



QUALIDADE DE SOFTWARE

AULA 3



Prof.^a Maristela Weinfurter



CONVERSA INICIAL

Modelos e técnicas são ferramentas importantes para a gestão da garantia da qualidade. E aqui falaremos de modelos de processos de desenvolvimento de software e modelos de processos de gestão da qualidade.

Dentre as normas e padrões, daremos destaque a ISO 12207, IEEE 730-201, ISO 25010, CMMI, IEEE 1028, ISO 8402, ISO 8402, ISO 14000 e ISO 13407. Cada qual apresentada em um momento diferente, mas sempre convergindo ao aspecto de melhoria no processo de desenvolvimento e garantia da qualidade.

Como a qualidade de software não busca apenas por erros no código, precisamos adotar modelos e técnicas que auxiliem no processo e em todo o desenvolvimento. Para tanto, são validações, verificações, inspeções, revisões e auditorias que proporcionam melhorias e correções com antecedência a problemas de usabilidade, segurança, requisitos funcionais e não funcionais, implantação e manutenção. Desenvolver um software é uma atividade complexa e multidisciplinar que envolve muitos profissionais. Garantir a qualidade de software não é apenas um processo antes ou após o desenvolvimento, mas durante o ciclo de vida do projeto.

Mesmo com todo esse direcionamento sobre a garantia da qualidade através de modelos, normas e técnicas para conduzir o processo, não é uma receita de bolo. Cada empresa e projeto possui suas particularidades e precisa adaptar-se à realidade individual.

Processos de inspeções, revisões, auditorias e usabilidade, que são tratados neste capítulo, possuem muitos aspectos subjetivos e que precisam ser adaptados a cada projeto.

Vamos então percorrer o caminho de muito descritivo, que parece cansativo, mas que é muito importante para que possamos estabelecer a melhor gestão da garantia da qualidade de software.

TEMA 1 – MODELOS DE PROCESSOS

Para compreendermos melhor qual a relação entre modelos de processos e técnicas de gestão de qualidade, é importante nos voltarmos ao processo de desenvolvimento de software.



O planejamento de um novo software sempre é um desafio, ainda mais diante das várias tecnologias e plataformas existentes atualmente. É um processo bem estabelecido, sendo ágil ou não, envolve uma especificação, um projeto, a implementação com sua devida validação, a manutenção e sua evolução.

A especificação consiste em engenharia do sistema, engenharia de requisitos e a descrição funcional do software. Já o projeto abrange a arquitetura, a UI e o detalhamento de cada funcionalidade do novo software. Durante a implementação temos, além da programação, testes unitários, de módulo e de integração. A manutenção e sua evolução consideram correções e melhorias do software.

Um processo de desenvolvimento de software pode ser do tipo cascata, incremental, espiral, unificado ou ágil (Sommerville, 2018).

Figura 1 – Relação existente entre processos e qualidade de software



Fonte: Teixeira, 2022.

A qualidade de software, por sua vez, também possui seu próprio processo, bem como caminha em conjunto com qualquer modelo de processo de desenvolvimento. Por exemplo, as revisões e auditorias fazem parte dos



projetos de software e estas garantem que os dois processos (qualidade e desenvolvimento) sejam bem-sucedidos. A Figura 1 relaciona alguns aspectos importantes entre o processo de desenvolvimento e a gestão da qualidade de software.

Com base na preocupação de uma estrutura comum a processos de ciclo de vida de software, surgiu o padrão ISO/IEC/IEEE 12207.

1.1 Norma ISO/IEC 12207

O padrão ISO/IEC/IEEE 12207 estabelece uma estrutura comum para processos de ciclo de vida de software. Ela é ampla e aplicável à aquisição de sistemas e produtos e serviços de software, fornecimento, desenvolvimento, operação, manutenção e descarte de produtos de software, bem como ao desenvolvimento de software, podendo ser desenvolvido interna ou externamente a uma organização (Laporte, 2018). A seguir, encontram-se os atributos relacionados à norma:

- O título transmite o escopo de todo o processo.
- O propósito descreve os objetivos da execução do processo.
- Os resultados observáveis são esperados com um desempenho bem-sucedido do processo.
- As atividades são conjuntos de tarefas coesas de um processo.
- As tarefas são requisitos, recomendações ou ações permitidas destinadas a apoiar a obtenção dos resultados.

Além dos atributos anteriores, a ISO 12207 ainda defende quatro conjuntos de processos conforme a Figura 2. Detalhamento dos Processos da ISO 12207:

1. **Processos Fundamentais** são os processos básicos e responsáveis pelo desenvolvimento do software e subdivididos em:
 - a. **Processo de aquisição:** refere-se a todas as aquisições necessárias para atendimento aos clientes, desde compras de prateleiras, até ferramentas computacionais.
 - b. **Processo de fornecimento:** refere-se à proposta de revisão e finalização de contrato com fornecedores.



- c. **Processo de desenvolvimento:** refere-se a tudo que diz respeito ao código, atividades de levantamento de requisitos, análise, testes, implantação e aceitação.
- d. **Processo de operação:** refere-se às atividades entre a utilização do software e o suporte ao usuário.
- e. **Processo de manutenção:** refere-se às atividades de solução de problemas do produto.

Figura 2 – Conjunto de Processos da ISO 12207



Fonte: Teixeira, 2022.

2. **Processos de Apoio** são os processos que estão ligados aos processos fundamentais e que podem ser aplicados antes ou no meio da utilização do processo fundamental, subdivididos em:
- a. **Documentação:** refere-se a toda documentação produzida para o ciclo de vida do software.
 - b. **Gerência de Configuração:** refere-se ao gerenciamento de versões dos artefatos produzidos.
 - c. **Gerência da Qualidade:** refere-se às verificações que satisfazem os requisitos e a execução dos processos planejados.
 - d. **Processo de verificação:** refere-se às verificações das funcionalidades em cada artefato do projeto.



- e. **Processo de validação:** refere-se às validações do produto final com o objetivo de realizar vários testes, como por exemplo testes unitários e testes de estresse.
 - f. **Processo de revisão conjunta:** refere-se à avaliação em conjunto com as verificações do processo e artefatos produzidos. Também são feitas revisões do gerenciamento de projeto para que não ocorram desvios de percurso do bom andamento.
 - g. **Processo de auditoria:** refere-se em assegurar que as atividades e tarefas estão sendo realizadas da forma correta conforme a norma estabelece.
 - h. **Processo de resolução de problemas:** refere-se aos processos de resolução de conflitos e problemas que possam causar riscos e prejuízos ao andamento do projeto.
3. **Processos Organizacionais** são os processos com atividades relacionadas ao gerenciamento e capacitação de pessoas, subdivididos em:
- a. **Processo de Gerência:** refere-se à implantação das atividades de gestão de processos e produtos nos projetos.
 - b. **Processo de Infraestrutura:** refere-se à designação da infraestrutura que permita a integração com ferramentas, técnicas, padrões, hardware e software.
 - c. **Processo de melhoria:** refere-se à avaliação, mensuração, controle e melhoria do processo do ciclo de vida do software.
 - d. **Processo de Treinamento:** refere-se à capacitação dos profissionais alocados para executarem as atividades de forma rápida e eficiente.
4. **Processos de Adaptação** são os processos que observam a realidade de cada organização e definem seus ciclos de desenvolvimento de acordo com as peculiaridades mais convenientes aos processos do ciclo de vida, subdivididos em:
- a. **Identificação do ambiente do projeto:** refere-se à análise e adaptação de fatores relacionados a pessoas, requisitos, software, políticas de qualidade, pontos estratégicos e cultura organizacional.
 - b. **Solicitação de informações:** refere-se às informações decorrentes para o processo de implantação e adaptação dos requisitos da norma ao ciclo de vida do desenvolvimento.



- c. **Seleção de processos, atividades e tarefas:** refere-se aos processos, suas atividades e tarefas que devem possuir documentos contendo características e responsáveis por desenvolvê-las e executá-las.
- d. **Documentação das decisões e motivos de adaptação:** refere-se ao plano de viabilidade adaptativa, análises de custos e riscos.

Vimos que o processo de garantia da qualidade (QA), segundo a ISO 12207, tem por finalidade garantir a aplicação efetiva de um processo de gestão da qualidade em projetos. Esse processo de QA fornece a confiabilidade dos requisitos de qualidade atendidos.

Como resultado da implementação do processo de controle de qualidade (QC), temos:

- Os procedimentos de controle de qualidade do projeto são definidos e implementados.
- Critérios e métodos para avaliações de garantia da qualidade são definidos.
- As avaliações dos produtos, serviços e processos do projeto são realizadas, consistentes com as políticas, procedimentos e requisitos de gestão da qualidade.
- Os resultados das avaliações são fornecidos às partes interessadas relevantes.
- Incidentes são resolvidos.
- Os problemas priorizados são tratados.

O projeto deve implementar atividades e tarefas de acordo com as políticas e procedimentos da organização, bem como aplicar medição em relação ao processo. Além da ISO 12207, a norma IEEE 730-201 fornece um guia adicional para processos de garantia de qualidade de software (SQA) (IEEE, 2014). Guia para atividades e tarefas de garantia de qualidade de software:

- Preparação do controle de qualidade. Essa atividade consiste nas seguintes tarefas:
 - Definição de uma estratégia de controle de qualidade.
 - Independência da QA de outros processos do ciclo de vida.
- Avaliação de produtos ou serviços. Esta atividade consiste nas seguintes tarefas:



- Avaliação de produtos e serviços em conformidade com critérios, contratos, padrões e regulamentos estabelecidos.
- Monitoramento da verificação e da validação das saídas dos processos do ciclo de vida são realizadas para determinar a conformidade com os requisitos especificados.
- Realizar avaliações de processos. Essa atividade consiste nas seguintes tarefas:
 - Avaliar a conformidade dos processos do ciclo de vida do projeto.
 - Avaliar ferramentas e ambientes que suportam ou automatizam o processo de conformidade.
 - Avaliar os processos do fornecedor quanto à conformidade com os requisitos do processo.
- Gerenciar registros e relatórios de controle de qualidade. Essa atividade consiste nas seguintes tarefas:
 - Criar registros e relatórios relacionados às atividades de controle de qualidade.
 - Manter, armazenar e distribuir registros e relatórios.
 - Identificar incidentes e problemas associados a avaliações de produtos, serviços e processos.
- Tratar incidentes e problemas. Essa atividade consiste nas seguintes tarefas:
 - Registrar, analisar e classificar os incidentes.
 - Identificar incidentes selecionados para associar a erros ou problemas conhecidos.
 - Registrar, analisar e classificar problemas.
 - Identificar as causas-raiz e o tratamento dos problemas, sempre que possível.
 - Priorizar o tratamento de problemas (resolução de problemas) e acompanhar as ações corretivas.
 - Analisar tendências em incidentes e problemas.
 - Identificar melhorias em processos e produtos que possam prevenir futuros incidentes e problemas.
 - Informar às partes interessadas designadas sobre o status de incidentes e problemas.
 - Acompanhar incidentes e problemas até o fechamento do projeto.



O padrão ISO 12207 pode ser usado em um ou mais dos seguintes modos (Laporte, 2018):

- **Por uma empresa:** para ajudar a estabelecer um ambiente de processos desejados. Esses processos podem ser apoiados por uma infraestrutura de métodos, procedimentos, técnicas, ferramentas e pessoal treinado. A organização pode então empregar este ambiente para executar e gerenciar seus projetos e sistemas de software de progresso através de seus estágios de ciclo de vida. Neste modo, este documento é usado para avaliar a conformidade de um ambiente declarado e estabelecido com suas disposições.
- **Por um projeto:** para ajudar a selecionar, estruturar e empregar os elementos de um conjunto estabelecido de processos de ciclo de vida para fornecer produtos e serviços.
- **Por um fornecedor:** para ajudar a desenvolver um acordo sobre processos e atividades.
- **Por organizações e avaliadores:** servir como modelo de referência de processo para uso no desempenho de avaliações de processo que podem ser usadas para apoiar a melhoria de processos organizacionais.

O padrão ISO 12207 descreve as limitações de seu uso da seguinte forma: não é aplicado a um sistema específico ou modelo de ciclo de vida de software, metodologia de desenvolvimento, método, modelo ou técnica. Ele é genérico e pode funcionar dentro das equipes de trabalho mais tradicionais ou ágeis.

Os usuários desse documento são responsáveis por selecionar um modelo de ciclo de vida para o projeto e mapear os processos, atividades e tarefas deste documento nesse modelo. As partes também são responsáveis por selecionar e aplicar metodologias, métodos, modelos e técnicas adequados ao projeto.

Ele ainda não estabelece um sistema de gestão nem exige o uso de qualquer padrão de sistema de gestão. Também não detalha os itens de informação em termos de nome, formato, conteúdo explícito e mídia de gravação.

No entanto, essa norma é compatível com as fábricas de Software, que buscam minimizar suas etapas no gerenciamento do ciclo de vida do software. Sua composição é definida por processos, atividades e tarefas.



Figura 3 – Estrutura do Ciclo de Vida na ISO/IEC 12207



Fonte: Teixeira, 2022.

A estrutura do ciclo de vida na ISO/IEC 12207 estabelece um detalhamento em processos, para os quais são atribuídas atividades e que por sua vez são especificadas em Tarefas (Figura 3).

A norma ISO/IEC 12207 busca a conscientização e organização das melhorias na gestão dos processos. Como já comentamos, ela é aberta e pode ser adaptada a cada realidade.

TEMA 2 – Modelos de Qualidade para Produtos de Software

1.1 Norma ISO/IEC 25010)

A norma ISO/IEC 25010 está relacionada à definição de modelos de avaliação para garantia da qualidade de software e outros sistemas em geral. Possui cinco características principais, as quais podem ter subcaracterísticas. É um modelo aplicável à disciplina de sistemas humano-computador e inclui o uso de sistemas computacionais e software. Já em relação ao um modelo de qualidade de um produto de software, possui oito características principais, com suas subcategorias, relacionadas à qualidade: funcionalidade, eficiência de desempenho, usabilidade, confiabilidade, compatibilidade, segurança, manutenção e portabilidade. (ISO 25010, 2017). É importante salientar que essa norma substituiu a ISO 9126.



A ideia central é que com a aplicação dessa norma conseguimos avaliar a qualidade do software. A norma também possui um modelo de qualidade de dados, o qual é complementar a esse modelo. As principais atividades aplicadas ao desenvolvimento de software incluem modelos de qualidade que auxiliam na identificação de requisitos, validação da abrangência da definição dos requisitos, identificação dos objetivos do projeto de software, identificação dos critérios de controle de qualidade para garantia da qualidade, identificação dos critérios de aceitação para que um produto de software seja concluído, bem como o estabelecimento de medidas e métricas de qualidade para apoiar toda a atividade de garantia de qualidade de software.

Figura 4 – Características da ISO/IEC 25010 - Parte I



Fonte: Teixeira, 2022.

Figura 5 – Características da ISO/IEC 25010 - Parte II



Fonte: Teixeira, 2022.

Com base nas figuras 4 e 5, vamos compreender de forma mais detalhada as características e subcaracterísticas estabelecidas pela ISO/IEC 25010:

1. **Funcionalidade:** característica que representa o quanto produto fornece as funcionalidades para atendimento das necessidades implícitas e explícitas nos documentos de especificação do software.
Subcaracterísticas:
 - a. **qualidade funcional:** objetivos do usuário cumpridos através do conjunto de funcionalidades de todas as tarefas especificadas.
 - b. **correção funcional:** grau de precisão nos resultados fornecidos pelo software.
 - c. **adequação funcional:** quando as funcionalidades realizam as tarefas e objetivos especificados para o software.
2. **Confiabilidade:** característica que representa a execução correta das funcionalidades especificadas para um período especificado.
Subcaracterísticas:
 - a. **maturidade:** quando o software atende às necessidades de confiabilidade.
 - b. **tolerância a falhas:** quando o software opera mesmo diante de falhas de hardware, software ou plataforma em nuvem.



- c. **recuperabilidade**: quando o software é capaz de recuperar os dados em caso de interrupção ou falha.
- d. **disponibilidade**: software ou feature operacional e acessível.
- 3. **Usabilidade**: característica que representa como os usuários utilizam o software com eficácia, satisfação e eficiência. Subcaracterísticas:
 - a. **reconhecimento de adequação**: quando o software pode ser reconhecido por ser apropriado às necessidades.
 - b. **aprendizagem**: quando é fácil o aprendizado no uso do software.
 - c. **operabilidade**: quando o software possui atributos que facilitam a operação e o controle.
 - d. **proteção contra erros do usuário**: quando o software protege o usuário contra erros.
 - e. **estética da interface do usuário**: quando a UI é agradável ao usuário.
 - f. **acessibilidade**: quando o software se preocupa em ampliar a gama de características e capacidades de uso.
- 4. **Eficiência de Desempenho**: sobre o desempenho relacionado à quantidade de recursos utilizados. Subcaracterísticas:
 - a. **comportamento temporal**: característica sobre tempo de resposta e taxa de rendimento para execução do software.
 - b. **utilização de recursos**: quantidade e tipos de recursos que o software possui durante a execução.
 - c. **capacidade**: limites máximos de parametrização do software.
- 5. **Manutenibilidade**: como o software é modificado, melhorado, corrigido ou adaptado de acordo com as mudanças no ambiente ou de requisitos. Subcaracterísticas:
 - a. **analísabilidade**: impacto sobre a avaliação das mudanças pretendidas.
 - b. **modificabilidade**: impacto sobre mudanças no software sem introdução de defeitos ou problemas com a qualidade.
 - c. **modularidade**: impacto mínimo quando os componentes ou o software são alterados.
 - d. **testabilidade**: eficácia dos critérios de testes para o software ou componentes.
 - e. **reutilização**: quão bom é o reaproveitamento de componentes existentes no software.



6. **Portabilidade:** característica do software ao ser transferido de um ambiente para outro. Subcaracterísticas:
 - a. **adaptabilidade:** como um software se adapta a um hardware, software ou outro ambiente de uso diferentes.
 - b. **instalação:** sucesso na instalação e desinstalação do software.
 - c. **capacidade de substituição:** como é fácil a substituição de um software por outro.
7. **Segurança:** como o software protege informações e dados contra vulnerabilidades de segurança. Subcaracterísticas:
 - a. **confidencialidade:** como o software garante que os dados sejam acessíveis apenas para quem é autorizado.
 - b. **integridade:** como o software impede o acesso não autorizado e a modificação de programas ou dados.
 - c. **não repúdio:** como comprovar ações ou eventos que ocorrem.
 - d. **responsabilidade:** como o usuário não autorizado pode ser rastreado.
 - e. **autenticidade:** como a identidade de um recurso pode ser provada.
8. **Compatibilidade:** característica que define o quanto o software é capaz de trocar informações e executar funções necessárias quando for compartilhado. Subcaracterísticas:
 - a. **coexistência:** quão bem o software desempenha suas funções com eficiência ao compartilhar ambiente e recursos comuns com outro software, sem impactar negativamente.
 - b. **interoperabilidade:** quão capaz o software é capaz de trocar informações ou usar informações.

A ISO/IEC 25010 é a base de um sistema de avaliação para garantia da qualidade do software. Esse modelo determina características inerentes à avaliação das características e propriedades do software.

2.2 Outras Normas e Modelos de Qualidade

É importante ressaltar que, quando falamos de qualidade de processo do ciclo de vida e do software, não temos uma receita de bolo. Precisamos adaptar características de normas e modelos de maturidade.

A ideia principal é conhecermos o caminho na busca da excelência através de modelos de qualidade, padrões, referências ou processos específicos



para o desenvolvimento de software, pensando tanto nos processos quanto no software. São várias referências que nos auxiliam na busca pela qualidade, como por exemplo ITIL, ISO/IEC 20000-1, processos de governança de TI através do CobiT, padrões ISO 27000 para segurança da informação e ISO/IEC 29110 para pequenas empresas (Laporte, 2018).

O CMMI para Desenvolvimento (CMMI-DEV) abrange uma área ampla e agrega práticas de engenharia de sistemas e desenvolvimento de processos e produtos integrados. São duas versões do CMMI: uma versão inicial em estágios e uma versão contínua e está disponível em vários idiomas.

O desenvolvimento de software não se baseia nas leis da natureza, portanto o número de softwares com problemas de prazos não cumpridos, defeituosos ou que não atendam aos usuários são consideráveis. Ao contrário, ele é baseado nas leis da lógica e da matemática. Também alia o uso de práticas bem definidas para garantia e qualidade do software. Os padrões aplicados à engenharia de software de fato são guias para práticas de gestão. Os padrões ISO possuem quatro princípios:

1. as normas ISO são utilizadas para uma necessidade do mercado.
2. normas baseadas na experiência mundial.
3. são resultado de um processo com várias partes interessadas.
4. são normas que seguem um consenso.

A grande questão encontra-se na aplicação de padrões ou modelos que atuem diretamente na melhoria contínua e na maturidade de processos de desenvolvimento de software através da gestão da garantia da qualidade de software.

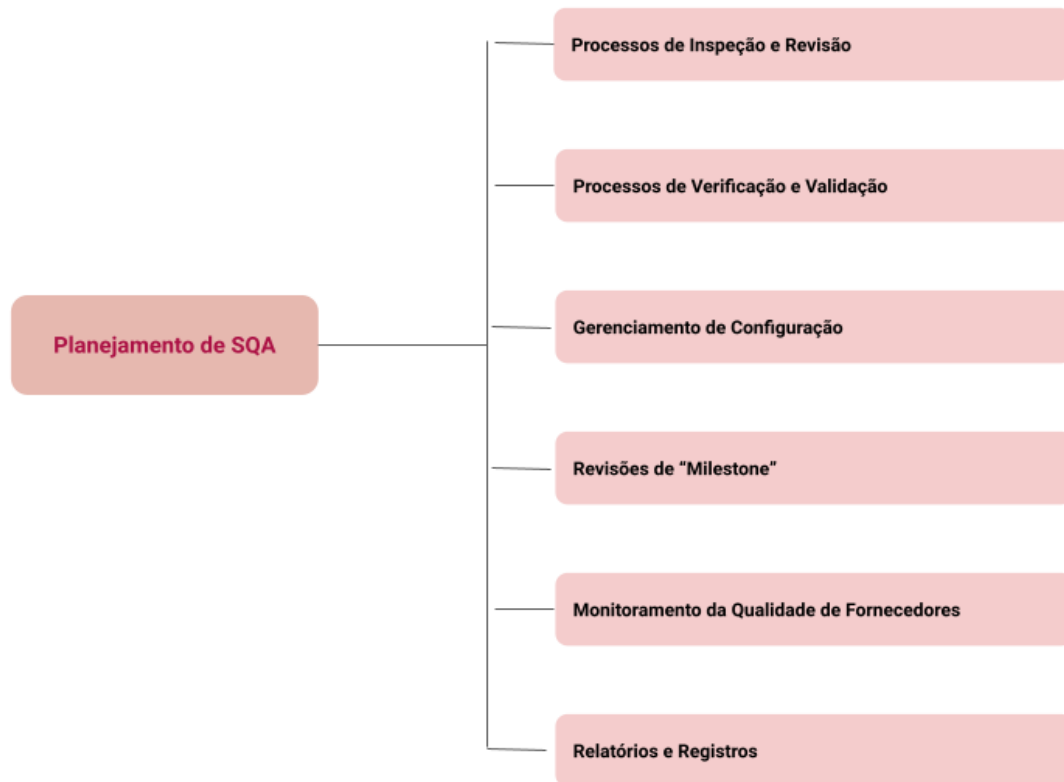
TEMA 3 – REVISÕES E INSPEÇÕES

As revisões, inspeções e auditorias na engenharia de software são importantes para a concepção de uma estrutura e requisitos detalhados para verificação e validação nos processos de desenvolvimento. A garantia de sucesso das revisões e auditorias garantem o cumprimento de todos os requisitos especificados para projeto do software, para os testes e controle de configuração. Também contribuem para que os esforços individuais ou do time sobre a manutenção gerem lucro, redução de custos e melhoria da qualidade.



E afinal de contas, quando é necessário aplicarmos revisões e auditorias? Quando queremos garantir que o software atenda aos requisitos exigidos (Summers, 2011).

Figura 6 – Revisões e Auditorias através do Ciclo de Vida do Desenvolvimento de Software



Fonte: Teixeira, 2022.

A Figura 6 esquematiza como podemos fazer a preparação dos planos de implementação e uso dos modelos e padrões de qualidade através do ciclo de vida do desenvolvimento de software. Quem conduz este plano (Figura 6) e a execução do gerenciamento da qualidade é o time de SQA (Garantia da Qualidade de Software).

3.1 Planos de Revisões

Os planos de revisões são feitos de forma incremental, sendo agendados e realizados pelo grupo focal SQA. Essas revisões formam a base para garantir a conformidade de que as atividades de desenvolvimento realizadas estão de



acordo com os planos e procedimentos, além de estarem de acordo com os requisitos. São seis etapas para conclusão da revisão:

1. Planejamento;
2. Execução;
3. Registro;
4. Relatório;
5. Ação Corretiva;
6. Melhoria do Processo.

A execução que é elaborada no planejamento através de processos, procedimentos, entrevistas e outros documentos mensais tem por objetivo a revisão e inspeção para assegurar que as atividades de desenvolvimento sejam completadas conforme o planejado.

Todas as atividades de SQA são registrados e também traduzidos em relatórios, diagramas e gráficos para auxiliar a melhoria do processo de desenvolvimento de software.

Esses relatórios e registros auxiliarão nas ações corretivas que contribuirão para melhoria contínua do processo e também do software (Summers, 2011). Um exemplo de lista de verificação de revisões e inspeções contém:

1. detalhes do escopo da revisão bem definida para cobertura de todos os itens relacionados ao software.
2. material de apresentação claro em detalhes e consistente com o escopo das revisões.
3. roteiro para revisão dos dados que devem ser disponibilizados:
 - a. requisitos do software;
 - b. definição de UI;
 - c. definição de Design;
 - d. documentação de versionamento de código;
 - e. código e os resultados dos planos de testes;
 - f. rastreabilidade dos requisitos;
 - g. inspeção e demonstração de resultados;
 - h. resultado das análises;
 - i. lista de desvios;
 - j. configuração e instalação do software.



As inspeções e revisões de software analisam e verificam requisitos, modelos de projeto, código e testes estabelecidos, sejam unitários, de sistema, de integração, ponto a ponto ou quaisquer outros tipos de testes. As inspeções sempre buscam como fonte de trabalho o código e outros modelos de projeto.

Durante os testes, é possível que deixemos para trás erros por vários motivos, e as inspeções e revisões feitas posteriormente colaboram para que defeitos e saídas inesperadas possam ser capturadas. Versões não concluídas de um software também são inspecionadas e, se for necessário, pode-se aplicar testes para as funcionalidades já entregues. Na inspeção, é importante considerar aspectos como padrões de design, portabilidade, ineficiências em algoritmos, estilos de programação inadequados, entre outras questões (Summers, 2011).

Inspeção de código não é algo inovador, pois desde o início da engenharia de software, há mais de 50 anos, a preocupação com essa atividade existe.

Uma questão muito importante é que uma inspeção não deve substituir todo o planejamento de testes. Inspeções e revisões são feitas num segundo momento para descoberta de defeitos obtidos através de interações não esperadas, problemas de performance, entre outros requisitos não funcionais.

TEMA 4 – AUDITORIA DE SOFTWARE

Uma auditoria de software é como um processo de revisão com detalhamento dos recursos utilizados pelo software, no qual o procedimento de auditar objetiva que as ferramentas utilizadas estejam em conformidade para garantir segurança e integridade. Existem diferentes tipos de auditoria (Summers, 2011):

1. auditorias para verificação da conformidade com uma determinada norma: descritas nas normas da International Organization for Standardization (ISO). Por exemplo ISO 9001 e IEEE 1028.
2. auditorias de conformidade com um modelo de maturidade do software: Por exemplo, com o modelo Capability Maturity Model Integration (CMMI).
3. auditorias solicitadas pelo time de gestão: por exemplo, para verificação do progresso de um projeto em relação ao seu plano aprovado, ou por quaisquer motivos internos da empresa.



A execução da auditoria pode ser atribuída a um consultor externo ou a membros do time que não estejam envolvidos com o projeto a ser auditado. Ela é conduzida através da especificação de perguntas que a auditoria deve responder, um cronograma da auditoria, quais participantes da auditoria e o formato do relatório de auditoria (por exemplo, descobertas e recomendações).

Para condução das auditorias temos a norma ISO 12207, a norma IEEE 1028, um modelo de auditoria recomendado pela CMMI, uma seção de auditoria do plano SQA do padrão IEEE 730.

E juntamente com as normas, temos diferentes tipos de certificados de conformidade para as auditorias responderem às várias necessidades da empresa, do time de desenvolvimento, dos clientes e dos fornecedores. A independência dos auditores e dos custos variam de acordo com o tipo de auditoria interna ou externa.

A auditoria interna é realizada em intervalos planejados para fornecimento de informações sobre o sistema de gestão da qualidade. O planejamento considera os requisitos da empresa para seu sistema de qualidade e os requisitos das normas internacionais. Como deve-se organizar o processo de auditoria interna:

1. planejamento, estabelecimento, implementação e manutenção do programa de auditoria, inclusive estabelecendo a frequência, métodos, responsabilidades, requisitos de planejamento e relatórios que consideram as mudanças e os resultados de auditorias anteriores.
2. definição dos critérios de auditoria e o escopo de cada auditoria.
3. seleção dos auditores que garantam a objetividade e imparcialidade do processo de auditoria.
4. garantia de que os resultados sejam de fato relatados à administração.
5. planejamento das ações de correções.
6. retenção das informações documentadas como evidência da implementação do programa de auditoria e de seus resultados.

No caso das auditorias de qualidade, o processo é independente e estruturado para determinar se as atividades do projeto estão em conformidade com as políticas, processo e procedimentos organizacionais do projeto.

Entre os objetivos de uma auditoria de qualidade, encontram-se (Summers, 2011):



1. identificação de todas as boas e melhores práticas implementadas.
2. identificação de todas as não conformidades, lacunas e deficiências.
3. compartilhamento de boas práticas introduzidas e implementadas em projetos similares.
4. proatividade na assistência de forma positiva para melhoria da implementação de processos para auxiliar o time no aumento da produtividade.
5. destaque das contribuições de cada auditoria no repositório de lições aprendidas da organização.

As auditorias de qualidade podem ser planejadas ou feitas de forma aleatória, além de serem conduzidas por auditores internos ou externos. Elas ainda podem confirmar a execução de modificações de mudanças aprovadas, como correções de defeitos e ações corretivas. Além de verificar a conformidade, deve-se notar que essa definição de auditoria destaca o fato de que a auditoria também pode ser feita para melhorar o desempenho organizacional.

Um ponto importante na questão das auditorias é que o modelo de custo de qualidade abrange a auditoria como prática de avaliação e a verificação dos custos do software durante as várias fases do ciclo de vida do projeto.

4.1 Auditoria e Resolução de Problemas de Software

A ISO/IEC/IEEE 12207 define os requisitos de auditoria e do processo de gerenciamento de decisão. E aborda dois principais momentos:

1. Processo de Avaliação e Controle do Projeto.
2. Processo de Gestão de Decisões.

O objetivo do Processo de Avaliação e Controle do Projeto é o de avaliar se os planos estão alinhados e viáveis além de determinar o status do projeto, de desempenho técnico e do processo. Outro aspecto relevante é a execução direta para ajudar a garantir que o desempenho esteja de acordo com os planos e cronogramas, dentro dos orçamentos projetados, para satisfazer os objetivos técnicos. Uma importante tarefa e obrigatória desse processo é a realização de **revisões gerenciais e técnicas, auditorias e inspeções**.

Já no caso do Processo de Gestão de Decisões, ele tem por objetivo fornecer uma estrutura analítica estruturada para identificar, caracterizar e



avaliar objetivamente um conjunto de alternativas para uma decisão em qualquer ponto do ciclo de vida e selecionar a melhor ação possível.

Dentre as principais atividades do processo de gestão de decisões, encontram-se:

1. determinar alternativas preferenciais para cada decisão.
2. registrar a resolução, a fundamentação da decisão e as premissas.
3. registrar, acompanhar, avaliar e relatar decisões.

A ISO 12207 não se limita às auditorias a serem realizadas, mas também a auditorias de gerenciamento de configuração.

4.2 Auditoria segundo o padrão IEEE 1028

O padrão IEEE 1028 descreve diversos tipos de revisões e auditorias, além de procedimentos para execução das revisões. Ele explica como realizar as auditorias, porém não qual a necessidade e como utilizar os relatórios de auditoria.

Quadro 1 – Características da Auditoria segundo a IEEE 1028

ETAPA DA AUDITORIA	DESCRIÇÃO
Objetivo	Avaliar a conformidade com padrões e regulamentos objetivos
Tomada de Decisão	Organização, início, adquirente, cliente ou usuário a ser auditado
Verificação de mudanças	Responsabilidade da empresa auditada
Tamanho do time	De uma a cinco pessoas
Participação do time	Auditores e a empresa auditada podem ser chamadas a fornecer evidências
Auditor Líder	Liderança do time
Volume do Material	De moderado a alto, dependendo dos objetivos específicos da auditoria
Apresentador	Os auditores coletam e examinam as informações fornecidas pela organização auditada

Fonte: Elaborado por Teixeira, 2022 com base em Summers, 2011.

O objetivo do padrão IEEE 1028 é definir as revisões sistemáticas e auditorias que podem ser aplicadas aos processos de aquisição, fornecedor,



desenvolvimento, operação e manutenção de software. A norma descreve como conduzir uma auditoria, além de descrever os requisitos mínimos aceitáveis para auditorias de software.

As auditorias são procedimentos um pouco mais formais, e, portanto, acabam por se basear em normas e padrões, gerando relatórios e outros tipos de resultados formais.

TEMA 5 – A USABILIDADE E A QUALIDADE DE SOFTWARE

O teste de usabilidade é considerado como uma técnica utilizada para avaliação de software e são realizados com usuários representativos do público-alvo. Os participantes do teste recebem tarefas e são observados e registrados pelo analista que conduz as atividades. São três principais tipos de testes de usabilidade (Lewis, 2017):

1. em laboratório
2. em contato
3. remoto.

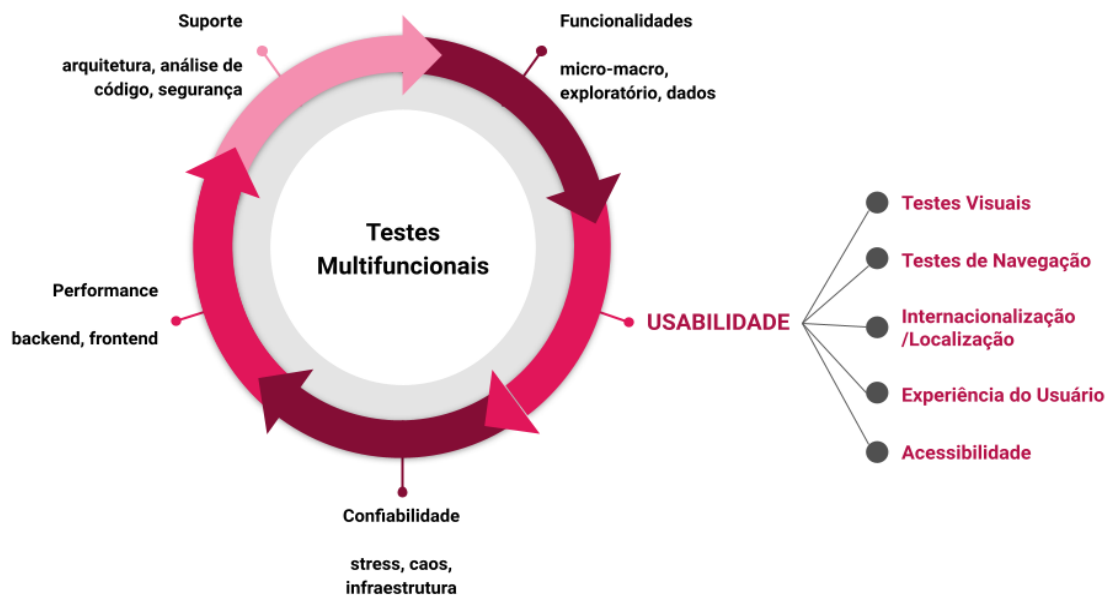
Todos esses tipos são considerados testes comportamentais, pois avaliam todas as ações sobre o software, que pode ser um app, app-Web ou quaisquer outros tipos de software que interajam com o humano.

Inclusive, a usabilidade é uma técnica que surgiu dentro da área de IHC (Interação Humano-Computador), a qual busca compreender como o ser humano usa dispositivos eletrônicos (computadores, celulares, tablets, IoT, entre outros) e quais tipos de investigações podem ser feitas em relação às formas de interação.

O teste, segundo Mohan (2022), é uma prática que valida o comportamento do aplicativo. Além dos testes granulares do aplicativo, como classes, entradas, mensagens de log, erros de código, integrações, fluxo de ponta a ponta, precisamos testar os aspectos multifuncionais de qualidade, como segurança, desempenho, acessibilidade, usabilidade e muito mais. A categoria testes de usabilidade representa o conjunto de requisitos que afetam a experiência do usuário, o visual do aplicativo, a compatibilidade do navegador, a acessibilidade e a facilidade de uso, conforme é observado na Figura 7.



Figura 7 – Testes Multifuncionais com destaque aos testes de usabilidade



Fonte: Teixeira, 2022.

Os testes de usabilidade são testes de avaliação, podem avaliar um software, um website, um protótipo ou até um simples desenho num papel. Possui usuários representativos, um roteiro de tarefas e um analista para observar e registrar. Investiga aspectos como se os filtros de busca são fáceis de usar, quando o usuário abandona um carrinho de compras no B2C, como podemos melhorar o checkout, entre tantas outras funcionalidades.

Apesar de ser um teste não apenas aplicável a software, é muito utilizado especialmente por questões da interação do software com o ser humano. Dentro da garantia da qualidade, há algumas normas que auxiliam com roteiros e boas práticas devido à subjetividade do assunto.

Dentre as normas aplicáveis à usabilidade, temos a ISO 8402, com normas para estabelecer a capacidade de como um item desempenha uma função requerida. Também temos a ISO 14000, que consta de um conjunto de atributos que dizem respeito à capacidade do software de manter o nível de desempenho. Outra norma utilizada é a ISO 13407, a qual atinge a qualidade no uso através de técnicas de IHC durante todo o ciclo de desenvolvimento de software. Cada norma é específica para o desenvolvimento de software com considerações diversas.

Enfim, dentro da gestão da garantia da qualidade, encontramos algumas normas e boas práticas que auxiliam na melhor prática de testes de usabilidade para garantirmos a qualidade na experiência do nosso usuário de software.



FINALIZANDO

Trabalhamos neste capítulo conteúdos bastante teóricos, mas que auxiliam e proporcionam clareza nos processos de desenvolvimento de software quanto no produto final.

Inicialmente a adoção de normas e padrões é algo que nos parece difícil e complexo, pois envolve que todos estejam na mesma página, engajados e desejando o objetivo final: a gestão da garantia da qualidade. À medida que as coisas se consolidam, não conseguimos mais imaginar que possamos fazer as atividades sem um contexto bem definido com metodologias e técnicas que dão suporte a todo o processo de desenvolvimento e de gestão da qualidade.

Observação, análise, validações, verificações, auditorias e inspeções são atividades que precisam de colaboração e engajamento dos times, tanto de engenharia de software quanto de gestão da qualidade. São tarefas exaustivas e que exigem concentração, olhar clínico e antecipações aos problemas que possam ocorrer no uso do software pelos usuários. Uma vez que consigamos garantir a melhor experiência de uso do software, certamente o sucesso será garantido.

Usuários de aplicativos e outros tipos de software estão cada dia mais exigentes e ficam inseguros em relação a falhas, erros e bugs, tempo de resposta ruim, vazamentos de dados, falta de consistência e integridade.

A adoção de normas e modelos também fazem parte da evolução da maturidade dos processos e criam maior confiabilidade entre parceiros, fornecedores e clientes.



REFERÊNCIAS

ARNON, A. **Complete guide to test automation**: techniques, practices, and patterns for building and maintaining effective software projects. Apress, 2018.

ACM – Association for Computing Machinery. Disponível em: <<https://www.acm.org/>>.

IEEE SA. IEEE Standard for Software Quality Assurance Processes, 2014. Disponível em: <<https://standards.ieee.org/ieee/730/5284/>>. Acesso em: 8 ago. 2022.

_____. **ISO/IEC/IEEE International Standard - Systems and software engineering--Software life cycle processes--Part 2: Relation and mapping between ISO/IEC/IEEE 12207:2017 and ISO/IEC 12207:2008**. Disponível em: <<https://standards.ieee.org/ieee/12207-2/10353/>>. Acesso em: 8 ago. 2022.

GALIN, D. **Software quality**: concepts and practice. IEEE Press; Wiley, 2018.

IEEE – Advancing Technology for Humanity. Disponível em: <<https://www.ieee.org/>>.

ISO/IEC 25010:2011. Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — System and software quality models. 2017. Disponível em: <<https://www.iso.org/standard/35733.html>>. Acesso em: 8 ago. 2017.

JONES, C. **A guide to selecting software measures and metrics**. Taylor & Francis Group LLC, 2017.

LAPORTE, C; APRIL, A. **Software quality assurance**. IEEE Press. Wiley, 2018.

LEWIS, W. E. **Software testing and continuous quality improvement**. 3. ed. CRC Press, O'Reilly, 2009.

LÉLIS, E. C. **Gestão da qualidade**. São Paulo: Pearson Prentice Hall, 2012.

MOHAN, G. **Full stack testing**. CRC Press, O'Reilly, 2022.

MATSINOPOULOS, P. **Practical test automation**: learn to use jasmine, rspec, and cucumber effectively for your tdd and bdd. Los Altos, California, USA: CA Press/Apress, 2020.



NADER-REZVANI, N. **An executive's guide to software quality in an agile organization**: a continuous improvement journey. Los Altos, California, USA: CA Press/Apress, 2019.

PRESSMAN, R. S. **Engenharia de software**: uma abordagem profissional. 7. ed. Porto Alegre: AMGH, 2011.

RIOS, E. **Documentação de teste de software**. 2. ed. Iteste: Instituto de teste de software, 2008.

SUMMERS, L. S. **Software engineering reviews and audits**. CRC Press, O'Reilly, 2011.

SOMMERVILLE, I. **Engenharia de software**. São Paulo: Pearson Education do Brasil, 2018.

SALE, D. **Testing python**: applying unit testing, TDD, BDD, and acceptance testing. IEEE Press/Wiley, 2014.

SMART, J. F. **BDD in action**: behavior-driven development for the whole software lifecycle. Manning Publications Co., 2015.

SHIGUNOV NETO, A. **Introdução à gestão da qualidade e produtividade**: conceitos, história e ferramentas. Curitiba: InterSaberes, 2016.