇÃO

Tem por objetivo avaliar o comportamento do software após a ocorrência de um erro ou de determinadas condições anormais, tal como simular um saque com defeito no caixa eletrônico ou simular saque com queda de energia.

TESTE DE SEGURANÇA

Busca detectar as falhas de segurança que podem comprometer o sigilo e a fidelidade das informações, bem como provocar perdas de dados ou interrupções de processamento, tal como avaliar se a senha do cartão está sendo requisitada antes e depois da transação ou avaliar se a senha adicional e randômica está sendo requisitada no início da operação.

TESTE POR ESFORÇO

Visa a simular condições atípicas de utilização do software, provocando aumentos e reduções sucessivas de transações que superem os volumes máximos previstos para o software,

gerando contínuas situações de pico e avaliando como o software e toda a infraestrutura estão se comportando, tal como simular 10.000 saques simultâneos.

TESTE DE DESEMPENHO

Objetiva determinar se o desempenho, nas situações previstas de pico máximo de acesso e concorrência, está consistente com os requisitos definidos, tal como garantir que a manipulação com dispositivos físicos no saque não ultrapasse 10 segundos da operação.

TESTE DE DISPONIBILIZAÇÃO

Também chamado de teste de configuração, tem por objetivo executar o software sobre diversas configurações de softwares e hardwares; a ideia é garantir que a solução tecnológica seja executada adequadamente sobre os mais variados ambientes de produção previstos nas fases de levantamento dos requisitos, tal como simular um saque com impressoras dos fornecedores X, Y e Z.



Fonte: Blue Planet Studio/Shutterstock

CONTROLE DE QUALIDADE

Controlar a qualidade é o processo de monitorar e registrar resultados da execução das atividades de gerenciamento da qualidade verificando se as entregas e o trabalho do projeto

cumprem os requisitos especificados. O processo Controlar a Qualidade é realizado para medir a integridade, conformidade e adequação para uso de um produto ou serviço antes da aceitação do usuário e entrega final. Isso é feito medindo todos os passos, atributos e variáveis usados para verificar a conformidade com ou o cumprimento das especificações definidas no planejamento.

Entradas	Técnicas e Ferramentas	Saídas
Plano de gerenciamento do projeto - Plano de gerenciamento da qualidade	Coleta de dados - Listas de verificação - Folhas de verificação - Amostragem estatística - Questionário e pesquisas	Medições de controle da qualidade
Documentos do projeto - Registro das lições aprendidas - Métricas da qualidade - Documentos de teste e avaliação	Análises de dados - Análises de desempenho - Análise de causa-raiz	Entregas verificadas
Solicitações de mudança aprovadas	Inspeção	Informações sobre o desempenho do trabalho
Entregas	Testes/avaliação de produtos	Solicitações de mudança
Dados de desempenho do trabalho	Representação de dados - Diagramas de causa	Atualizações do plano de gerenciamento do projeto

	e efeito - Gráficos de controle - Histograma - Diagramas de dispersão	- Plano de gerenciamento da qualidade
Fatores ambientais da empresa	Reuniões	Atualizações de documentos do projeto - Registro das questões - Registro das lições aprendidas - Registro dos riscos - Documentos de teste e Avaliação
Ativos de processos organizacionais		

☐ **Atenção!** Para visualizaçãocompleta da tabela utilize a rolagem horizontal

Tabela 3 – Controlar a Qualidade.

Fonte: O Autor.

ATENÇÃO

- Devemos destacar que a entrada do processo “Documentos de testes e avaliação”, na coluna Entradas, incluem os testes aplicados na garantia do processo e do produto descritos anteriormente.
- Um segundo destaque, na coluna Saídas, são as “Solicitações de mudança”, pois uma não conformidade observada poderá gerar uma atualização no registro de mudanças do projeto, sendo que as solicitações de mudança aprovadas podem incluir modificações, como reparos de defeitos, revisão dos métodos de trabalho e revisão dos cronogramas.

RESUMINDO

Neste módulo, compreendemos o Planejamento da Qualidade com destaque ao trabalho realizado pela equipe de garantia da qualidade, que tem como atribuição auxiliar a equipe de software a obter um produto final de alta qualidade.

O plano de qualidade de software, elaborado pela referida equipe, fornece um roteiro para instituir a garantia da qualidade de software, servindo como um gabarito para as atividades de garantia da qualidade de software que serão instituídas para cada projeto de software.

Existem padrões para os referidos planos propostos pelo IEEE, *PMBOK*, entre outros.

Vimos também os testes de verificação que estão relacionados com a qualidade do processo de desenvolvimento de software e os testes de validação, relacionados com a qualidade do produto software.

O processo Controlar a qualidade permite atender à máxima “Não se controla o que não se mede”, permitindo monitorar e registrar resultados da execução das atividades de gerenciamento da qualidade, verificando se as entregas e o trabalho do projeto cumprem os requisitos especificados.

Agora é com você! Vamos verificar se atingimos o objetivo deste módulo? Você está convidado a realizar as atividades a seguir.

VERIFICANDO O APRENDIZADO

1. (UNIVERSIDADE FEDERAL DE PERNAMBUCO – UFPE – ANALISTA DE TECNOLOGIA DA INFORMAÇÃO – SISTEMAS – COVEST-COPSET – 2019) O ENGENHEIRO DE SOFTWARE AVALIA QUE NA SUA EQUIPE, EM DATAS PRÓXIMAS DA DATA DE ENTREGA DE UMA VERSÃO DO SISTEMA, A PRODUTIVIDADE E O NÍVEL DE ESTRESSE DA EQUIPE SÃO IMPACTADOS. ELE DESEJA AUTOMATIZAR O PROCESSO, DE FORMA A MITIGAR ESSES EFEITOS. PARA TANTO:

- A)** Como parte da implantação contínua, ele automatiza o processo de forma que impeça que a nova versão do sistema entre em produção, caso o teste falhe.
- B)** Como parte da entrega contínua, ele automatiza uma série de rotinas para que o sistema seja automaticamente posto em produção.
- C)** Como parte da implantação contínua, ele elabora uma série de testes para garantir que, ao implementar uma nova rotina ou funcionalidade, as outras partes do sistema continuem operando normalmente.
- D)** Como parte da integração contínua, ele automatiza o processo de agregar novas mudanças na forma de recursos e funcionalidades, em uma nova versão.
- E)** Como parte da entrega contínua, ele elabora smoke tests para garantir o funcionamento do sistema, antes de enviá-lo para produção.

2. (MINISTÉRIO PÚBLICO DO ESTADO DE ALAGOAS – MPE-AL – ANALISTA – DESENVOLVIMENTO DE SISTEMAS – FGV – 2018) EDUARDO É O LÍDER TÉCNICO DO SISTEMA DE VENDAS DE UMA REDE DE FARMÁCIAS. O SISTEMA DEVE SER UTILIZADO EM MAIS DE 40 UNIDADES ESPALHADAS POR VÁRIOS ESTADOS. O SISTEMA ENTROU EM PRODUÇÃO E, JÁ NA PRIMEIRA SEMANA DE USO, FICOU MUITO LENTO E DIVERSAS VEZES INDISPONÍVEL PARA OS OPERADORES DAS LOJAS. DIANTE DESSE CENÁRIO, ASSINALE A OPÇÃO QUE INDICA A TÉCNICA DE TESTE QUE FOI NEGLIGENCIADA.

- A)** De fumaça
- B)** Funcional de limite
- C)** De desempenho
- D)** Caixa-branca
- E)** De análise de valor-limite

GABARITO

1. (Universidade Federal de Pernambuco – UFPE – Analista de Tecnologia da Informação – Sistemas – COVEST-COPSET – 2019) O engenheiro de software avalia que na sua

equipe, em datas próximas da data de entrega de uma versão do sistema, a produtividade e o nível de estresse da equipe são impactados. Ele deseja automatizar o processo, de forma a mitigar esses efeitos. Para tanto:

A alternativa "A " está correta.

Sempre que ocorre uma falha no teste de software, esse não deverá migrar do ambiente de desenvolvimento para o ambiente de produção.

2. (Ministério Público do Estado de Alagoas – MPE-AL – Analista – Desenvolvimento de Sistemas – FGV – 2018) Eduardo é o líder técnico do sistema de vendas de uma rede de farmácias. O sistema deve ser utilizado em mais de 40 unidades espalhadas por vários estados. O sistema entrou em produção e, já na primeira semana de uso, ficou muito lento e diversas vezes indisponível para os operadores das lojas. Diante desse cenário, assinale a opção que indica a técnica de teste que foi negligenciada.

A alternativa "C " está correta.

Teste de desempenho é similar ao teste de carga, mas com o intuito de testar o software a fim de encontrar o seu limite de processamento de dados no seu melhor desempenho. No teste, normalmente é avaliada a capacidade de resposta em determinados cenários e configurações.

MÓDULO 4

🕒 **Descrever as medições e métricas do software**

MEDIÇÃO

Um elemento-chave de qualquer processo de engenharia é a medição, pois podemos aplicar a máxima “Não se controla o que não se mede”.

Vamos entender o conceito de medição por meio dos termos medida, medição e métricas. No contexto de engenharia de software, uma medida proporciona uma indicação quantitativa da

extensão, quantidade, capacidade ou tamanho de algum atributo de um produto ou processo, sendo a medição ato de determinar uma medida.

A métrica é uma medida quantitativa do grau com o qual um sistema, componente ou processo possui determinado atributo.

★ EXEMPLO

Exemplificando, uma medida é estabelecida quando ocorre a coleta do número de erros descobertos em um componente de software. A medição ocorre como resultado de um conjunto de revisões de componentes e testes de unidade investigados na coleta de medidas do número de erros para cada um. A métrica de software relaciona as medidas individuais, ou seja, o número médio de erros encontrados por revisão ou o número médio de erros encontrados por teste de unidade.

A medição de software está preocupada com a quantificação de algum atributo de um sistema de software, como a sua complexidade ou a sua confiabilidade. Comparando os valores medidos uns com os outros e com os padrões que se aplicam a uma organização, deve ser possível tirar conclusões sobre a qualidade do software ou avaliar a eficácia de processos de software, de ferramentas e de métodos, ou seja, o gerenciamento da qualidade pode contar com medições de atributos que afetam a qualidade do software.

PROCESSO DE MEDIÇÃO

O fluxo abaixo ilustra um processo de medição proposto por Sommerville (2019), que pode ser parte de um processo de avaliação da qualidade de software, no qual cada componente do sistema pode ser analisado separadamente usando uma variedade de métricas. Os valores dessas métricas podem, então, ser comparados com diferentes componentes e, talvez, com dados de medição históricos, coletados em projetos anteriores. Medições anômalas, que se desviam significativamente da norma, em geral indicam problemas com a qualidade desses componentes.

ESCOLHER AS MEDIÇÕES A SEREM FEITAS

Inclui a formulação das questões às quais a medida se destina a responder, bem como a definição das medições necessárias para respondê-las.

SELECIONAR OS COMPONENTES A SEREM AVALIADOS

Selecionar um conjunto representativo de componentes para medição, a fim de permitir uma avaliação geral da qualidade do sistema.

MEDIAR AS CARACTERÍSTICAS DOS COMPONENTES

Os componentes selecionados são medidos, e os valores de métricas associados são calculados, sendo sugerido o uso de uma ferramenta de coleta automática de dados.

IDENTIFICAR MEDIÇÕES ANÔMALAS

Depois que as medições do componente foram feitas, é possível compará-las umas com as outras e com as medições anteriores registradas em um banco de dados de medição, devendo-se procurar valores atipicamente altos ou baixos para cada métrica, pois eles sugerem que pode haver problemas com o componente que está exibindo esses valores.

ANALISAR COMPONENTES ANÔMALOS

Ao identificar componentes que têm valores anômalos, é preciso examiná-los para decidir se esses valores de métricas anormais significam que a qualidade do componente está comprometida ou não, pois um valor de métrica anormal, e.g., para a complexidade, não significa necessariamente um componente de má qualidade, pois pode haver algum outro motivo para o alto valor, ou seja, pode não haver problema de qualidade com o componente.

Uma boa prática inclui a manutenção dos dados coletados como um recurso organizacional e guardar registros históricos de todos os projetos, pois o estabelecimento de um banco de dados de medição robusto permitirá realizar comparações de qualidade de software em projetos e validar as relações entre atributos de componentes internos e características de qualidade.



Fonte: G-Tech Studios/Shutterstock

MÉTODO GQM

O método GQM (**Goal – Question – Metric**) é uma técnica aplicada no planejamento do trabalho de medição, ou seja, permite a identificação de métricas significativas para qualquer parte do processo de software. O GQM organiza o planejamento de uma medição de software em etapas de acordo com as descrições a seguir:

GQM

Goal Question Metric - GQM é uma abordagem de métrica de software em engenharia de software, desenvolvida por Victor Basili, da Universidade de Maryland, e pelo Laboratório de Engenharia de Software do Goddard Space Flight Center da NASA, após orientar uma tese de doutorado do Dr. David M. Weiss. (Wikipedia)

OBJETIVO

QUESTÕES

MÉTRICAS

OBJETIVO

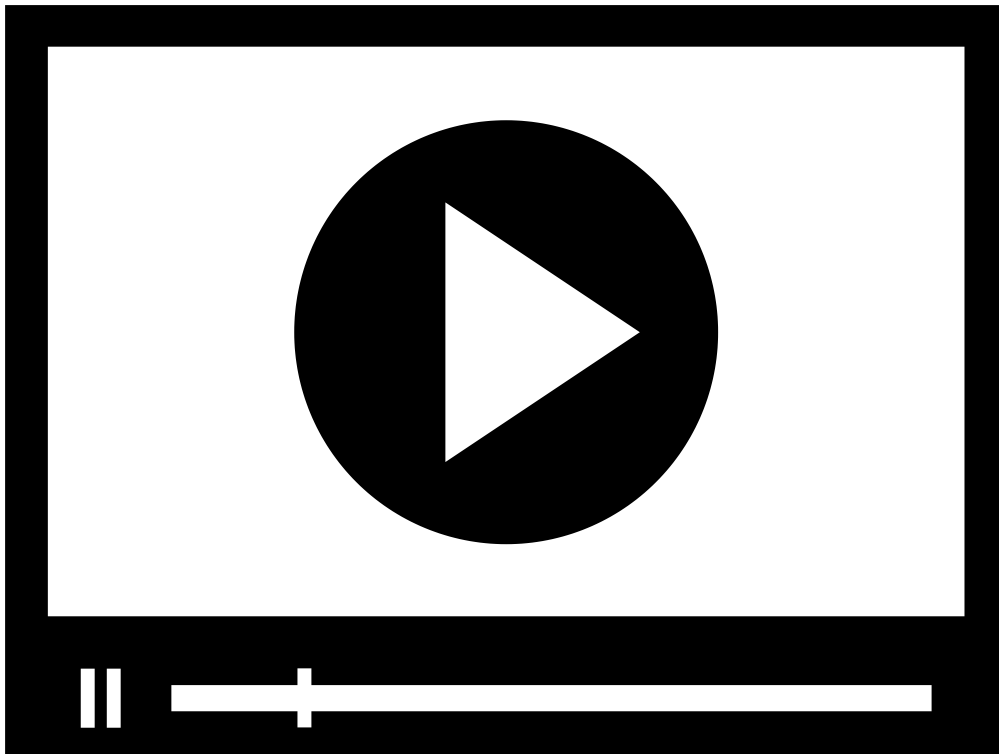
Um objetivo de medição deve ser específico para uma determinada atividade do processo ou característica de produto a ser avaliada e de acordo com os requisitos do software, tal como “Garantir que todos os requisitos funcionais deverão ser testados”.

QUESTÕES

Definição de um conjunto de questões que devem ser respondidas para atingir o objetivo, tal como “Qual a cobertura dos testes?”; as questões estabelecem uma ponte entre os objetivos planejados e as métricas.

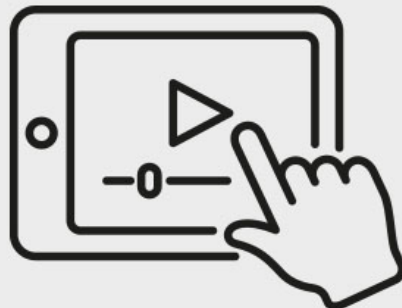
MÉTRICAS

Deverão ser formuladas em função das respostas às questões elaboradas, tal como “Número de requisitos testados”.



Conheça as métricas utilizadas no processo de garantia da qualidade de software:

Para assistir a um vídeo sobre o assunto, acesse a versão online deste conteúdo.



MÉTRICAS

RELEMBRANDO

Como afirmado anteriormente, uma métrica é uma medida quantitativa do grau com o qual um sistema, componente ou processo possui determinado atributo, tal como o número médio de erros encontrados por revisão ou o número médio de erros encontrados por teste de unidade.

A Engenharia de Software é uma área de conhecimento quantitativa, portanto, uma métrica de produto ajuda os engenheiros de software a visualizar o projeto e a construção do software por meio de atributos específicos e mensuráveis permitindo aos mesmos criar um software de alta qualidade.

Vamos entender como são utilizadas as métricas.

No contexto do processo de definição e após a definição das métricas, coletam-se os dados necessários para derivar as métricas formuladas, sendo as mesmas analisadas com base em diretrizes preestabelecidas e dados do passado; os resultados das análises são interpretados para obter informações sobre a qualidade do software e os dados da interpretação podem gerar alterações nos requisitos, modelos de projeto, código-fonte ou processo de software.

+ SAIBA MAIS

Uma melhor prática é descrever um conjunto de atributos a serem aplicados às métricas, tais como: Fácil aprendizado em relação à sua derivação e com uma computação sem demanda de esforço; deverá satisfazer as ideias do engenheiro sobre o atributo do produto considerado (por exemplo, uma métrica que mede coesão de módulo deverá crescer em valor na medida em que aumenta o nível de coesão); sempre produzir resultados que não sejam ambíguos; a computação matemática da métrica deverá usar medidas que não resultem em combinações incomuns de unidades, e.g., multiplicar o número de pessoas nas equipes de projeto pelas variáveis da linguagem de programação no programa resulta em uma mistura duvidosa de unidades que não é claramente convincente; baseadas no modelo de requisitos, modelo de projeto ou na própria estrutura do programa; fornecer informações que podem levar a um produto final de melhor qualidade.

Podemos afirmar que uma métrica de software é uma característica de um sistema de software, da documentação do sistema ou do processo de desenvolvimento, que pode ser medida objetivamente. De acordo com Sommerville (2019), as métricas de software podem ser:

MÉTRICAS DE CONTROLE OU DE PROCESSO

Apoiam o gerenciamento de processos, ou seja, geralmente estão associadas aos processos de software, tais como o esforço médio ou o tempo necessário para reparar defeitos relatados. Podem ser usados três tipos de métricas de processo:

O tempo que um determinado processo leva para ser concluído.

Os recursos necessários para um determinado processo, tais como o esforço total, em pessoas-dia, custos de viagens ou recursos de computadores.

O número de ocorrências de um determinado evento, tais como o número de defeitos detectados durante a inspeção de código, o número de alterações de requisitos solicitadas, o número de relatórios de defeitos em um sistema entregue e o número médio de linhas de código modificadas em resposta a uma mudança de requisitos.

MÉTRICAS DE PREVISÃO OU DE PRODUTO

Também denominadas de métricas de produto, estão associadas ao produto software e permitem quantificar atributos internos de um sistema de software, tais como o tamanho do sistema, medido em linhas de código, ou o número de métodos associados a cada classe.

As métricas de produtos enquadram-se em duas classes:

Métricas dinâmicas, cujas medições são realizadas no software em execução durante um teste de sistema ou em operação.

Métricas estáticas, cujas medições são feitas a partir de representações do sistema, como o projeto, o programa ou a documentação.

As métricas estáticas ajudam a avaliar a complexidade, a compreensibilidade e a manutenibilidade de um sistema ou de seus componentes. A tabela a seguir exemplifica algumas métricas estáticas de produto software.

Métrica	Descrição
Fan-in / fan-out	<p>Fan-in: Número de funções ou métodos que chamam outra função ou método (por exemplo, x).</p> <p>Fan-out: Número de funções que são chamadas pela função x.</p>

	<p>Um valor elevado para o fan-in significa que x está fortemente acoplado ao resto do projeto e as mudanças em x terão um amplo efeito dominó.</p> <p>Um valor elevado para fan-out sugere que a complexidade geral de x pode ser alta por causa da complexidade da lógica de controle necessária para coordenar os componentes chamados.</p>
Comprimento do código	<p>Medida do tamanho de um programa.</p> <p>Geralmente, quanto maior o tamanho do código de um componente, mais complexo e propenso a erros ele tende a ser.</p>
Complexidade ciclomática	<p>Visa a analisar a estrutura do código-fonte.</p>

□ **Atenção!** Para visualizaçãocompleta da tabela utilize a rolagem horizontal

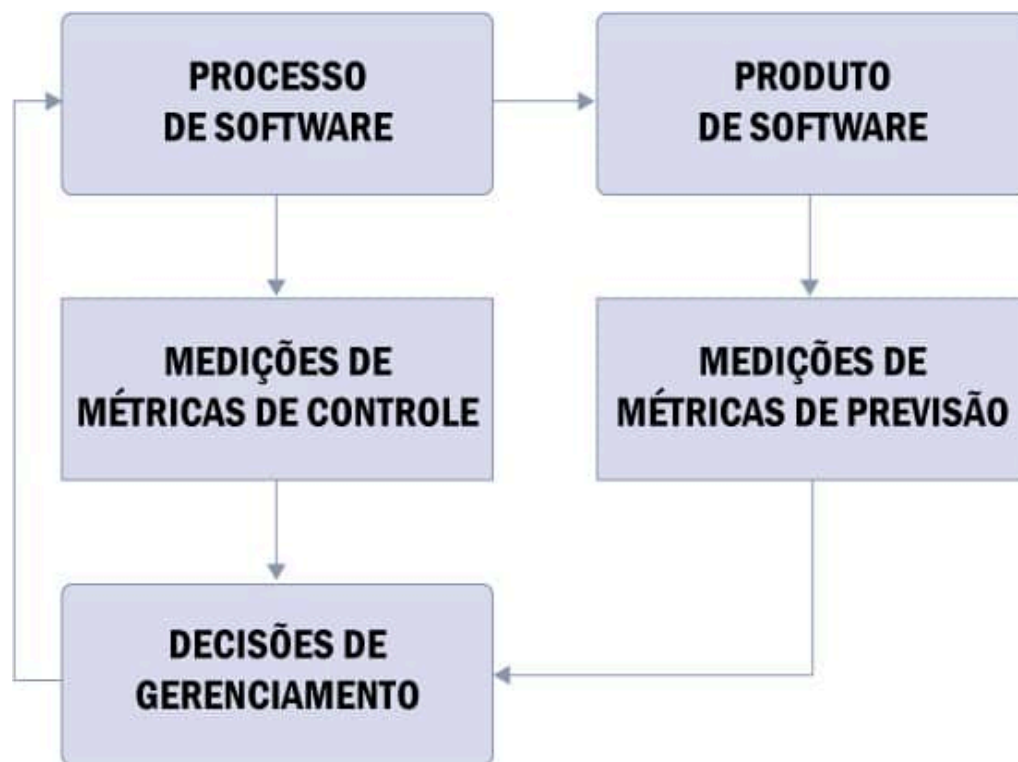
Tabela 4 – Métricas estáticas de produto.

As métricas dinâmicas ajudam a avaliar a eficiência e a confiabilidade de um sistema.

★ EXEMPLO

A confiabilidade é uma característica de qualidade de software a ser avaliada, pois, de forma simplificada, um software é confiável quando não falha (ver Figura 4. Fatores de qualidade de software de McCall). Nesse caso, podemos aplicar algumas métricas dinâmicas: Total de falhas relatadas durante o teste, número de falhas corrigidas, tempo médio entre as falhas, porcentagem de falhas corrigidas, entre outras.

As métricas de controle e de previsão podem influenciar a tomada de decisões de gerenciamento, como mostrado na Figura 9. Os gerentes usam as medições de processo para decidir se as alterações desse processo devem ser feitas e as métricas de previsão para decidir se as alterações de software são necessárias e se o software está pronto para entrar em produção.



📷 Figura 9 – Medições de previsão e controle.

Fonte: O Autor.

RESUMINDO

Neste módulo, destacamos a importância das medições e métricas para a qualidade do software.

A medição de software tem como objetivo a quantificação de algum atributo de um sistema de software, como a sua complexidade ou a sua confiabilidade. Comparando os valores medidos uns com os outros e com os padrões que se aplicam a uma organização, deve ser possível tirar conclusões sobre a qualidade do software ou avaliar a eficácia de processos de software, de ferramentas e de métodos. Um processo de medição pode ser parte de um processo de avaliação da qualidade de software, onde cada componente do sistema pode ser analisado separadamente usando uma variedade de métricas.

A Engenharia de Software é uma área de conhecimento quantitativa, portanto, uma métrica de produto ajuda os engenheiros de software a visualizar o projeto e a construção do software por meio de atributos específicos e mensuráveis permitindo aos mesmos criar um software de alta qualidade.

As métricas de controle geralmente estão associadas aos processos de software e as métricas de previsão associadas ao produto software, permitindo quantificar atributos de um sistema de software.

Agora é com você! Vamos verificar se atingimos o objetivo deste módulo? Você está convidado a realizar as atividades a seguir.

VERIFICANDO O APRENDIZADO

1. MINISTÉRIO PÚBLICO DE CONTAS DO ESTADO DO PARÁ – MPC-PA – ANALISTA – TECNOLOGIA DA INFORMAÇÃO – CESPE – 2019) A MÉTRICA DE QUALIDADE DE CÓDIGO QUE MEDE A COMPLEXIDADE ESTRUTURAL DE UM PROGRAMA COMPUTANDO O NÚMERO DE CAMINHOS LINEARMENTE INDEPENDENTES AO LONGO DO CÓDIGO-FONTE É DENOMINADA:

- A) Complexidade ciclomática**
- B) Complexidade de Halstead**
- C) Contagem de pontos de microfunção ponderados**
- D) Índice de manutenibilidade**
- E) Extensibilidade**

2. (ASSEMBLEIA LEGISLATIVA DO ESTADO DO RIO DE JANEIRO – ALERJ – ANALISTA – TECNOLOGIA DA INFORMAÇÃO– FGV – 2017) UM SISTEMA ESTÁ SENDO DESENVOLVIDO POR UMA EMPRESA TERCEIRIZADA PARA APOIAR AS VENDAS DE UM MERCADO VAREJISTA DA GRANDE

SÃO PAULO DENOMINADO “MENDES SÁ COLÃO”. APÓS O DESENVOLVIMENTO DO SISTEMA, A EMPRESA TERCEIRIZADA DEVERÁ PASSAR O CÓDIGO-FONTE PARA A ÁREA DE TI DA “MENDES SÁ COLÃO”, QUE PASSARÁ A SER RESPONSÁVEL PELA CONTINUIDADE DO SISTEMA. FOI RESSALTADA, TAMBÉM, A NECESSIDADE DE QUE O SISTEMA, CASO OCORRA UMA FALHA, RECUPERE-SE DE FORMA AUTOMÁTICA E RAPIDAMENTE.

NESSE CASO, OS ATRIBUTOS DE QUALIDADE DO SISTEMA COM MAIOR PESO SÃO:

- A) Portabilidade e confiabilidade**
- B) Manutenibilidade e confiabilidade**
- C) Portabilidade e eficiência**
- D) Confiabilidade e usabilidade**
- E) Manutenibilidade e eficiência**

GABARITO

1. Ministério Público de Contas do Estado do Pará – MPC-PA –Analista – Tecnologia da Informação – CESPE – 2019) A métrica de qualidade de código que mede a complexidade estrutural de um programa computando o número de caminhos linearmente independentes ao longo do código-fonte é denominada:

A alternativa **"A "** está correta.

Complexidade ciclomática é uma métrica de software usada para indicar a complexidade de um programa de computador, por meio da medição da quantidade de caminhos de execução independentes a partir de um código-fonte.

2. (Assembleia Legislativa do Estado do Rio de Janeiro – ALERJ – Analista – Tecnologia da Informação– FGV – 2017) Um sistema está sendo desenvolvido por uma empresa terceirizada para apoiar as vendas de um mercado varejista da Grande São Paulo denominado “Mendes Sá Colão”. Após o desenvolvimento do sistema, a empresa terceirizada deverá passar o código-fonte para a área de TI da “Mendes Sá Colão”, que

passará a ser responsável pela continuidade do sistema. Foi ressaltada, também, a necessidade de que o sistema, caso ocorra uma falha, recupere-se de forma automática e rapidamente.

Nesse caso, os atributos de qualidade do sistema com maior peso são:

A alternativa **"B "** está correta.

A medição atributo da manutenibilidade irá possibilitar reduzir o esforço na alteração do software após o recebimento do código-fonte do mesmo e a medição da confiabilidade permitirá uma rápida resposta a uma falha, caso ocorra.

CONCLUSÃO

CONSIDERAÇÕES FINAIS

A qualidade no contexto da Engenharia de Software possui duas dimensões: Processo e produto. A produção de software com qualidade exige a garantia da qualidade em todas as etapas do processo de desenvolvimento do software, bem como do produto software.

Destacamos que um sistema de garantia da qualidade define a estrutura organizacional responsável pela gestão da qualidade, garantindo o atendimento às expectativas do cliente. O processo de garantia de qualidade de software, parte do referido sistema, enfatiza a adoção de padrões que devem ser aplicados ao processo de desenvolvimento de software.

O plano de qualidade de software, elaborado pela equipe de garantia da qualidade, fornece um roteiro para instituir a garantia da qualidade de software, servindo como um gabarito para atividades de garantia da qualidade de software que são instituídas para cada projeto de software.

A medição de software tem como objetivo a quantificação de algum atributo de um sistema de software, como a sua complexidade ou a sua confiabilidade. Comparando os valores medidos uns com os outros e com os padrões que se aplicam a uma organização, deve ser possível tirar

conclusões sobre a qualidade do software ou avaliar a eficácia de processos de software, de ferramentas e de métodos. Um processo de medição pode ser parte de um processo de avaliação da qualidade de software.

No contexto da medição de software, as métricas ajudam os engenheiros de software a visualizar o projeto e a construção do software por meio de atributos específicos e mensuráveis, permitindo aos mesmos criar um software de alta qualidade.

Para ouvir um *podcast* sobre o assunto, acesse a versão online deste conteúdo.



REFERÊNCIAS

BARTIÉ, A. **Garantia da Qualidade de Software**. 1. ed. Rio de Janeiro: Elsevier, 2002.

DIJKSTRA, E. W. **The Humble Programmer**: Turing Award Lecture, Comm. *In*: ACM, v. 15, n. 10, p. 859-866, out. 1972.

PRESSMAN, R. S.; MAXIM, B. R. **Engenharia de Software**. 8. ed. Porto Alegre: AMGH, 2016.

SOMMERVILLE, I. **Engenharia de Software**. 10. ed. São Paulo: Pearson Prentice Hall, 2019.

EXPLORE+

Para saber mais sobre os assuntos tratados neste tema, leia:

Engenharia de Software, Pressman, capítulos 19, 21 e 30.

Engenharia de Software, Sommerville, capítulo 24.

Garantia da Qualidade de Software, Bartié, capítulo 4.

CONTEUDISTA

Alberto Tavares da Silva

 **CURRÍCULO LATTES**