QUALIDADE DE SOFTWARE

AUTORA
MAYB ANDRADE

QUALIDADE DE *SOFTWARE*

AUTOR DO ORIGINAL
MAYB ANDRADE

1ª EDIÇÃO
SESES
RIO DE JANEIRO 2015

Estácio

Conselho editorial FERNANDO FUKUDA, SIMONE MARKENSON, JEFERSON FERREIRA FAGUNDES

Autor do original MAYB ANDRADE

Projeto editorial ROBERTO PAES

Coordenação de produção RODRIGO AZEVEDO DE OLIVEIRA

Projeto gráfico PAULO VITOR BASTOS

Diagramação FABRICO

Revisão linguística ADERBAL TORRES BEZERRA

Imagem de capa NOME DO AUTOR — SHUTTERSTOCK

Diretoria de Ensino — Fábrica de Conhecimento
Rua do Bispo, 83, bloco F, Campus João Uchôa
Rio Comprido — Rio de Janeiro — RJ — CEP 20261-063

Sumário

Prefácio 7

1. Visão Geral de Qualidade Software 10

Conceito de Qualidade de *Software* 10
Fatores de Qualidade de *Software* 17
Métricas de *Software* 18
Revisões de *Software* 21

2. Normas de Qualidade de *Software* 26

Garantia de Qualidade de *Software* (Software Quality Assurance – SGA) 26
Princípios e Modelos da Norma ISO 9000 29
NBR ISO 9126 29
NBR ISO 12119 34

3. Modelos da Norma ISO 44

Norma ISO 9241 – Usabilidade do Produto 44
Norma ISO 9241 parte -11 45
Norma ISO 14598 (avaliação de produto) 50

4. Modelos de Melhoria e Avaliação de Processo de *Software* 62

Melhoria de Processo de Software 63
Modelos de Melhoria de Processo de Software 64
NBR/ISO 9000-3 70
NBR/ ISO 12207 - Processos de ciclo de vida do software 72

5. Modelos de Qualidade de Processo de *Software* 80

SPICE (*Software Process Improvement and Capability Determination*) 81
Transição para a Norma ISO/IEC 15504 87
Modelo CMMI (*Capability Maturity Model*) 88
Modelo Mps.Br 89
Gerência de Riscos na Qualidade de Software 91

Prefácio

Prezado(a) aluno(a)

A qualidade tem sido um fator de diferenciação no mercado atual. A exigência por qualidade tem aumentado em todas as áreas e afetado também a indústria de software. Com os computadores, cada vez mais, fazendo parte da vida das pessoas, a produção de softwares vem aumentando e tornando os clientes mais exigentes. A grande exigência dos clientes por melhores softwares tem obrigado os desenvolvedores a aperfeiçoarem o seu produto final para continuarem competindo no mercado. Esses clientes deixaram de se preocupar apenas com o preço e passaram a buscar um produto mais confiável e com mais qualidade.

Após alguns anos de experiência no desenvolvimento de software, percebeu-se que existem alguns fatores de qualidade, considerados pelos clientes, não estão relacionados especificamente às características de qualidade do produto final. Esses fatores relacionam-se mais ao processo de software, à forma como ele é gerenciado e controlado.

Ao se melhorar a qualidade do processo de software, tem-se maior probabilidade de se obter um produto final mais adequado às expectativas do cliente, no entanto, a realização de uma melhoria do processo de software não é uma tarefa trivial. Para que o processo de software possa cumprir seus objetivos é necessário um planejamento detalhado que mostre a realidade do processo atual, a meta que se almeja com a melhoria, a estratégia para se atingir essa meta e os planos de ação.

Para auxiliar a melhoria do processo existem abordagens que descrevem como a organização pode avaliar o seu estado atual e a partir dessa avaliação procurar melhorar o seu processo. A melhoria do processo deve ser consciente e o grau a ser atingido deve ser bem definido.

Dentro desse contexto, nossa disciplina vem lhe oferecer os conceitos necessários para compreender o significado da qualidade de software bem como a forma de alcançá-la.

Vamos lá que o assunto é atual e muito interessante!

Bom estudo !!

1

Visão Geral de Qualidade Software

1 Visão Geral de Qualidade Software

Neste capítulo definiremos o que vem a ser Qualidade de *Software*, veremos a qualidade pode ter diferentes interpretações, dependendo de quem a está avaliando.

Comentaremos sobre as diferenças de qualidade de produto e de processo de *software* e finalizaremos com alguns fatores relacionados à Qualidade de *Software*.

OBJETIVOS

- Compreender as diferentes visões da Qualidade de *Software*;
- Compreender os fatores da qualidade de um *software*;
- Entender o que são métricas de *software* e;
- A importância das revisões de *software*.

REFLEXÃO

Quando apareceram os primeiros computadores e depois com a evolução dos mesmos, todos ficamos fascinados e também curiosos como essas máquinas podiam fazer tantas coisas em tão pouco tempo, como de repente elas começaram a fazer parte das nossas vidas, de tal maneira que hoje não nos imaginamos sem elas, não é?

Pois bem, agora não tem mais como voltar atrás, não vivemos mais sem nossos amados computadores, que também não existem sem um *software*, não é verdade? Mas para que tudo funciona na mais perfeita ordem não basta simplesmente ter o *software*, esse precisa ter **qualidade**! Vamos lá, vamos entender o que vem a ser a Qualidade de *Software*!

1.1 Conceito de Qualidade de *Software*

Com a constante demanda gerada pela vida moderna, cada vez mais os computadores passam a integrar a rotina diária e a produção de *software* vem tendo um aumento constante. A exigência por qualidade estende-se também à área de *software* e pode ser considerada o centro das atenções para o desenvolvimento de *software*. Por exemplo, do ponto de vista dos fornecedores de *software*, qualidade não é mais um fator de vantagem no mercado, mas uma condição necessária e pode-se dizer indispensável para que seja possível competir com sucesso.

Mas vamos parar e analisar, como chegamos a essa era da Qualidade de *Software*?

Desde os tempos remotos, muitos problemas no desenvolvimento dos sistemas computacionais já se faziam sentir. Em 1968 o Comitê de Ciências da OTAN reuniu 50 especialistas, cientistas e profissionais da indústria de *software* para discutir possíveis soluções para o que passou a ser conhecido como a *Crise do Software*.

Nesse encontro se firmou o termo *Engenharia de Software*, e foi definida formalmente a necessidade da aplicação de uma abordagem sistemática, disciplinada e quantificável para o desenvolvimento, operação e manutenção de produtos de *software*.

Vamos relembrar algumas coisas e observar a engenharia de *software* através de uma perspectiva histórica:

- **Década de 60 e os anos que a antecedem:** podem ser chamados de Era Funcional – quando aprendeu-se a usar a tecnologia da informação para suprir as necessidades institucionais e começar a integrar o *software* nas operações diárias das instituições.

- **Década de 70:** ficou conhecida como a Era do Método - nessa fase, como as organizações de *software* foram caracterizadas por maciços atrasos nos planos e constantes ultrapassagens dos custos planejados, a maior preocupação era planejar e controlar os projetos de *software*. Foi quando os modelos de ciclo-de-vida, baseados em várias fases, foram introduzidos e analisados

- **Década de 80:** foi a era do Custo - O custo do hardware começou a cair e a tecnologia da informação se tornou acessível às pessoas, não mais apenas às instituições. A competição das indústrias tomou um rumo diferente pois aplicações de baixo custo puderam ser largamente implementadas. A importância da produtividade no desenvolvimento de *software* aumentou significativamente. Nessa fase, vários modelos de custo na Engenharia de *Software* foram implementados e usados. Foi também no final dessa década que se reconheceu a importância da Qualidade de *Software*.

- **Década de 90:** Era da Qualidade. A década de 90 e os anos que seguem podem, certamente, ser chamados de *Era da Qualidade*. Com a tecnologia do estado da arte, espera-se atender a demanda dos clientes com a crescente exigência de alta qualidade.

Qualidade é um termo que pode ter diferentes interpretações e para se estudar a Qualidade de *Software* de maneira efetiva é necessário, inicialmente, obter um consenso em relação à definição de Qualidade de *Software* que está sendo abordada. Existem muitas definições de Qualidade de *Software* propostas na literatura, sob diferentes pontos de vistas, vejamos alguns:

- Qualidade de *Software* é a conformidade a requisitos funcionais e de desempenho que foram explicitamente declarados, a padrões de desenvolvimento claramente documentados, e a características implícitas que são esperadas de todo *software* desenvolvido por profissionais" (Pressman,1994).
- "Um produto de *software* apresenta qualidade dependendo do grau de satisfação das necessidades dos clientes sob todos os aspectos do produto" (Sanders, 1994).
- Qualidade é a totalidade de características e critérios de um produto ou serviço que exercem suas habilidades para satisfazer as necessidades declaradas ou envolvidas "(ISO9126 1994).
- Qualidade é a totalidade das características de uma entidade, que lhe confere a capacidade de satisfazer necessidades explícitas e implícitas (NBR ISO 8402, 1994).

CONCEITO

Existe, ainda, uma visão de Qualidade de *Software* do ponto de vista gerencial, que diz que o *software* que possa ser desenvolvido dentro do prazo e do orçamento especificados pode ser um *software* de alta qualidade. Isso demonstra que, ainda dentro da Qualidade de *Software*, pode-se definir várias visões diferentes, como tem sido para a definição da qualidade como um termo geral.

De um modo geral, Qualidade de *Software* pode ser definida como:

Um conjunto de atributos de *software* que devem ser satisfeitos de modo que o *software* atenda às necessidades do usuário (seja ele um usuário final, um desenvolvedor ou uma organização), onde a determinação dos atributos relevantes para cada *software* varia em função:

- do domínio da aplicação;
- das tecnologias utilizadas;
- das características específicas do projeto;
- das necessidades do usuário e da organização;

Podemos dizer ainda que a qualidade depende também do ponto de vista de quem a avalia, onde usuários, desenvolvedores e organizações podem ter pontos de necessidades diferentes:

- **Usuário:** avalia o *software* sem conhecer seus aspectos internos, está apenas interessado na facilidade do uso, no desempenho, na confiabilidade dos resultados e no preço;
- **Desenvolvedores:** avaliam aspectos de conformidade em relação aos requisitos dos clientes e também aspectos internos do *software*;
- **Organização:** avalia aspectos de conformidade em relação aos requisitos dos clientes e desenvolvedores e também aspectos de custo e cronograma.

1.1.1 Qualidade de Produto X Qualidade de Processo de Software

Qualidade de *Software* pode ser vista como um conjunto de características que devem ser alcançadas em um determinado grau para que o produto atenda as necessidades de seus usuários (Rocha et al.,2001). A qualidade do produto final é profundamente afetada pela qualidade do processo de desenvolvimento, portanto a qualidade deve ser uma meta a ser alcançada e aprimorada ao longo do processo de *software*.

Aqui se faz importante definirmos o que vem a ser um produto de *software*, assim como esclarecermos alguns pontos de um processo de *software*, vamos lá?

1.1.2 Produto de Software

Um produto de *software* compreende os programas e procedimentos de computador e a documentação e dados associados, que foram projetados para serem liberados para o usuário (ISO/IEC 12207-1, 1995).

Da mesma forma como existem diversas interpretações para qualidade de um modo geral, também existem diversas interpretações para qualidade de um produto de *software*, tais como:

- Boa fabricação.
- Deve durar muito.
- Bom desempenho.
- Adaptável às minhas necessidades específicas
- Fácil de usar.
- Sem defeitos, entre outros.

A especificação de Qualidade de Produto de *Software* deve ser mais precisa e detalhada. A formalização de Qualidade de Produto de *Software* pode ser feita usando-se um Modelo de Qualidade de Produto de *Software*.

CONEXÃO

Para compreender um pouco mais sobre nosso assunto, veja o link: http://cbsoft2013.unb.br/wp-content/uploads/2013/10/ST1-2.pdf, um excelente artigo para aumentar nossos conhecimentos. Boa leitura!

Como mesmo as proposições bem sucedidas trazem dificuldades de aplicação, por causa dos muitos aspectos de qualidade oferecidos, surgiu a necessidade de um modelo padronizado. Por essa razão o comitê técnico da ISO/IEC começou a trabalhar para desenvolver o consenso requerido e encorajar a padronização em nível mundial. As primeiras tentativas de padronização surgiram em 1978. Em 1985 foi iniciado o desenvolvimento da Norma Internacional ISO/IEC 9126 – *"Information Technology – Software product evaluation – Quality characteristics and guidelines for thei use"* - publicada em 1991.

A Norma NBR 13596 é uma tradução da Norma ISO/IEC 9126. Foi elaborada pela comissão de Estudos de Qualidade de *Software* do Subcomitê de *Software* do Comitê de Informática da ABNT – Associação Brasileira de Normas Técnicas e publicada em agosto de 1996. A norma avalia a qualidade de um produto de *software* através de um conjunto de características. Essas características são divididas em seis grandes grupos, os quais serão apresentados detalhadamente no capítulo 2.

1.1.3 Processo de Software

Vejamos algumas definições da palavra processo:

- Um processo é "a maneira pela qual se realiza uma operação, segundo determinadas normas" (dicionário Aurélio)
- Um processo é uma seqüência de passos realizados para um dado propósito" (IEEE)
- O processo integra pessoas, ferramentas e procedimentos.
- Processo é o que as pessoas fazem, usando procedimentos, métodos, ferramentas e equipamentos, para transformar matéria prima (entrada) em um produto (saída) que tem valor para o cliente.

Um processo de *software* pode ser definido como um conjunto de atividades, métodos, práticas e transformações que as pessoas usam para desenvolver e manter o *software* e os produtos associados (por exemplo: planos de projeto, documentos, código, casos de teste e manuais de usuário) (IEEE-STD-610).

Um processo de *software* envolve um grande conjunto de elementos, tais como objetivos organizacionais, políticas, pessoas, comprometimentos, ferramentas, métodos, atividades de apoio e as tarefas da engenharia de *software*. Para que o processo de *software* seja eficiente ele precisa ser constantemente avaliado, medido e controlado. Quando as empresas não possuem conhecimento suficiente de todos os elementos do processo, essas atividades de avaliar, medir e controlar ficam difíceis de serem realizadas, nesse caso o processo de *software* passa a ser descontrolado, sem gerência e até mesmo caótico. Algumas características de processos de *software*s caóticos são (Pressman, 2002).

- O processo é improvisado (profissionais e gerentes);
- O processo não é rigorosamente seguido e o cumprimento do mesmo não é controlado;
- O processo é altamente dependente dos profissionais atuais;
- A visão do progresso e da qualidade do processo é baixa;

- A qualidade do produto decorrente do processo é comprometida em função de prazos;
- Quando são impostas datas urgentes para entrega dos produtos, freqüentemente, a funcionalidade e a qualidade dos mesmos são comprometidas para atender o cronograma;
- Não existe nenhuma base objetiva para julgar a qualidade do produto ou para resolver problemas de processo ou de produto, portanto, a qualidade do produto é difícil de ser prevista;
- As atividades ligadas à melhoria da qualidade, tais como revisões e testes, freqüentemente são encurtadas ou eliminadas quando os projetos ultrapassam o cronograma previsto.

Já quando as coisas caminham bem e o processo é controlado e gerenciado com eficiência o processo passa a ser bom, passa a ser maduro e eficiente, gerando resultados positivos, tais como:

- O processo continua a despeito de problemas inesperados (Robustez).
- Rapidez na produção do sistema (Velocidade).
- O processo é aceito por todos os envolvidos nele (Aceitabilidade).
- Os erros do processo são descobertos antes que resultem em erros no produto (Confiabilidade).
- O processo evolui para atender alterações de necessidades organizacionais (Manutenibilidade).
- O processo é compreendido (usualmente através de documentação e de treinamento), utilizado, vivo e ativo.
- O processo é bem controlado – a fidelidade ao processo é objeto de auditoria e de controle.
- Medidas do produto e do processo são utilizadas.
- Os papéis e responsabilidades no processo estão claros ao longo de todo o projeto e por toda a organização.

- A produtividade e a qualidade dos produtos de *software* resultantes podem ser melhoradas com o tempo através de ganhos consistentes na disciplina obtida pelo uso do processo de *software*.
- Os gerentes monitoram a qualidade dos produtos de *software* e a satisfação do cliente.
- Existe uma base quantitativa, objetiva para julgar a qualidade dos produtos e analisar problemas com o produto e o processo.
- As organizações com processo de *software* de alta qualidade tem esse processo institucionalizado através de políticas, padrões e estruturas organizacionais.

Bom pessoal, acredito que tenha ficado claro que para se alcançar os objetivos de produtividade e qualidade é necessário que o processo de *software* seja eficiente, definido, gerenciado, medido e controlado

1.2 Fatores de Qualidade de Software

Existem dois tipos de Qualidade de *Software*: um tipo de qualidade com a qual o usuário do programa interage- essa é a qualidade externa. E um tipo de qualidade com a qual outros desenvolvedores interagem - essa é a qualidade interna, sendo assim podemos dizer te temos os fatores de qualidade interno e os fatores de qualidade externo (Pressman 2002).

Fatores externos – ponto de vista do usuário:

- Correção: característica do *software* que realiza as tarefas como foram definidas em sua especificação de requisitos.
- Robustez: um *software* é robusto se realiza as suas tarefas de forma correta mesmo quando submetido a condições anormais.
- Extensibilidade: característica de um *software* poder ser facilmente adaptado a inclusões e alterações de requisitos.
- Reusabilidade: característica de um *software* que pode ser reutilizado ao todo ou em parte por outros *software*s.

- Compatibilidade: facilidade de se combinar o *software* com outros *softwares*. Essa característica é importante porque raramente um *software* é construído sem interação com outros *softwares*.
- Eficiência: refere-se ao bom uso que o *software* faz dos recursos de hardware, tais como memória e processadores.
- Portabilidade: é a facilidade de se utilizar o *software* em diferentes ambientes de hardware e *software*.
- Verificabilidade: é a facilidade de se preparar rotinas para se verificar a conformidade do *software* com os seus requisitos.
- Integridade: é uma característica relacionada à segurança de dados, programas e documentos. Integridade é a habilidade de proteger tais componentes contra acessos não autorizados.
- Facilidade de uso: também denominada usabilidade, é a facilidade com que o *software* pode ser aprendido e utilizado.

Fatores internos: ponto de vista estrutural do *software*. Permitem atingir os fatores externos:

- Modularidade: característica de um *software* que é constituído por unidades denominadas módulos.
- Legibilidade
- Manutenibilidade: facilidade de realizar manutenção em um *software*.

A forma com um *software* é construído permite atingir os fatores internos de qualidade. Os fatores internos de qualidade permitem atingir os fatores externos de qualidade.

1.3 Métricas de Software

Segundo Tom De Marco, "Não se pode gerenciar o que não se pode medir". Roger Pressmas, 2002, considera que "Se você não sabe para onde você quer ir, qualquer caminho você pode seguir. Se você não sabe onde você está, um mapa não vai ajudar!".

Ainda, segundo Pressman, uma métrica é a medição de um atributo (propriedades ou características) de uma determinada entidade (produto, processo ou recursos). Exemplos:

- Tamanho do produto de *software* (ex: Número de Linhas de código)
- Número de pessoas necessárias para implementar um caso de uso
- Número de defeitos encontrados por fase de desenvolvimento
- Esforço para a realização de uma tarefa
- Tempo para a realização de uma tarefa
- Custo para a realização de uma tarefa
- Grau de satisfação do cliente (ex: adequação do produto ao propósito, conformidade do produto com a especificação).

Mas, acho que aqui seria interessante um questionamento que já ouvi muito de meus alunos: Por que devemos medir o *software*? É realmente importante ou seria uma perda de tempo?

Veremos a seguir algumas vantagens de realizarmos essas medições no *software*:

- Entender e aperfeiçoar o processo de desenvolvimento
- Melhorar a gerência de projetos e o relacionamento com clientes
- Reduzir frustrações e pressões de cronograma
- Gerenciar contratos de *software*
- Indicar a qualidade de um produto de *software*
- Avaliar a produtividade do processo
- Avaliar os benefícios (em termos de produtividade e qualidade) de novos métodos e ferramentas de engenharia de *software*
- Avaliar retorno de investimento
- Identificar as melhores práticas de desenvolvimento de *software*

- Embasar solicitações de novas ferramentas e treinamento
- Avaliar o impacto da variação de um ou mais atributos do produto ou do processo na qualidade e/ou produtividade
- Formar uma *baseline* para estimativas
- Melhorar a exatidão das estimativas
- Oferecer dados qualitativos e quantitativos ao gerenciamento de desenvolvimento de *software*, de forma a realizar melhorias em todo o processo de desenvolvimento de *software*

Para realizarmos as medições precisamos fazer uso de alguma ou algumas métricas, a qual deve ser válida (quantifica o que queremos medir), confiável (produz os mesmos resultados dadas as mesmas condições) e prática (barata, fácil de computar e fácil de interpretar).

1.3.1 Categorização de métricas:

- **Métricas diretas (fundamentais ou básicas):** medida realizada em termos de atributos observados (usualmente determinada pela contagem). Ex.: custo, esforço, no. linhas de código, capacidade de memória, no. páginas, no. diagramas, etc.
- **Métricas indiretas (derivadas):** medidas obtidas a partir de outras métricas. Ex.: complexidade, eficiência, confiabilidade, facilidade de manutenção.
- **Métricas orientadas a tamanho:** são medidas diretas do tamanho dos artefatos de *software* associados ao processo por meio do qual o *software* é desenvolvido. Ex.: esforço, custo, no. KLOC, no. páginas de documentação, no. Erros.
- **Métricas orientadas por função:** consiste em um método para medição de *software* do ponto de vista do usuário, determinando de forma consistente o tamanho e a complexidade de um *software*.
- **Métricas de produtividade:** concentram-se na saída do processo de engenharia de *software*. Ex.: no. de casos de uso/iteração.

LEITURA

Uma boa leitura para aguçar nossa curisosidade com relação à definição de métricas de software seria o material disponível no link a seguir, aproveitem! http://www.portaisgoverno.pe.gov.br/web/metricas-de-software/definicoes

- **Métricas de qualidade:** Oferecem uma indicação de quanto o *software* se adequa às exigências implícitas e explícitas do cliente.Ex.: erros/fase .

- **Métricas técnicas:** concentram-se nas características do *software* e não no processo por meio do qual o *software* foi desenvolvido. Ex.: complexidade lógica e grau de manutenibilidade

1.4 Revisões de Software

De acordo com SEI CMM (3-1.5, 1988), "um processo de revisão pode ser definido como uma avaliação crítica de um objeto (...) assim walkthroughs, inspeções e auditorias podem ser visualizados como formas de processos de revisão."

As revisões são métodos de validação de qualidade de um processo ou produto amplamente usado pela equipe técnica do projeto. São consideradas como verdadeiros "filtros" de erros e inconsistências no processo de desenvolvimento de *software*.

A atividade de revisão começou como uma ferramenta de controle gerencial – Revisão de progresso, onde o progresso não pode ser avaliado simplesmente contando-se o número de tarefas finalizadas.

Era preciso estabelecer um meio de avaliar também a qualidade do trabalho executado, surgiram então as revisões que avaliam aspectos técnicos do produto.

Qualquer produto pode ser submetido a uma revisão aplicada desde as primeiras fases do ciclo de vida. As revisões devem ser fomais e tamb'em informais.

Deve sempre ser realizado um planejamento para a realização das revisões de *software*. Cabe ao engenheiro de *software* planejar:

- o que deve ser revisado
- quais os resultados esperados
- quem deve fazer a revisão,
- determinar "*checkpoints*" dentro do ciclo de vida onde a revisão deve ser aplicada,
- determinar resultados esperados.

Algumas informações importantes no planejamento das revisões são:

- quem participa?
- qual informação é requerida antes da revisão?
- quais pré-condições que devem ser satisfeitas antes que a revisão possa ser conduzida?
- Como Organizar?
- Gerar *checklists* ou outra indicação do que deve ser coberto na revisão;
- Determinar as condições de término ou critérios que devem ser satisfeitos para que a revisão termine;
- Gerar registros e documentos que devem ser produzidos.

As revisões são o principal mecanismo para avaliar o progresso do desenvolvimento de maneira confiável, trazendo vários benefícios para o bom desenvolvimento do *software*, tais como:

- As revisões trazem à luz as capacidades de cada indivíduo envolvido no desenvolvimento,
- revisões são capazes de revelar lotes ou classes de erros de uma só vez;
- revisões proporcionam retorno já nas primeiras fases, prevenindo que erros mais sérios surjam;
- revisões treinam e educam os participantes e
- têm significante efeito positivo na competência dos desenvolvedores.

Quando se realiza revisões e correções desde o início do ciclo de desenvolvimento de *software*, um dos grandes ganhos é com relação aos custos. Vejamos o custo da remoção de erros:

Atividades de projeto são responsáveis por 50 a 65% dos erros, a revisão pode revelar até 75% desses erros e, revelar erros cedo diminui o custo de validação e correção.

- Fase de projeto: custo 1
- Fase anterior ao teste: custo 6.5
- Fase de teste: custo 15
- Fase de manutenção: custo 60 a 100

"Uma má revisão pode ser pior do que nenhuma revisão" , por isso 'e importante considerar as atividades a seguir:

- Determine uma agenda (e mantenha-a)

- Limite os debates
- Levante as áreas problemáticas – não tente resolver todos os problemas
- Tome notas
- Revise o produto, não o produtor
- Limite o número de participantes e insista na preparação;
- Prepare um *checklist*, de acordo com o produto a ser revisado;
- Reserve recursos do projeto para revisões;
- Promova treinamento para os revisores;
- Revise suas antigas revisões

Para fixarmos os conhecimentos apresentados nesse capítulo, a seguir, seguem algumas questões a serem respondidas.

ATIVIDADE

1. Defina Qualidade de Software.

2. Defina processo de software e comente as características de um bom processo de software.

3. Comente sobre a categorização das métricas.

4. Quando falamos de revisões de software, o que é importante que o engenheiro considere no planejamento?

REFLEXÃO

Neste capítulo pudemos conhecer um pouco mais sobre o assunto Qualidade de *Software*, entendendo a sua importância.

A Qualidade de *Software* pode ser subdividida em qualidade de produto e qualidade de processo, sendo as duas extremamente importantes e relacionadas.

Para podermos dizer que um *software* tem qualidade é necessário que o mesmo seja avaliado, o que pode ser realizado com o auxílio das métricas e constantes revisões.

LEITURA

Introdução a Qualidade de Software http://www.ufpa.br/srbo/Disciplinas/Tecnologia-Processos*Software*/Aulas/Qualidade.pdf

Levantamento de métricas de avaliação de projetos de software livre http://ccsl.ime.usp.br/files/relatorioPauloMeirelles_final.pdf

REFERÊNCIAS BIBLIOGRÁFICAS

Métricas e Estimativas de Software – O início de um rally de regularidade. Url: http://www.apinfo.com/artigo44.htm

PRESSMAN, Roger S. **Engenharia de Software**. Rio de Janeiro: MacGraw Hill, 2002.

ROCHA, ANA REGINA CAVALCANTI; MALDONADO, JOSÉ CARLOS; WEBER, KIVAL CHAVES. **Qualidade de Software** – Teoria e Prática. 1ª edição. São Paulo: Prentice Hall, 2001.

SANCHES, ROSELY. Material Didático: **Qualidade de Software**. ICMC-USP, 2003.

SANCHES, ROSELY; FABBRI, SANDRA PINTO; MALDONADO, JOSÉ CARLOS. **Qualidade de Software**: da engenharia de *software* aos modelos de qualidade. VI Escola Regional de Informática, SBC, 2003.

SOMERVILLE, IAN. **Engenharia de Software**. 6ª edição. São Paulo: Addison Wesley, 2003.

NO PRÓXIMO CAPÍTULO

Dando continuidade ao nosso estudo, no próximo capítulo comentaremos sobre a garantia da Qualidade de *Software* (SQA) e discutiremos sobre as normas relacionadas a Qualidade de *Software* ISO/NBR 9126 e ISO/NBR 12119.

2

Normas de Qualidade de *Software*

2 Normas de Qualidade de *Software*

No capítulo anterior vimos os conceitos de qualidade de software e percebemos que o mesmo é extremamente importante, visto que, com a grande demanda de software, os clientes encontram-se cada vez mais exigentes e o mercado cada dia mais competitivo.

Apesar da necessidade da qualidade nos softwares, garantir isso nem sempre é uma tarefa fácil para as empresas desenvolvedoras de software, dentro desse contexto, vamos analisar a atividade SQA – garantia de qualidade de software e as normas de qualidade de software ISO/NBR 9126 e ISO/NBR 12119..

OBJETIVOS

- Descrever os passos necessários para realizar a garantia de qualidade de *software* estatística.
- Discorrer sobre a diferença entre confiabilidade de *software* e segurança de *software*.
- Abordar os princípios da ISO 9000 e suas vertentes.
- Discorrer sobre os modelos ISO 9000 para sistemas de garantia da qualidade.

REFLEXÃO

Você se lembra da definição de qualidade de software que vimos no capítulo anterior?
Na verdade vimos mais de uma!
A qualidade de software possui várias definições, porém todas elas possuem em comum a grande preocupação com o usuário final.
Pensando no usuário final foi que surgiram normas preocupadas em avaliar produtos e pacotes de software, vamos conhecê-las nesse capítulo!

2.1 Garantia de Qualidade de *Software* (Software Quality Assurance – SGA)

Segundo Pressman (2002), *Software Quality Assurance* é um padrão sistemático e planejado de ações que são exigidas para garantir a qualidade de ***software***. Essas ações englobam:

5. Aplicação de Métodos Técnicos: ajudam o analista a conseguir uma especificação de elevada qualidade e o projetista a desenvolver um projeto de elevada qualidade.

6. Realização de Revisões Técnicas Formais: para avaliar a qualidade da especificação e do projeto.

7. Atividades de Teste de Software: para ajudar a garantir uma detecção de erros efetiva.

8. Aplicação de Padrões e Procedimentos Formais no processo de engenharia de software.

9. Processo de Controle de Mudanças: atividade que faz parte do gerenciamento de configuração de software.

10. Mecanismos de Medição: para ser possível rastrear a qualidade de software

11. Anotação e Manutenção de Registros: procedimentos para a coleta e disseminação de informações de garantia de qualidade de software.

2.1.1 Abordagens Formais Para a Garantia de Qualidade de Software (SGA)

As principais abordagens são comentadas a seguir:

1. **Prova de Corretitude —** se o modelo dos requisitos (especificação) e a linguagem de programação podem ser representadas de uma maneira rigorosa, deveria ser possível aplicar provas matemáticas de corretitude para demonstrar que o programa atende exatamente suas especificações.

2. **Garantia Estatística de Qualidade —** implica coletar e categorizar informações sobre os defeitos do software; tentar descobrir a causa de cada defeito; isolar 20% das causas (princípio de Pareto); corrigir os problemas que causaram os defeitos

3. **Proesso Sala Limpa (*Cleanroom*) —** combina a prova de corretitude e a Garantia Estatística de Qualidade para melhorar a qualidade do produto *software*

A necessidade de qualidade de software é reconhecida por praticamente todos os gerentes e profissionais da área, porém muito poucos estão interessados em estabelecer funções de Garantia de Qualidade de Software Formais. Alguma razões para essa aparente contradição:

1. os gerentes relutam em incorrer em custos extras logo de cara
2. os profissionais acham que estão fazendo absolutamente tudo o que precisa ser feito
3. ninguém sabe onde colocar essa função organizacionalmente
4. todos querem evitar a burocracia que, segundo entendem, a Garantia de Qualidade de Software introduzirá no processo de engenharia de software.

Alguns aspectos positivos da garantia de qualidade de ***software*** que podemos mencionar são:

- o *software* terá menos defeitos latentes resultando em redução do esforço e do tempo gasto durante as atividades de teste e manutenção
- a maior confiabilidade resultará em maior satisfação do cliente
- os custos de manutenção podem ser reduzidos
- o custo do ciclo de vida global do software é reduzido

Pressman (2004), destaca alguns passos necessários para realizar a SQA estatística e criar um processo adaptativo de engenharia de *software* no qual são feitas modificações para aprimorar os elementos do processo que promovem erro:

- Coletar e categorizar os defeitos de *software* encontrados.
- Rastrear o defeito até sua causa subjacente.
- Considerar que 20% do código têm 80% dos defeitos.
- Corrigir os problemas que causaram os defeitos.

CONEXÃO

Para sabermos um pouco mais sobre a garantia de qualidade de software seria interessante você ler o artigo: A importância do processo de garantia de qualidade, disponível em: http://www.univicosa.com.br/arquivos_internos/artigos/AImportanciadoProcessodeGarantiadaQualidadedeSoftware.pdf

2.2 Princípios e Modelos da Norma ISO 9000

Conforme já comentamos anteriormente, para se alcançar os objetivos de produtividade e qualidade é necessário que o ***software*** seja eficiente, definido, gerenciado, medido e controlado, o que nem sempre é uma tarefa fácil.

Para auxiliar nessas tarefas existem norma e modelos padrões que possuem como principal objetivo a obtenção da qualidade. O conjunto de Normas da ISO 9000 é um exemplo disso.

A seguir comentaremos sobre duas normas de qualidade, sendo uma voltada para **produto de software** (você se lembra o que é um produto de software, não é?) e outra voltada para a avaliação de **pacotes de software**, ou software de prateleiras, como alguns preferem chamar.

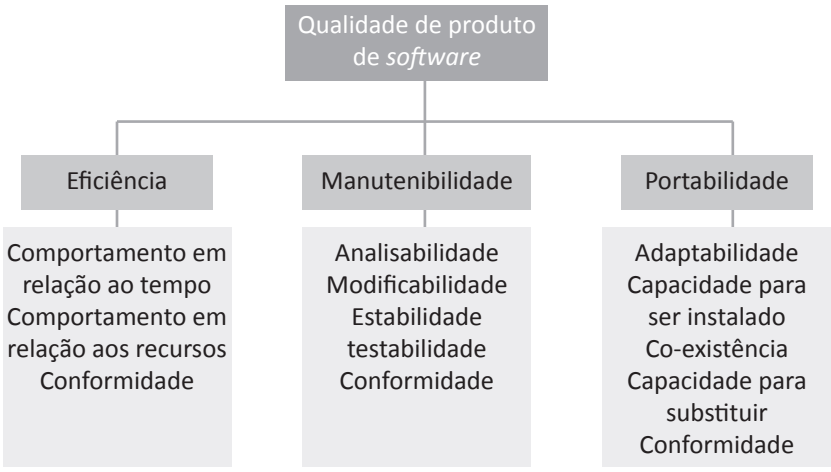
2.3 NBR ISO 9126

Como já comentamos anteriormente, a qualidade é um conceito muito importante para o software. Durante muito tempo, foram propostos modelos de qualidade, mas a confiabilidade era o único meio de se avaliar a qualidade de um software. Surgiu então a necessidade de um modelo padronizado. Por essa razão, o comitê técnico da ISO (*International Organization for Standardization*) e IEC (*International Electrotechnical Comission*), iniciaram em 1985 o desenvolvimento da Norma Internacional ISO/IEC 9126, que estabelece características baseadas no conceito de qualidade para um produto de software.

A Norma ISO/IEC 9126:1991 ou NBR 13596:1996 apresenta a padronização mundial do Software como Produto considerado como um "Software de Qualidade".

Esta norma fornece um modelo de propósito geral o qual define 6 categorias de características de qualidade de software que são, por sua vez, divididas em subcaracterísticas. As subcaracterísticas podem ser avaliadas por um conjunto de métricas.

A ISO/IEC 9126 e foi publicada em 1991, sendo uma das mais antigas da área de qualidade de software e já possui sua tradução para o Brasil, publicada em agosto de 1996 como NBR 13596. Estas normas listam o conjunto de características que devem ser verificadas em um software para que ele seja considerado um "software de qualidade". São seis grandes grupos de características, cada um dividido em algumas subcaracterísticas, os quais são descritos abaixo:



Segundo a ISO/IEC 9126-1, as definições das características e subcaracterísticas de qualidade interna e externa são:

Funcionalidade: capacidade do produto de software de prover funções que atendam às necessidades explícitas e implícitas, quando o software estiver sendo utilizado sob condições específicas.

- Adequação: capacidade do produto de software de prover um conjunto apropriado de funções para tarefas e objetivos do usuário especificados.
- Acurácia: capacidade do produto de software de prover, com o grau de precisão necessário, resultados ou efeitos corretos ou conforme acordados.

- Interoperabilidade: capacidade do produto de software de interagir com um ou mais sistemas especificados.
- Segurança de Acesso: capacidade do produto de software de proteger informações e dados, de forma que pessoas ou sistemas não autorizados não possam lê-los nem modificá-los e que não seja negado o acesso às pessoas ou sistemas autorizados.
- Conformidade: capacidade do produto de software de estar de acordo com normas, convenções ou regulamentações previstas em leis e prescrições similares relacionadas à funcionalidade.

Confiabilidade: capacidade do produto de software de manter um nível de desempenho especificado, quando usado em condições específicas.

- Maturidade: capacidade do produto de software de evitar falhas decorrentes de defeitos no software.
- Tolerância a Falhas: capacidade do produto de manter um nível de desempenho especificado em casos de defeitos no software ou de violação de sua interface especificada.
- Recuperabilidade: capacidade do produto de software de restabelecer seu nível de desempenho especificado e recuperar os dados diretamente afetados no caso de uma falha.
- Conformidade: capacidade do produto de software de estar de acordo com normas, convenções ou regulamentações relacionadas à confiabilidade.

Usabilidade: capacidade do produto de software de ser compreendido, aprendido, operado e atraente ao usuário, quando usado sob condições específicas.

- Inteligibilidade: capacidade do produto de software de possibilitar ao usuário compreender se o software é apropriado e como ele pode ser usado para tarefas e condições de uso específicas.
- Apreensibilidade: capacidade do produto de software de possibilitar ao usuário aprender sua aplicação.

- Operacionalidade: capacidade do produto de software de possibilitar ao usuário operá-lo e controlá-lo.
- Atratividade: capacidade do produto de software de ser atraente ao usuário.
- Conformidade: capacidade do produto de software de estar de acordo com normas, convenções, guias de estilo ou regulamentações relacionadas à usabilidade.

Eficiência: capacidade do produto de software de apresentar desempenho apropriado, relativo à quantidade de recursos usados, sob condições específicas.

- Comportamento em relação ao tempo: capacidade do produto de software de fornecer tempos de resposta e de processamento, além de taxas de transferência, apropriados, quando o software executa suas funções, sob condições estabelecidas.
- Comportamento em relação aos recursos: capacidade do produto de software usar tipos e quantidades apropriados de recursos, quando o software executa suas funções, sob condições estabelecidas.
- Conformidade: capacidade do produto de software de estar de acordo com normas e convenções relacionadas à eficiência.

Manutenibilidade: capacidade do produto de software ser modificado. As modificações podem incluir correções, melhorias ou adaptações do software devido a mudanças no ambiente e nos seus requisitos ou especificações funcionais.

- Analisabilidade: capacidade do produto de software de permitir o diagnóstico de deficiências ou causas de falhas no software, ou a identificação de partes a serem modificadas.
- Modificabilidade: capacidade do produto de software que uma modificação especifica seja implementada.
- Estabilidade: capacidade do produto de software de evitar efeitos inesperados decorrentes de modificações no software.
- Testabilidade: capacidade do produto de software de permitir que o software, quando modificado, seja validado.

- Conformidade: capacidade do produto de software de estar de acordo com normas ou convenções relacionadas à manutenibilidade.
- Portabilidade: capacidade do produto de software de ser transferido de um ambiente para outro.
- Adaptabilidade: capacidade do produto de software de ser adaptado para diferentes ambientes especificados, sem necessidade de aplicação de outras ações ou meios além daqueles fornecidos para essa finalidade pelo software considerado.
- Capacidade de ser instalado: capacidade do produto de software ser instalado em um ambiente especificado.
- Coexistência: capacidade do produto de software de coexistir com outros produtos de software independentes, em um ambiente comum, compartilhando recursos comuns.
- Capacidade para substituir: capacidade do produto de software de ser usado em substituição a outro produto de software especificado, com o mesmo propósito e no mesmo ambiente.
- Conformidade: capacidade do produto de software de estar de acordo com normas ou convenções relacionadas à portabilidade.

Na norma cada característica é refinada em um conjunto de subcaracterísticas e cada subcaracterística é avaliada por um conjunto de métricas. A norma não fornece métricas e nem métodos para medição, pontuação ou julgamento.

- Cada organização pode estabelecer seus próprios modelos de processo de avaliação e métodos para a criação e validação de métricas relacionadas com as características.
- Também é necessário estabelecer níveis de pontuação e critérios específicos para a organização ou para a aplicação.

2.4 NBR ISO 12119

A Norma NBR 12119 foi publicada em 1996 com o objetivo de fornecer subsídios para a avaliação de pacotes de software.

O escopo da norma NBR 12119 refere-se a pacotes de software, na forma oferecida no mercado, e não aos processos de desenvolvimento e fornecimento de *software*.

Vamos definir o que é considerado um pacote de software:

Pacote de Software: trata-se de um produto de software que envolve um conjunto completo e documentado de programas fornecidos a diversos usuários para uma aplicação ou função genérica.

Também conhecido como "software de prateleira".

Um Pacote de Software envolve todos os componentes do produto disponíveis aos usuários, tais como documentação, manual de instruções e guia para instalação.

Exemplos:

- Processadores de Texto
- Planilhas Eletrônicas
- Gerenciadores de Banco de Dados
- Software Gráficos
- Programas para Funções Técnicas ou Científicas
- Programas Utilitários

Os usuários dessa norma são normalmente fornecedores, orgãos de certificação, laboratórios de teste, auditores e compradores/usuários

LEITURA

Como vocês estão podendo perceber, essa norma é bastante extensa.
Para ajudá-los no entendimento da mesma seria interessante a leitura de uma artigo que apresenta um estudo de caso dessa norma.
O artigo - Avaliação da Qualidade de Um Pacote de Software Utilizando a Norma ISO/IEC 12119: Um Estudo de Caso – encontra-se disponível em:
file:///C:/Users/Mayb/Downloads/exemplo-avaliacaopacoteuml.pdf

A norma de avaliação de pacote de software NBR ISO 12119 faz a avaliação do pacote de software como um todo, ou seja, deve ser avaliado a parte externa da caixa do pacote de software e também a parte interna que nesse caso seria o software em si. A avaliação externa é também de supra importância, uma vez que na caixa do software deve conter todas as informações que o cliente necessita saber antes de executar a compra do software.

A norma é dividida em duas partes:

- **Requisitos de Qualidade**, onde apresenta a descrição do produto, manual do usuário e programas e dados (que é o software propriamente dito) e,

- **Instruções para teste**, onde apresenta as principais atividades de teste a serem realizadas, como deve ser feito o registro destes testes e como deve ser elaborado o relatório final dos testes.

A seguir, apresentamos detalhadamente cada uma das etapas da Norma NBR ISO 12119.

Requisitos de qualidade:

- **Descrição do Produto**: Documento expondo as propriedades de um pacote de software, com o principal objetivo de auxiliar os potenciais compradores na avaliação da adequação do produto antes de sua aquisição. A descrição do produto fornece informações sobre a documentação do usuário, programas e, se existirem, sobre os dados.

  A descrição do produto inclui as principais propriedades do pacote e é um documento disponível ao usuário, independente da aquisição do produto. A descrição do produto é dividida em: Requisitos gerais, identificações e indicações, e declarações sobre as características de funcionalidade, confiabilidade, usabilidade, eficiência, manutenibilidade e portabilidade.

  É importante ressaltar aqui que a avaliação sobre essas características é feita com base no conteúdo externo da caixa do pacote do software, quando o cliente ainda não comprou o software. Vamos detalhar com mais cuidado cada uma das partes da Descrição do produto:

Requisitos gerais

A descrição deve ser inteligível, completa, bem organizada e bem apresentada para auxiliar os compradores em potencial na avaliação da adequação do produto às suas necessidades, antes de comprá-lo.

Deve ser livre de inconsistências internas e é interessante que cada termo tenha um único significado.

Identificações e Indicações

O documento de descrição de produto deve possuir uma única identificação, essa indicação do produto deve ter no mínimo o nome do produto e uma versão ou data.

A identificação do fornecedor deve conter o nome e o endereço de, no mínimo, um fornecedor.

As tarefas que podem ser realizadas utilizando o produto devem ser identificadas. A descrição do produto pode fazer referência aos documentos de requisitos com os quais o produto está em conformidade. Nesse caso as edições relevantes devem ser identificadas.

Os requisitos de hardware e software para colocar o produto em uso devem ser especificados, incluindo nomes de fabricantes e identificação do tipo de todos os componentes. Se a descrição do produto faz referências a interfaces com outros produtos, as interfaces ou produtos devem ser identificados.

Todo os itens a serem entregues (componentes físico do produto) devem ser identificados, incluindo todos os documentos impressos e todos os meios de armazenamento de dados. Deve ser declarado se a instalação do produto pode ou não ser conduzida pelo usuário.

Deve ser declarado se o suporte para operação do produto é oferecido ou não. Deve ser declarado se a manutenção é oferecida ou não. Em caso afirmativo deve ser declarado especificamente o que é incluído

Declarações sobre Funcionalidade

A descrição do produto deve fornecer uma visão geral das funções disponíveis para o usuário do produto, os dados necessários e as facilidades oferecidas. Nem toda função disponível para o usuário necessita ser mencionada, e nem todos os detalhes de como uma função é chamada necessitam ser descritos.

Declarações sobre Confiabilidade
A descrição do produto deve incluir informações sobre procedimentos para preservação de dados. Uma declaração do tipo: "é possível fazer backup através de funções do sistema operacional" é suficiente na descrição do produto.

Declarações sobre Confiabilidade
Convém que propriedades adicionais do produto sejam descritas para assegurar sua capacidade funcional. Exemplo: verificar se a entrada é aceitável; proteger contra conseqüências danosas decorrentes de erro de usuário; recuperar erro.

Declarações sobre Usabilidade
Deve ser especificado o tipo de interface com o usuário, como por exemplo, linha de comando, menu, janelas, teclas de função e função de auxílio. Deve ser descrito o conhecimento específico requerido para a aplicação do produto.

Declarações sobre Usabilidade
Se a proteção técnica contra infrações a direitos autorais pode dificultar a usabilidade, então essa proteção deve ser declarada.
Exemplos: proteção técnica contra cópias, datas programadas de expiração de uso, lembretes interativos para pagamento por cópia.

Declarações sobre Usabilidade
A descrição do produto deve incluir dados sobre a eficiência de uso e satisfação de usuário.

Declarações sobre Eficiência
Na descrição do produto podem ser incluídos dados sobre o comportamento do produto em relação ao tempo, tais como tempo de resposta para uma dada função sob condições estabelecidas. Por exemplo, a configuração do sistema

Declarações sobre Manutenibilidade
A descrição do produto pode conter declarações sobre a manutenibilidade do produto.

Declarações sobre Portabilidade
A descrição do produto pode conter declarações sobre a portabilidade do produto.

Manual do usuário: é um conjunto completo de documentos, disponível na forma impressa ou não, que é fornecido para a utilização de um produto, sendo também uma parte integrante do produto. Deve incluir todos os dados necessários para a instalação, para o uso da aplicação e para a manutenção do software produto. Deve-se observar o manual do usuário com relação à completitude, correção, consistência, inteligibilidade, apresentação e organização:

Completitude: O manual deve conter todas as informações necessárias para o uso do produto, tais como estabelecer todas as funções do pacote e procedimentos de instalação.

Correção: A informação apresentada no manual deve estar correta e sem ambigüidade.

Consistência: Deve haver plena coerência entre a documentação no manual e a descrição do produto. Cada termo deve ter um único significado.

Inteligibilidade: A documentação deve ser compreensível pela classe de usuários que desenvolve atividades com o produto, utilizando termos apropriados, exibições gráficas e explicações detalhadas.

Apresentação e Organização: O manual deve ser apresentado de uma forma que facilite uma visão geral de índices e tabelas de conteúdo. Se o documento não está na forma impressa, deve haver indicação de como efetuar a impressão.

! ATENÇÃO

É importante ressaltar que ass características de Funcionalidade, Confiabilidade e Usabilidade são destacadas e devem ser verificadas através do uso do produto. Não há requisitos específicos para os aspectos de Eficiência, Manutenibilidade e Portabilidade, porém se algum desses requisitos estiver declarado na documentação do pacote, eles devem estar em conformidade.

Programas e Dados: são os requisitos de programas e dados que devem estar descritos, caso existam, para o funcionamento do produto. Os requisitos de qualidade para Programas e Dados utilizam as mesmas definições das características de qualidade da norma ISO/IEC 9126.

Funcionalidade: Devem ser verificados os procedimentos para instalação do produto; a presença de todas as funções mencionadas; a execução correta dessas funções; a ausência de contradições entre a descrição do produto e a documentação do usuário.

Confiabilidade: O usuário deve manter o controle do produto, sem corromper ou perder dados, mesmo que a capacidade declarada seja explorada até os limites ou fora deles, se uma entrada incorreta for efetuada, ou ainda se instruções explícitas na documentação forem violadas.

Usabilidade: A comunicação entre o programa e o usuário deve ser de fácil entendimento, através das entradas de dados, mensagens e apresentação dos resultados, utilizando um vocabulário apropriado, representações gráficas e funções de auxílio (help), entre outras

Usabilidade: O programa também deve proporcionar apresentação e organização que facilitem uma visão geral das informações, além de procedimentos operacionais que o auxiliem, por exemplo, a reversão de uma função executada.

Vamos agora analisar a segunda parte da Norma, a parte que se refere às Instruções para teste. Nós já comentamos no início desse tópico, mas como a estrutura dessa norma é bastante extensa, vale a pena relembrar que a parte da Norma de Instruções para testes é dividida em: pré-requisitos de teste, atividades de teste, registro de teste e relatório de teste:

Os pré-requisitos de teste são:

- Presença de itens de produto,
- Presença do sistema necessário e,
- Treinamento (se mencionado na descrição do produto).

As **atividades de teste** consistem em testar se estão de acordo com os requisitos de qualidade:

- Descrição do produto
- Documentação do usuário
- Programas e dados

Os **registros de teste** devem conter informações suficientes para permitir a repetição do teste:

- plano de teste
- casos de teste
- registrar resultados (falhas/sucessos)
- identificar pessoas envolvidas

O **Relatório de testes** deve abordar:

1. Produto
2. Hw/Sw utilizado no teste
3. Documentos usados
4. Resultados dos testes (descrições, documentação, programas e dados)
5. Lista de não conformidades dos requisitos
6. Lista de não conformidade de recomendações
7. Data do término do teste

Para fixarmos melhor os conhecimentos adquiridos até aqui, seguem abaixo algumas questões para serem refletidas.

ATIVIDADE

1. Comente algumas vantagens da realização da garantia de qualidade de software.

2. Explique as subcaracterísticas da característica **funcionalidade** da norma ISO 9126.

3. O que é um pacote de software? Dê alguns exemplos.

4. Quais são os pré-requisitos de testes da norma ISO 12119?

5. O que deve ser testado de acordo com a norma ISO 12119?

REFLEXÃO

O esforço para a elaboração conjunto de normas do Modelo ISO 9000 foi uma grande colaboração para o auxilio na obtenção da qualidade.

As normas vistas nesse capítulo (ISO 9126 para avaliação de produto de software e a ISO 12119 para avaliação de pacote de software). são amplamente reconhecidas e, cada vez mais utilizadas pelas empresas.

Você, como profissional da área de informática, muito provavelmente irá se deparar cem seu ambiente de trabalho com termos e assuntos abordados por essas normas. Espero que os esclarecimentos desse capítulo 2 sejam úteis na sua vida profissional!

LEITURA

Qualidade de software: visão geral
file:///C:/Users/Mayb/Downloads/Aula03_QualidadeSoftware.pdf

Estudo da garantia da qualidade de software utilizando a metodologia de desenvolvimento de sistemas criada por uma instituição financeira
http://www.fatecsp.br/dti/tcc/tcc0034.pdf

Um modelo para avaliação de produtos de software
http://www.cin.ufpe.br/hermano/laps/download/laps-um-modelo-para-avaliacao-de-produtos-de-software.pdf

REFERÊNCIAS BIBLIOGRÁFICAS

CARPINETTI, Luiz Cesar Ribeiro, et al. **Gestão da Qualidade ISO 9001:2008 Princípios e Requisitos**. Editora Atlas. 3ªed 2010.

ISO/IEC 9126. **International Standard**. Information Technology. Software Product

Evaluation. **Quality characteristics and guidelines for their use**. Geneve, 1991.

ISO/IEC 12119. **International Standard**. Information Technology – Software Packages – Quality Requirements and testing. 1994.

NBR ISO/IEC 9126-1: **Engenharia de Software – Qualidade de produto – Parte 1**: Modelo de qualidade. Rio de Janeiro: ABNT, 2003.

PRESSMAN, Roger S. **Engenharia de Software**. Rio de Janeiro: MacGraw Hill, 2002.

SOMERVILLE, IAN. **Engenharia de Software**. 6ª edição. São Paulo: Addison Wesley, 2003.

NO PRÓXIMO CAPÍTULO

Dando continuidade ao nosso estudo das normas de qualidade de software, no próximo capítulo estaremos analisando a Norma ISO Norma ISO 9241, que diz respeito da Usabilidade do Produto e a Norma ISO 14598 que trata da avaliação do produto de Software.

3

Modelos da Norma ISO

3 Modelos da Norma ISO

A qualidade de software pode ser avaliada por diferentes visões, ou seja, pode ser avaliada do ponto de vista do usuário, do desenvolvedor e também da empresa que o comercializa.

Independente da visão, todos concordam que a usabilidade de um software é um fator extremamente importante, um software difícil de usar dificilmente será considerado com qualidade por nenhuma das visões citadas acima.

Nesse capítulo estaremos dando ênfase para a ISO 9241-11 a qual comenta os benefícios de se medir a usabilidade de um produto de software

OBJETIVOS

- Entender a norma ISO/IEC 9241 com foco na parte que destaca as orientações para usabilidade de software.
- Entender o processo de avaliação de produto de software segundo a norma ISO/IEC 14598.

REFLEXÃO

Comentamos no capítulo anterior sobre a qualidade de produto de software e, falando desse assunto, vimos que a usabilidade é uma característica extremamente importante para que um software tenha qualidade.

Ninguém fica satisfeito com um software o qual não consegue usar, o qual possui dificuldades de uso. Devido a sua grande importância, a característica usabilidade é tratada na Norma ISO 9241-11, a qual veremos a seguir.

3.1 Norma ISO 9241 – Usabilidade do Produto

A grande quantidade de computadores que fazem parte do nosso cotidiano não possuem muita razão de existirem se não forem para nos auxiliar, para tornar nossa vida mais fácil, não é verdade?

De nada nos serve um *software* se tivermos grandes dificuldades de usá-lo, se tivermos grandes dificuldades de operá-lo e tirarmos proveito dele em sua aplicação.

Desta forma, a ISO/IEC 9241-11 esclarece os benefícios de medir usabilidade em termos de desempenho e satisfação do usuário, a usabilidade dos computadores depende do contexto de uso e afirma que o nível de usabilidade alcançado dependerá das circunstâncias específicas nas quais o produto é usado (NBR 9241-11, 1998).

Na avaliação de usabilidade de sistemas interativos, o padrão internacional mais comum é a norma ISO 9241. Ela considera mais o ponto de vista do usuário e seu contexto de uso do que as características ergonômicas do produto.

A ISO/IEC 9241 é dividida em 14 partes, a saber:

Parte 1: Introdução Geral
Parte 2: Orientações sobre requisitos da tarefa
Parte 3: Requisitos para apresentação visual
Parte 4: Requisitos para teclado
Parte 5: Requisitos posturais e de layout para posto de trabalho
Parte 6: Requisitos para ambiente
Parte 7: Requisitos para monitores quanto à reflexão
Parte 8: Requisitos para apresentação de cores
Parte 9: Requisitos para outros dispositivos de entrada que não o teclado
Parte 10: Princípios de diálogo
Parte 11: Orientações sobre usabilidade
Parte 12: Apresentação da informação
Parte 13: Orientações ao usuário
Parte 14: Diálogos por menu

CONEXÃO

Um contexto interessante da usabilidade é com relação à web. Veja isso no artigo Teste de usabilidade no website da UniversidadeFederal do Maranhão disponível em http://www.academia.edu/2064036/Teste_de_usabilidade_no_website_da_Universidade_Federal_do_Maranh%C3%A3o

3.2 Norma ISO 9241 parte -11

A parte 11 (1998) desta norma redefine usabilidade como "a capacidade de um produto ser usado por usuários específicos para atingir objetivos específicos com eficácia, eficiência e satisfação em um contexto específico de uso."

Para melhor compreensão desse enunciado, a norma ISO 9241-11 (1998) também esclareceu outros conceitos:

- Usuário - pessoa que interage com o produto.
- Contexto de uso - usuários, tarefas, equipamentos (hardware, software e materiais), ambiente físico e social em que o produto é usado.
- Eficácia - precisão e completeza com que os usuários atingem objetivos específicos, acessando a informação correta ou gerando os resultados esperados.
- Eficiência - precisão e completeza com que os usuários atingem seus objetivos, em relação à quantidade de recursos gastos.
- Satisfação - conforto e aceitabilidade do produto, medidos por meio de métodos subjetivos e/ou objetivos.

A ISO 9241-11 enfatiza que a usabilidade dos computadores é dependente do contexto de uso e que o nível de usabilidade alcançado dependerá das circunstâncias específicas nas quais o produto é usado. O contexto de uso consiste de usuários, tarefas, equipamentos (hardware, software e materiais), e do ambiente físico e social, pois todos esses podem influenciar a usabilidade de um produto dentro de um sistema de trabalho. As medidas de desempenho e satisfação do usuário avaliam o sistema de trabalho como um todo, e, quando um produto é o foco de interesse, estas medidas fornecem informações sobre a usabilidade daquele produto no contexto particular de uso proporcionado pelo restante do sistema de trabalho. Os efeitos das mudanças em outros componentes do sistema de trabalho, tal como: tempo de treinamento do usuário ou melhoria de iluminação, podem também ser medidos pelo desempenho e satisfação do usuário.

A abordagem adotada na NBR 9241-11 tem benefícios que incluem:

- A estrutura pode ser usada para identificar os aspectos de usabilidade e os componentes do contexto de uso aserem considerados no momento da especificação, projeto ou avaliação de usabilidade de um produto.
- O desempenho (eficácia e eficiência) e a satisfação dos usuários podem ser usados para medir o grau em que umproduto é usável em um contexto particular.

- Medidas de desempenho e satisfação dos usuários podem fornecer uma base de comparação da usabilidade relativa de produtos, com diferentes características técnicas, que são usados no mesmo contexto.

- A usabilidade planejada para um produto pode ser definida, documentada e verificada (p.ex. como parte de um plano de qualidade).

Para especificar ou medir usabilidade, são necessárias as seguintes informações:

- uma descrição dos objetivos pretendidos;

- uma descrição dos componentes do contexto de uso incluindo usuários, tarefas, equipamento e ambientes. Estapode ser uma descrição de um contexto existente ou uma especificação dos contextos pretendidos. Os aspectos relevantes do contexto e o nível de detalhes requeridos irão depender do escopo das questões apresentadas. A descrição do contexto precisa ser suficientemente detalhada de modo que aqueles aspectos que possam ter uma influência significativa sobre a usabilidade possam ser reproduzidos;

- valores reais ou desejados de eficácia, eficiência e satisfação para os contextos pretendidos.

Convém que os objetivos de uso de um produto sejam descritos. Objetivos podem ser decompostos em sub-objetivos os quais especificam componentes de um objetivo global e os critérios que irão satisfazer aquele objetivo. Por exemplo, um vendedor de telefones pode ter o objetivo de "Manter pedidos do cliente". Este objetivo global pode então ser decomposto em sub-objetivos como:

- "Fazer registros exatos de todos os pedidos feitos pelos clientes";

- "Fornecer rapidamente informações às dúvidas dos clientes sobre pedidos feitos".

O nível no qual o objetivo global é estabelecido é uma função do limite do sistema de trabalho em consideração e que fornece o contexto de uso. No exemplo acima, o sistema de trabalho em consideração consiste de vendedores recebendo pedidos por telefone.

Contexto de uso

- **Descrição de usuários**:
  As características relevantes dos usuários precisam derdescritas. Elas podem incluir conhecimento, habilidade, experiência, educação, treinamento, atributos físicos e capacidades sensoriais e motoras. Pode ser necessário definir as características de diferentes tipos de usuários, por exemplo, usuários com diferentes níveis de experiência ou desempenhando diferentes funções.

- **Descrição das tarefas**:
  Tarefas são atividades executadas para alcançar um objetivo. Convém que sejam descritas as características das tarefas que podem influenciar a usabilidade, p.ex. a freqüência e a duração de uma tarefa. Uma descrição detalhada das atividades e processos pode ser requisitada se a descrição do contexto for usada como base para o projeto ou avaliação de detalhes da interação com o produto. Isto pode incluir a descrição da alocação de atividades e passos entre os recursos humanos e tecnológicos. As tarefas não devem ser descritas somente em termos das funções ou funcionalidades fornecidas por um produto ou sistema.
  Convém que qualquer descrição das atividades e passos envolvidos no desempenho da tarefa estejam relacionados aos objetivos a serem alcançados. Com o propósito de avaliar a usabilidade, um conjunto de tarefas-chave será tipicamente selecionado para representar aspectos significantes da tarefa global.

- **Descrição dos equipamentos:**
  As características relevantes do equipamento precisam ser descritas. A descrição do hardware, software e dos materiais associados com o computador podem ser um conjunto de produtos (ou componentes do sistema), um ou mais dos quais podem ser o foco da especificação ou avaliação de usabilidade, ou um conjunto de atributos ou características de desempenho do *hardware*, *software* ou outros materiais.

- **Descrição de ambientes:**
  As características relevantes do ambiente físico e social precisam ser descritas. Os aspectos que podem ser necessários descrever incluem

atributos de um amplo ambiente técnico (p.ex. a rede de trabalho local) o ambiente físico (p.ex. local de trabalho, mobiliário), o ambiente atmosférico (p.ex. temperatura, umidade) e o ambiente cultural e social (p.ex. práticas de trabalho, estrutura organizacional e atitudes).

Especificações de características das medidas de uso:

- **Eficácia:**
  Medidas de eficácia relacionadas aos objetivos ou sub-objetivos do usuário quantoa acurácia e completude com que estesobjetivos podem ser alcançados.
  Por exemplo, se o objetivo desejado for reproduzir com acurácia um documento de duas páginas em um formato específico, então a acurácia pode ser especificada ou medida pelo número de erros de ortografia e pelo número de desvios do formato especificado e a completude pelo número de palavras do documento transcrito dividido pelo número de palavras do documento de origem.

- **Eficiência**:
  Medidas de eficiência relacionam o nível de eficácia alcançada ao dispêndio de recursos. Recursos relevantes podem incluir esforço mental ou físico, tempo, custos materiais ou financeiros. Por exemplo, a eficiência humana pode ser medida como eficácia dividida pelo esforço humano, eficiência temporal como eficácia dividida pelo tempo ou eficiência econômica como eficácia dividida pelo custo.
  Se o objetivo desejado for imprimir cópias de um relatório, então a eficiência pode ser especificada ou medida pelo número de cópias usáveis do relatório impresso, dividido pelos recursos gastos na tarefa tal como horas de trabalho, despesas com o processo e materiais consumidos.

- **Satisfação**:
  A satisfação mede a extensão pela qual os usuários estão livres de desconforto e suas atitudes em relação ao uso do produto. A satisfação pode ser especificada e medida pela avaliação subjetiva em escalas de desconforto experimentado, gosto pelo produto, satisfação com o uso do produto ou aceitação da carga de trabalho quando da realização de diferentes

tarefas ou a extensão com os quais objetivos particulares de usabilidade (como eficiência ou capacidade de aprendizado) foram alcançados. Outras medidas de satisfação podem incluir o número de comentários positivos e negativos registrados durante o uso. Informação adicional pode ser obtida através de medidas de longo-termo como as taxas de absenteísmo, observação de sobrecarga ou subcarga de trabalho físico ou cognitivo do usuário, ou de problemas de saúde relatados, ou a freqüência com que os usuários requerem transferência para outro trabalho.

3.3 Norma ISO 14598 (avaliação de produto)

De acordo com a ISO 14598, avaliar a qualidade de um produto de software é:

Verificar, através de técnicas e atividades operacionais, o quanto os requisitos são atendidos. - Tais requisitos, de uma maneira geral, expressam as necessidades explicitadas em termos quantitativos ou qualitativos e apresentam como objetivo a definição das características de um software, a fim de permitir o exame de seu entendimento.

! ATENÇÃO

Vale a pena ressaltar que as responsabilidades de avaliação e de garantia da qualidade devem constar em um plano que vise ao uso e melhoria da tecnologia de avaliação, implementação, transferência e avaliação da tecnologia de avaliação usada e, por fim, o gerenciamento da experiência de avaliar.

Deve-se avaliar a qualidade do produto liberado por diversas razões, tais como:

- Identificar e entender as razões técnicas para as deficiências e limitações do produto, que podem manifestar-se através de problemas operacionais e problemas de manutenção;
- Comparar um produto com outro, mesmo que indiretamente;
- Formular um plano de ação de como fazer o produto de software evoluir.

Um plano de avaliação quantitativa deve conter introdução, objetivos, características da qualidade aplicáveis, lista de prioridades, objetivos da qualida-

de, cronograma, definição de responsabilidades, categorias de medição, uso e análise de dados, relatórios e demais requisitos de interesse.

3.3.1 Etapas da Norma ISO/IEC 14598

A série de normas ISO/IEC 14598 descreve um processo para avaliação de produtos de software, que consiste de quatro passos, conforme a Tabela 3.1. O padrão definido distingue três perspectivas de avaliação: desenvolvedor, adquirente e avaliador.

	ANÁLISE	ESPECIFICAÇÃO	PROJETO	EXECUÇÃO
PROCESSO PARA DESENVOLVEDORES	Definição de requisitos de qualidade e análise de sua exequibilidade	Quantificação dos requisitos de qualidade	Planejamento da avaliação durante o desenvolvimento	Monitoramento da qualidade e controle durante o desenvolvimento
PROCESSO PARA ADQUIRENTES	Estabelecimento do propósito e escopo da avaliação	Definição de métricas externas e medições correspondentes a serem realizadas	Planejar, programar e documentar a avaliação	A avaliação deviria ser realizada, documentada e analisada
PROCESSO PARA AVALIADORES	Descrição dos objetivos da avaliação	Definição do escopo da avaliação e das medições	Documentação dos processos a serem usados pelo avaliador	Obtenção dos resultados a partir da realização de ações de medição e verificação do produto

Tabela 3.1 Definição do processo de avaliação segundo a ISO/IEC 14598

A Norma ISO/IEC 14598 é dividida em seis partes, as quais são apresentadas a seguir:

ISO/IEC 14598-1: (ISO/IEC 14598-1:1999) – Visão geral

Nessa primeira parte é apresentada uma visão geral do processo de avaliação da qualidade dos produtos de software e definida também toda a estrutura de funcionamento da série de normas ISO/IEC 14598.

Fornece-se um estrutura para avaliar a qualidade de quaisquer produtos de *software* e estabelece os requisitos para medição e avaliação de produtos de *software*.

Para avaliar a qualidade do software, primeiro se estabelecem os requisitos da avaliação, então se especifica, projeta e executa a avaliação, como mostra a Figura 1.

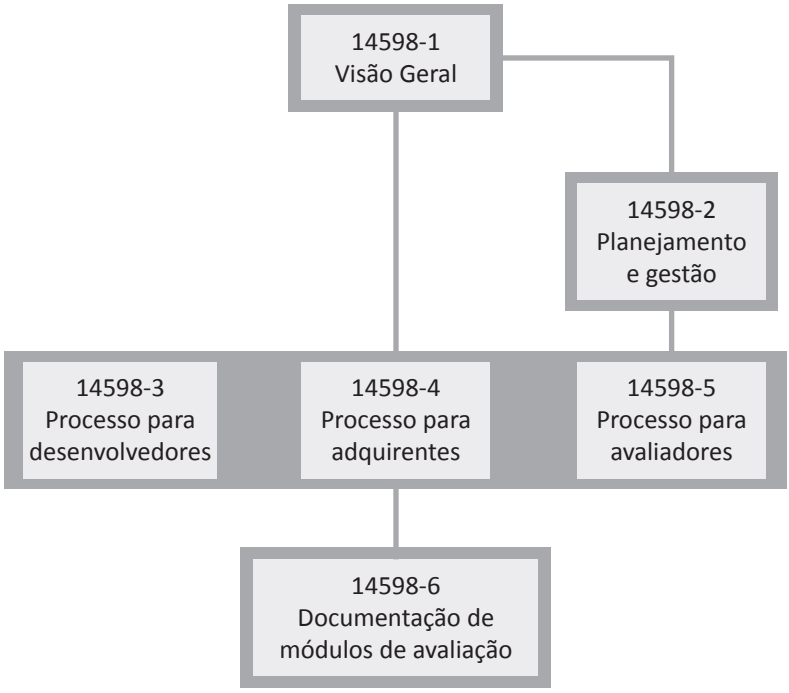


Figura 1 – Estrutura da série de normas ISO/IEC 14598

Para avaliar a qualidade do software, primeiro se estabelecem os requisitos da avaliação, então se especifica, projeta e executa a avaliação, como mostra a Figura 2.

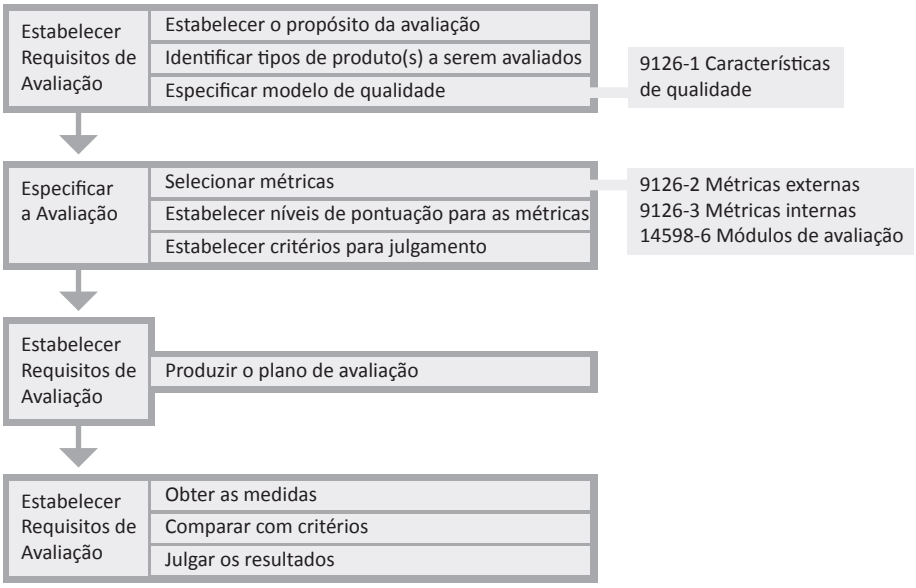


Figura 2 – Processo de avaliação segundo a ISO/IEC 14598-1

Abaixo serão descritas cada uma das etapas do processo de avaliação segundo a ISO/IEC 14598:

Estabelecer requisitos de avaliação: visa levantar os requisitos gerais da avaliação.

- Estabelecer o propósito da avaliação: define quais os objetivos da avaliação. Tais objetivos estão relacionados ao uso pretendido do produto de software e aos riscos associados. Podem ser definidos pontos de vista diferentes de vários usuários do produto, tais como: adquirente, fornecedor, desenvolvedor, operador ou mantenedor do produto.

- Identificar tipos de produto(s) a serem avaliados: define o tipo de produto a ser avaliado, se são um dos produtos intermediários ou o produto final.

- Especificar modelo de qualidade: a primeira etapa na avaliação de software consiste em selecionar as características de qualidade relevantes, utili-

zando um modelo de qualidade que desdobre a qualidade de *software* em diferentes características. Nesta fase da avaliação é escolhido o modelo de qualidade a ser utilizado visando definir os requisitos de qualidade para o produto de software.

Especificar a avaliação: define a abrangência da avaliação e das medições a serem realizadas sobre o produto submetido para avaliação.

- Selecionar métricas: a forma pela qual as características de qualidade tem sido definidas não permite sua medição direta. É necessário estabelecer métricas que se correlacionem às características do produto de software. Neta fase da avaliação são selecionadas as métricas a serem utilizadas durante a avaliação.
- Estabelecer níveis de pontuação para as métricas: para cada métrica selecionada devem ser definidos os níveis de pontuação e uma escala relacionada, onde poderão ser representados o nível planejado, o nível atual e o nível que representa o pior caso para o atributo a ser medido.
- Estabelecer critérios para julgamento: para julgar a qualidade do produto, o resultado da avaliação de cada característica precisa ser sintetizado. É aconselhável que o avaliador prepare um procedimento para isto, onde cada característica poderá ser representada em termos de suas subcaracterísticas ou de uma combinação ponderada de suas subcaracterísticas.

Projetar a avaliação: deve documentar os procedimentos a serem utilizados pelo avaliador para realizar as medições contidas na especificação de avaliação.

- Produzir plano de avaliação: o avaliador deve produzir um plano de avaliação que descreva os recursos necessários para realizar a avaliação especificada, bem como a distribuição destes recursos entre as diversas ações a serem realizadas.

Executar a avaliação: obter os resultados da execução das ações de medição e verificação do produto de software de acordo com os requisitos de avaliação, como especificado na especificação de avaliação e planejado no plano de avaliação.

- Obter as medidas: as métricas selecionadas são aplicadas ao produto de software obtendo como resultado valores nas escalas das métricas.
- Comparar com critérios: o valor medido para cada métrica é comparado com os critérios determinados na especificação da avaliação.
- Julgar os resultados: o julgamento é a etapa final da avaliação, onde um conjunto de níveis pontuados é resumido. O resultado é uma declaração de quanto o produto de software atende aos requisitos de qualidade.

ISO/IEC 14598-2 (ISO/IEC 14598-2:2000): Planejamento e gestão
Aqui apresenta-se o planejamento e gestão do processo de avaliação apresentando requisitos, recomendações e orientações para uma função de suporte ao processo.

ISO/IEC 14598-3 (ISO/IEC 14598-3:2000): Processo para desenvolvedores
Na terceira parte da norma é definido o processo para desenvolvedores. Destina-se ao uso durante o processo de desenvolvimento e manutenção de software.

ISO/IEC 14598-4 (ISO/IEC 14598-4:1999): Processo para adquirentes:
Nesta parte da norma define-se o processo para adquirentes, estabelecendo um processo sistemático para avaliação de produtos de software tipo pacote (com equivalência a NBR ISO/IEC 12119) [15], produtos de software sob encomenda, ou ainda modificações em produtos já existentes.

Ao referenciar o processo de aquisição definido pela Norma ISO/IEC 12119 ressalta a existência das seguintes atividades:

- **Iniciação**: o adquirente inicia o processo de aquisição ao considerar a necessidade de obter, desenvolver ou melhorar um sistema, produto ou serviço de software.
- **Preparação do pedido de proposta**: Elaboração dos documentos de requisitos de aquisição bem como determinados os pontos de controle para revisão e auditoria do progresso do fornecimento.
- **Preparação e atualização do controle**: Seleção de fornecedores e suas capacidades firmadas em contrato e, negociado o custo, cronograma de execução. As mudanças no contrato devem ser controladas pelo adquirente.

- **Monitoração do fornecedor**: Consiste em atividades de avaliação durante a execução do contrato levando à aceitação e entrega do produto de software.

- **Aceitação e conclusão**: Aceitação e entrega do produto de software final, obedecidos aos critérios de aceitação previamente definidos durante a atividade de iniciação.

ISO/IEC 14598-5 (ISO/IEC 14598-5:1998): Processo para avaliadores:
Define o processo para avaliadores, fornecendo orientações para a implementação prática de avaliação de produtos de software (quando diversas partes necessitam entender, aceitar e confiar em resultados da avaliação). Conta com a participação de avaliadores de laboratório, fornecedores e adquirentes de software, usuários e entidades certificadoras que defendem objetivos diferentes.

Os avaliadores podem ser especificamente os laboratórios de testes, as equipes de testes integradas de organizações de produção ou de distribuição de software, os compradores ou usuários de software ou de integradoras de sistemas ou ainda as organizações que realizam comparações de softwares.

Nessa etapa da norma são gerados alguns documentos, os quais passam a fazer parte do produto de software, tais como documentos de projetos, relatórios de testes e avaliação, código fonte e documentação do usuário.

As atividades do processo de avaliação são as seguintes:

- **Estabelecimento de requisitos da avaliação:** Descrição dos objetivos da avaliação coerentes com o produto de software e possíveis riscos associados. As percepções dos usuários do produto, fornecedores, compradores, desenvolvedores, operadores e mantenedores do produto devem ser levados em consideração.

- **Especificação da avaliação:** Definição do escopo da avaliação e as medições a que o produto será submetido. Dependente dos requisitos de avaliação previamente definidos pelo requisitante.

- **Projeto de avaliação:** Preparo de um plano de ação de acordo com a especificação do avaliador e com base nos métodos a serem usados para a realização das medições estabelecidas na especificação. Um documento guarda, portanto, as especificações da avaliação, o cronograma, os recursos necessários e disponíveis para realizar a avaliação.

- **Execução da avaliação:** Consiste na inspeção, medição e teste dos produtos e de seus componentes aplicados de acordo com as orientações do plano. Aplicadas as ações de medição, segue-se para as interpretações dos resultados registrados e depurados durante a avaliação. Ao final da execução, uma versão preliminar é gerada do relatório de avaliação.

- **Conclusão da avaliação:** Revisão do relatório de avaliação e liberação dos dados de avaliação, bem como a devolução, pelo avaliador, do produto avaliado e seus componentes.

Ainda na parte 5 da norma, as características esperadas do processo de avaliação são :

- **Repetitividade:** Avaliação repetida de um mesmo produto, com mesma especificação de avaliação, realizada pelo mesmo avaliador, deve produzir resultados que podem ser aceitos como idênticos.

- **Reprodutividade:** A avaliação do mesmo produto, com mesma especificação de avaliação, realizada por um avaliador diferente, deve produzir resultados que podem ser aceitos como idênticos.

- **Imparcialidade:** A avaliação não deve ser influenciada frente a nenhum resultado particular.

- **Objetividade:** A avaliação não deve ser influenciada frente a nenhum resultado particular.

O processo de avaliação proposto pela norma NBR ISO/IEC 14598-5 é semelhante ao da parte 1, incluindo uma etapa a mais para as etapas da avaliação, a saber: Conclusão da avaliação. Esta etapa será descrita abaixo.

Conclusão da avaliação: nesta etapa, deve-se revisar o relatório da avaliação e disponibilizar os dados resultantes da mesma para o requisitante da avaliação. Para uma avaliação profissional, esses dados são sigilosos e exclusivos ao requisitante da avaliação, qualquer publicação deles é de sua inteira responsabilidade.

CONEXÃO

O artigo o uso da norma 14598 na avaliação de software com relação à qualidade disponível em http://www.waltenomartins.com.br/intercursos_v8n1.pdf é uma leitura interessante para complementar nossa discussão sobre esse assunto!

ISO/IEC 14598-6 (ISO/IEC 14598-6:2001): Documentação de módulos para avaliação:

- Fornece orientação para documentação de módulos de avaliação. Estes módulos contêm a especificação do modelo de qualidade, as informações e dados relativos à aplicação prevista do modelo e informações sobre a real aplicação do modelo.

Após a finalização das etapas da Norma ISO/IEC 14598 no próximo tópico seguem algumas atividades sugeridas.

ATIVIDADE

6. Comente uma das vantagens de se usar a Norma ISO 9241?

7. Como a ISO 9241-11 redefine usabilidade?

8. De acordo com a ISO 14598, o que significa avaliar a qualidade de um produto de software?

9. Você se lembra de quantas e quais são as partes da ISO/IEC 9241? São várias, mas faça um esforço.

REFLEXÃO

Com o estudo desse capítulo podemos entender a norma ISO/IEC 9241, onde foi dado especial atenção às orientações para usabilidade de software, devido a sua grande importância na qualidade de um software.

Pudemos ver também o funcionamento do processo de avaliação de produto de software segundo a norma ISO/IEC 14598, a qual faz referência ao processo de aquisição definido pela ISO/IEC 12207.

A avaliação de um produto de software envolve vários fatores e características, aqui demos ênfase a usabilidade, o que é fácil de entender, uma vez que ninguém quer ter um software que não consegue usar, não é mesmo?!

LEITURA

Guia para utilização das normas sobre avaliação de qualidade de produto de software ISO/IEC 9126 e ISO/IEC 14598 – disponível em: http://www2.dem.inpe.br/ijar/GuiaUtilNormTec.pdfhttp://www2.dem.inpe.br/ijar/

Utilizando a norma IS/IEC 14598-5 na avaliação de qualidade de hiperdo-cu-mentos web – disponível em: http://www.comp.ita.br/~cunha/download/CES-32CE230-2002/Sem11/1_AS3-2%20Utilizando%20a%20Norma%20ISO-IEC%20 14598-5%20na%20Avaliacao%20de%20Qualidade%20de%20Hiperdocumentos%20 Web.pdf

REFERÊNCIAS BIBLIOGRÁFICAS

(ISO/IEC 14598-1:1999), **International Standard. Information Technology** – Software Product Evaluation - Part 1: General Overview.

(ISO/IEC 14598-2:2000), **International Standard. Information Technology** – Software Product Evaluation - Part 2: Planning and Management.

(ISO/IEC 14598-3:2000), **International Standard. Information Technology** – Softwa-

re Product Evaluation - Part 3: Process for Developers.

(ISO/IEC 14598-4:1999), **International Standard. Information Technology** – Software Product Evaluation - Part 4: Process for Acquirers.

(ISO/IEC 14598-5:1998), **International Standard. Information Technology** – Software Product Evaluation - Part 5: Process for Evaluators.

(ISO/IEC 14598-6:2001), **International Standard. Information Technology** – Software Product Evaluation - Part 6: Evaluation Modules.

NO PRÓXIMO CAPÍTULO

Dando continuidade ao nosso estudo sobre qualidade de software, no próximo capítulo veremos alguns modelos de melhoria de processo de software.

4

Modelos de Melhoria e Avaliação de Processo de *Software*

4 Modelos de Melhoria e Avaliação de Processo de *Software*

Desde o início desta disciplina estamos falando da grande importância de se ter qualidade em um produto de software, falamos também que a obtenção da qualidade nem sempre é uma tarefa fácil, na verdade ela é bem complexa, pois um envolve uma grande quantidade de fatores a serem avaliados, medidos e melhorados continuamente.

Nesse capítulo iremos comentar sobre modelos que compõem a melhoria do processo de software.

OBJETIVOS

- Abordar o modelo de garantia da qualidade em produção e instalação, de acordo com a ISO 9000-3;
- Compreender as etapas dos Modelos PDCA e IDEA;
- Descrever os objetivos da Norma NBR ISO/IEC 12207;
- Apresentar os critérios para definição dos seguintes processos na Norma NBR ISO/IEC 12207;
- Processos fundamentais;
- Processos de apoio;
- Processos organizacionais;
- Processos de adaptação.

REFLEXÃO

Você se lembra o que vem a ser um processo de software?

Bom, vamos relembrar juntos:

Um processo de software pode ser definido como um conjunto de atividades, métodos, práticas e transformações que as pessoas usam para desenvolver e manter o software e os produtos associados (por exemplo: planos de projeto, documentos, código, casos de teste e manuais de usuário). (IEEE-STD-610).

Já comentamos que a qualidade de um produto de software está ligada diretamente a qualidade do seu processo de software. Sendo assim, vamos agora ver alguns modelos que nos ajudam a entender como avaliar e melhorar de forma contínua um processo de software.

4.1 Melhoria de Processo de Software

Para a obtenção de um produto de software de qualidade é extremamente importante que tomemos cuidado com o processo desse software.

Já vimos que um processo de software caótico possui muitas características ruins e, que podem influenciar diretamente na qualidade do produto final do software.

CONEXÃO

Antes de definirmos a melhoria de processo de acordo com a ISO 9000-3 sugiro a vocês a leitura do artigo Melhoria de processo de software brasileiro disponível em http://www.devmedia.com.br/melhoria-do-processo-de-software-brasileiro/29915

Vimos também que um processo de software maduro trás muitos benefícios para todos os envolvidos no seu desenvolvimento. Porém, a obtenção de um processo de software maduro nem sempre é uma tarefa fácil para as empresas. Para auxiliá-las nessa função, temos na literatura vários modelos que tem como objetivo avaliar e melhorar continuamente um processo de software.

Vamos começar definindo o que vem a ser melhoria de processo de software.

De acordo com a ISO 9000-3, melhoria de processo consiste a abordagem na prática de ações orientadas para a alteração dos processos aplicados para aquisição, fornecimento, desenvolvimento, manutenção e/ou suporte de sistemas de software.

Vamos relembrar, entende-se por um processo de software maduro, capaz, aquele que é executado de forma eficiente e eficaz, com a presença de alguma características importantes, tais como:

- Execução consistente e sempre que necessária.
- Flexibilidade no processo de execução para melhor adaptação das necessidades específicas.

- Documentação com uma notação representativa de processo identificado por meio de texto, figuras, fluxos, etc.
- Deve ser apropriado para que as pessoas possam realizar o trabalho.
- Treinamento para as pessoas inseridas nas atividades do processo de forma a obterem conhecimento, habilidade e experiência.
- Manutenção para garantir a evolução contínua.
- Controle de mudanças para garantir a integridade e disponibilidade dos artefatos.
- Apoio de equipes, ferramentas e recursos apropriados para realização do processo.

As organizações com processo de software de alta qualidade tem esse processo institucionalizado através de políticas, padrões e estruturas organizacionais. A Institucionalização acarreta construir uma infraestrutura (processos eficazes, utilizáveis e consistentemente aplicados em toda organização) e uma cultura corporativa que apóia os métodos, práticas e procedimentos dos negócios para que eles perdurem mesmo depois que aqueles que originariamente os definiram tenham saído da organização.

4.2 Modelos de Melhoria de Processo de Software

Um modelo de melhoria deve permitir que se compreenda o estado atual do processo, que se desenvolva a visão do processo desejado, que se estabeleça uma lista de ações necessárias à melhoria do processo, em ordem de prioridade e que se gere um plano para acompanhar as ações e alocação de recursos para a execução do plano.

Os modelos de melhoria de processo de software ajudam a empresa a se organizar e estabelecer uma ordem de prioridades de tal forma que a qualidade possa efetivamente ser "construída" ao longo do processo e fazer parte, inerentemente, do produto gerado.

Em uma melhoria de processo de software, o processo passa por uma avaliação, os resultados dessa avaliação conduzem à melhorias, as quais mostram mudanças que devem ser realizados na processo e, isso deve ser feito continuamente.

A capacidade dos processos de software de uma organização influencia na sua capacidade de atingir metas de custo, qualidade e cronograma.

Os modelos de melhoria de processo de software seguem estratégias de melhoria de processo que ajudam a coordenar os esforços de melhoria contínua. Uma estratégia de melhoria de processo permite que:

- se compreenda o estado atual do processo,
- se desenvolva a visão do processo desejado,
- se estabeleça uma lista de ações necessárias à melhoria do processo, em ordem de prioridade, e
- que se gere um plano para acompanhar as ações e alocação de recursos para a execução do plano.

Como exemplos de estratégias de melhoria de processo, podemos citar o ciclo de melhoria PDCA e o modelo IDEAL.

4.2.1 Ciclo de Melhoria PDCA.

O ciclo PDCA é um método gerencial de tomada de decisões para garantir o alcance das metas necessárias à sobrevivência de uma organização. Os conceitos do ciclo PDCA foram originalmente desenvolvidos por Walter Shewart, um estatístico pioneiro que desenvolveu o controle estatístico nos Laboratórios da Bell nos Estados Unidos, na década de 30. A eficiência do ciclo se tornou famosa, na década de 50 quando surgiu, no contexto de Gerenciamento de Qualidade, a grande autoridade, W. Eduard Deming.

O uso do PDCA ajuda a coordenar os esforços de melhoria contínua e demonstra que programas de melhoria devem iniciar sempre com:

- um planejamento cuidadoso,
- que devem resultar em ações efetivas que, por sua vez,
- devem levar ao planejamento,
- fechando, assim, um ciclo contínuo.

Etapa de Planejamento (P)

Nessa etapa as metas da organização são estabelecidas e são desenvolvidos métodos para alcançar essas metas. Isso significa selecionar as operações que são causadoras de problemas e desenvolver idéias para resolver esses problemas.

Metas para Manter: também conhecidas como metas padrão- representam uma faixa aceitável de valores que devem ser mantidos pelas características consideradas importantes no produto.

Metas para Melhorar: também conhecidas como metas de melhoria surgem do fato de que o mercado sempre deseja um produto cada vez melhor, a um custo cada vez mais baixo e com entrega cada vez mais precisa.
O surgimento de novos concorrentes no mercado, novas tecnologias, novos materiais também levam a necessidade de estabelecer metas de melhoria.
Atingir uma meta de melhoria implica em modificar a atual condição de trabalho.

Etapa de Execução (Do)
Nessa etapa as tarefas que foram previstas na etapa de planejamento são executadas e coletam-se dados que serão utilizados na próxima etapa de verificação do processo. É importante que as mudanças sejam implementadas em escalas menores para primeiro testar sua eficiência para que os erros não causem grandes catástrofes ao processo já em andamento. Assim, pode-se verificar se as mudanças funcionam bem ou não. São abordados a educação (disseminação de novos conceitos e tecnologia) e o treinamento (aperfeiçoamento de conceitos já conhecidos).

Etapa de Verificação (*Check*)
Nessa etapa é realizada uma comparação entre os dados coletados na etapa de Execução e as metas definidas na etapa de Planejamento. Para isso, verifica-se se a escala menor de implementação ou mudanças experimentais alcançaram os resultados esperados

Etapa de Ação (A)
Essa etapa consiste em atuar no processo em função dos resultados obtidos. Se o sucesso é obtido em uma escala menor, pode significar que a mudança pode fazer parte das atividades da organização, envolvendo outras pessoas, outros departamentos, fornecedores e clientes afetados pelas mudanças.

LEITURA

Para mais informações sobre o Modelo PDCA, leia o artigo PDCA: origem, conceitos e variantes dessa idéia de 70 anos Disponível em: http://www.qualypro.com.br/artigos/pdca-origem-conceitos-e-variantes-dessa-ideia-de-70-anos#sthash.m2YGAatn.dpuf

O envolvimento de outras pessoas é importante para ajudar na implementação das mudanças numa escala maior, ou mesmo para a conscientização dos novos benefícios que podem adquirir com as mudanças. Existem duas formas de atuação possíveis:

- adotar o plano como padrão, caso a meta tenha sido alcançada
- agir sobre as causas que não permitiram que as metas sejam efetivas.

4.2.2 Modelo IDEAL

O Modelo IDEAL foi desenvolvido pela SEI (Software Engineering Institute). IDEAL é um acrônimo que engloba os 5 estágios do ciclo de melhoria de processo de software, o o qual contém as atividades de diagnóstico do processo atual, a elaboração de um plano para melhoria, a implantação deste plano e a confirmação das atividades implantadas através do diagnóstico do processo, iniciando novamente o ciclo. (Gremba, 1997).

(I nitiating) - Inicialização
(D iagnosing)- Diagnóstico
(E stablishing) - Estabelecimento
(A cting) - Ação
(L everaging) - Lições

O modelo estabelece um programa de melhoria contínua de processo de software (Figura 3) que ajuda as organizações a melhorar o seu processo de software.

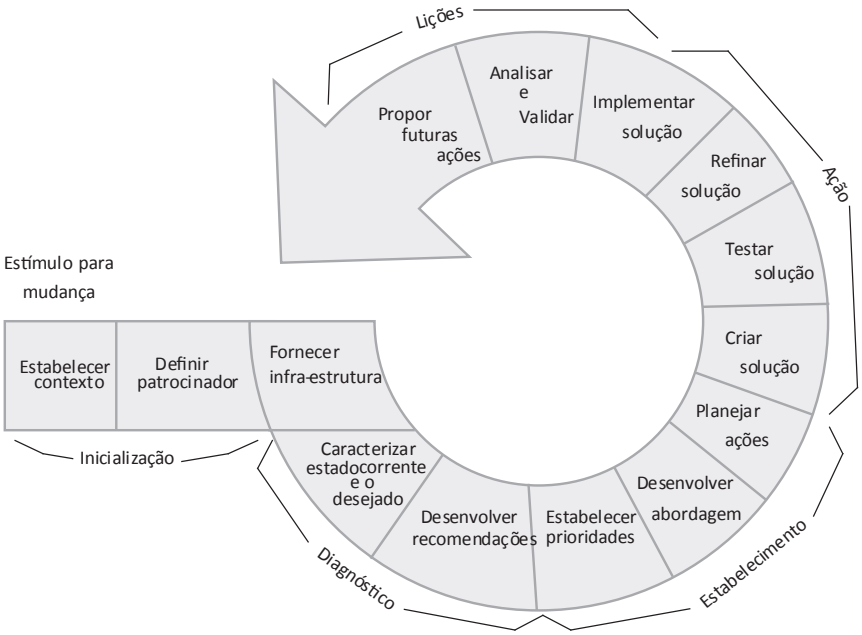


Figura 3 – Abordagem IDEAL (IDEAL, 1998).

Fase de Inicialização: Nessa fase são identificados os motivos e as razões para mudanças no processo de software, os quais necessitam ser extremamente claros e evidentes.

Essa fase envolve:

- **Estabelecer o contexto** que significa definir, claramente, onde os esforços vão se enquadrar na estratégia de negócio da organização, definir quais metas e objetivos de negócio serão afetados pelas mudanças, definir como isso vai incidir sobre outras iniciativas e trabalhos futuros e definir os benefícios esperados.

- **Definir o patrocinador**: um patrocinador efetivo é um dos fatores mais importantes para os esforços de melhoria de processo. O patrocinador deve ajudar a manter o compromisso nos momentos de dificuldades e principalmente na situação de caos que a organização pode se encontrar no princípio do processo de melhoria. Outra função do patrocinador é garantir os recursos essenciais utilizados durante a melhoria.

- **Fornecer infra-estrutura**: após definir as razões para mudança, o contexto e o comprometimento dos patrocinadores, é importante definir o mecanismo para gerenciar os esforços para os detalhes de implementação: a infra-estrutura

Fase de Diagnóstico: Nessa fase duas características da organização são consideradas

- Estado atual do processo de software da organização e o estado futuro desejado; esses estados serão utilizados para desenvolver as práticas de melhoria de processo. Caracterizar o estado corrente e o desejado é como identificar o início e o fim da jornada. Uma vez identificado o estado atual, pode-se usar algum modelo de processo (por exemplo o CMM) para definir o estado desejado.

- Desenvolvimento de recomendações: nessa atividade, desenvolve-se recomendações, sugerindo meios de como proceder em atividades subseqüentes. As atividades da fase de Diagnóstico são desempenhadas pelas pessoas mais experientes ou por especialistas relevantes na organização. Suas recomendações são baseadas nas decisões realizadas pelos gerentes chefes e patrocinadores.

Fase de Estabelecimento: Essa fase possui três atividade a serem realizadas:

- **Estabelecer prioridades**: essa prioridade deve considerar vários fatores, tais como as limitação de recursos, dependências entre as atividades recomendadas, intervenção de fatores externos e as prioridades globais da organização.

- **Desenvolver uma abordagem** - envolve o desenvolvimento de uma estratégia de acompanhamento do trabalho considerando o escopo do trabalho, o conjunto de prioridades e a disponibilidade dos recursos

- **Planejar ações:** Com a abordagem definida, pode-se desenvolver um plano detalhado de implementação. Esse plano inclui cronograma, tarefas, prazos, pontos de decisão, recursos, responsabilidades, medidas, mecanismo de acompanhamento , riscos e mitos, estratégias, outros elementos requeridos pela organização para o processo de melhoria contínua

Fase de Ação: As atividades dessa fase ajudam a organização a implementar o trabalho que foi contextualizado e planejado nas três fases anteriores. Essas atividades tipicamente consomem mais tempo e recursos que todas as outras combinadas. A fase de Ação começa associando todos os elementos chave para **criar a melhor solução** imaginável destinada às necessidades previamente identificadas da organização. Uma vez criada a solução, ela deve ser **testada**, pois por mais que a solução seja bem elaborada ela raramente funciona como o planejado. Após a solução ter sido testada, ela pode ser **modificada** para refletir o conhecimento, a experiência e as lições que foram obtidos pelos testes. Nessa etapa, pode-se **implementar** a solução por toda a organização, uma vez que ela já esteja preparada. Para a implementação podem ser utilizadas as abordagenstop-down (começando pela parte superior da organização) ou a abordagem just-in-time (implementando projeto por projeto em seu tempo apropriado).

Fase de Lições: Nessa fase as experiências são **revisadas**, para **analisar** o que foi realizado e como a organização pode implementar mudanças mais efetivamente, no futuro.
As lições são coletadas, analisadas e documentadas, as necessidades identificadas na fase de inicialização são reexaminadas para ver se foram atendidas e são fornecidas propostas de alterações para **melhoria futura**.

Após nossa discussão sobre os Modelos de Melhoria PDCA e IDEAL, a seguir vamos nos ater a algumas normas ISO, como a NBR/ISO 9000-3 e a NBR/ISO 12207, sendo a primeira define diretrizes para facilitar a aplicação da norma ISO 9001 a organizações que desenvolvem, fornecem e mantém software.por serem normas voltadas a melhoria de processo e, a segunda estabelece uma estrutura comum para os processos de ciclo de vida de software como forma de ajudar as organizações a compreenderem todos os componentes presentes na aquisição e fornecimento de software e, assim, conseguirem firmar contratos e executarem projetos de forma eficaz.

4.3 NBR/ISO 9000-3

A NBR/ISO 9000 foi desenvolvida para ajudar organizações na implementação e operação de sistemas da qualidade eficazes. A ISO 9000 é uma série de normas internacionais para gestão da qualidade e garantia da qualidade. Ela auxilia a empresa na seleção da norma mais apropriada par o seu negócio e a sua utilização.

Ela não é destinada a um produto nem para uma alguma empresa específica. Tem como objetivo orientar a implantação de sistemas de qualidade nas organizações.

A série é composta das seguintes normas:

- ISO 9000- Fundamentos e vocabulário
- ISO 9001- Sistemas de gerenciamento da qualidade – requisitos
- ISO 9004 – Sistemas de gerenciamento da qualidade – guia para melhoria da performance
- ISO 19011- Auditorias internas da qualidade e ambiental

Para facilitar a aplicação em desenvolvimento de software a ISO 9000-3, que são orientações para a aplicação da ISO 9001 ao projeto, desenvolvimento, fornecimento, instalação e manutenção de software. A norma define diretrizes para facilitar a aplicação da norma ISO 9001 a organizações que desenvolvem, fornecem e mantém software. Destina-se a fornecer orientação quando um contrato entre duas partes exigir a demonstração da capacidade do fornecedor em desenvolver, fornecer e manter produtos de software.

De fato, a ISO 9000-3 é um guia para a aplicação da ISO 9001 para o desenvolvimento, fornecimento e manutenção de software. As diretrizes propostas na ISO 9000-3 cobrem questões como:

- Entendimento dos requisitos funcionais entre contratante e contratado;
- Uso de metodologias consistentes para o desenvolvimento de software;
- Gerenciamento de projeto desde a concepção até a manutenção.

! ATENÇÃO

É importante ressaltar que a ISO 9000-3 considera apenas quais processos a organização deve ter e manter, mas não orienta quanto aos passos que devem ser seguidos para chegar a desenvolvê-los e nem de como aperfeiçoá-los. A ISO 9001 baseia-se em 20 diretrizes (ou critérios) que englobam vários aspectos da garantia da qualidade.

A ISO 9000-3 engloba somente 12 desses elementos. O ponto central dos critérios de um sistema de gestão da qualidade baseada nas normas ISO 9000 é a apropriada documentação desse sistema. As diretrizes da ISO 9000-3 são:

- **Estrutura** – descreve aspectos organizacionais relacionados ao sistema de qualidade.

- **Atividades do ciclo de vida:** descreve atividades de desenvolvimento de software.

- **Atividades de suporte:** descreve atividades que apoiam as atividades do ciclo de vida.

4.4 NBR/ ISO 12207 - Processos de ciclo de vida do software

A norma ISO 12207 em como objetivo o estabelecimento de uma estrutura comum para os processos de ciclo de vida de software como forma de ajudar as organizações a compreenderem todos os componentes presentes na aquisição e fornecimento de software e, assim, conseguirem firmar contratos e executarem projetos de forma eficaz.

O gerenciamento por processos trás várias vantagens para as organizações, tais como:

- Alinha estrategicamente a organização.
- Foca a organização no cliente.
- Obriga a organização a prestar contas pelo desempenho dos seus processos.
- Alinha a força de trabalho com os processos.
- Evidencia a necessidade de alocação de recursos.
- Melhora a eficiência.

! ATENÇÃO

É necessário deixar claro que segundo a ISO/IEC 12207, cada processo deve ser de responsabilidade de uma parte. A parte que executa tem a responsabilidade por todo o processo, mesmo que tarefas individuais possam ser realizadas por pessoas diferentes.

Os processos da ISO/IEC 12207 são fortemente coesos, ou seja, todas as partes de um processo são fortemente relacionadas. Porém, os processos são fracamente acoplados quanto à quantidade de interfaces entre os processos.

O escopo da Norma ISO/IEC 12207 abrange todo o ciclo de vida de software, desde a concepção inicial até a descontinuidade do software, e por todos os envolvidos com produção, manutenção e operação do software. A norma pode

ser aplicada para toda empresa desenvolvedora de software, mas existem casos de aplicação em projetos específicos por imposição contratual ou nas fases iniciais de implantação.

A Norma ISO/IEC 12207 foi a referência base para a elaboração da Norma ISO/IEC 15504-5 publicada em 2006 e que define um modelo para a avaliação de processos de software baseado no framework da Norma ISO/IEC 15504 (ARRUDA, 2006).

Os processos da Norma ISO/IEC 12207 são agrupados de acordo com o seu objetivo principal no ciclo de vida de software. Estes agrupamentos resultam em três classes de processos: Processos Fundamentais, Processos de Apoio e Processos Organizacionais. Segundo Arruda (2006), a classe dos Processos Fundamentais são basicamente todas as atividades que a empresa executa nos serviços de desenvolvimento, manutenção ou operação de software. Esses processos comandam a execução de todos os outros processos. Os cinco processos fundamentais de ciclo de vida são:

a) Aquisição;
b) Fornecimento;
c) Desenvolvimento;
d) Operação;
e) Manutenção.

A classe dos Processos de Apoio é constituída por um conjunto de processos que estão ligados ao software através de ações de produção de Intercursos - V.8 - N.2 - Jul-Dez 2009 – ISSN 2179-9059 171documentação, testes e avaliação do produto desenvolvido.

A classe dos Processos Organizacionais é um conjunto de processos que fazem referências à gestão dos processos e dos recursos humanos envolvidos.

Vamos analisar com mais calma a arquitetura dessa norma.

Arquitetura da Norma ISO/NBR 12207

A Norma estabelece uma arquitetura de alto nível do ciclo de vida de software, abrangendo desde a concepção até a descontinuidade do software. Esta arquitetura é baseada em processos-chave e o inter-relacionamento entre eles.A arquitetura segue dois princípios básicos:

- **Modularidade:** Os processos têm alta coesão e baixo acoplamento, ou seja, todas as partes de um processo são fortemente relacionadas e o número de interfaces entre os processos é mantido ao mínimo.

- **Responsabilidade:** Cada processo na Norma é de responsabilidade de uma "parte envolvida". Uma "parte envolvida" pode ser uma organização ou parte dela. As partes envolvidas podem ser da mesma organização ou de organizações diferentes.

Na Norma ISO/IEC 12207, os processos de ciclo de vida são agrupados em quatro classes, que representam a sua natureza, a saber:

- **Processos Fundamentais:** Atendem o início, contratação entre o adquirente e o fornecedor, e a execução do desenvolvimento, operação ou manutenção de produtos de software durante o ciclo de vida de software.

- **Processos de Apoio:** Auxiliam e contribuem para o sucesso e qualidade do projeto de software.

- **Processos Organizacionais:** São empregados por uma organização para estabelecer e implementar uma estrutura constituída de processos de ciclo de vida e pessoal associados, melhorando continuamente a estrutura e os processos.

- **Processo de Adaptação:** O processo de adaptação define as atividades necessárias para executar a adaptação da Norma para sua aplicação na organização ou em projetos.

Norma é flexível para as abordagens de engenharia de software envolvidas, sendo :

- Utilizável com com qualquer modelo de ciclo de vida (cascata, incremental, evolutivo, etc);

- Utilizável com qualquer método ou técnica de engenharia de software (projeto orientado a objetos, técnicas estruturadas, prototipação, etc);

- Utilizável com com quaisquer linguagens de programação

A Norma implementa os princípios da gerência de qualidade. Ela os executa em três passos básicos:

- Integração da Qualidade no Ciclo de Vida
- Processo de Garantia da Qualidade
- Processo de Melhoria

A Norma provê os requisitos para um conjunto compreensivo e integrado de processos dentro de todo o ciclo de vida, onde cada processo é construído dentro do ciclo do PDCA (*plan-do-check-act*).

- Trata todas as atividades relacionadas à qualidade como uma parte integrante do ciclo de vida de software e também apropria essas atividades para cada processo no ciclo de vida.

Com relação ao Processo de Garantia da Qualidade podemos dizer que:

- O processo de garantia da qualidade é dedicado a assumir a concordância dos produtos e serviços com seus requisitos contratuais.
- Pessoas responsáveis por este processo são investidas da necessária liberdade e autoridade organizacional.

Com relação ao Processo de Melhoria podemos afirmar que :

- A Norma contém um processo de melhoria, em nível de organização e corporação, para gerenciamento da qualidade de seus próprios processos estabelecidos.
- Não especifica o como implementar ou executar as atividades e tarefas.
- Não determina um modelo de ciclo de vida ou método de desenvolvimento.
- Deve ser adaptada de acordo com o organização e projetos específicos.

ATIVIDADE

1. Comente as atividades da etapa de Ação do Ciclo PDCA.

2. Qual a proposta da fase de Estabelecimento do Modelo IDEAL?

3. Quais são as diretrizes da ISO 9000-3 ?

4. A Norma ISO 12207 está relacionada ao processo de ciclo de vida do software. Quais as vantagens para e empresa em gerenciar os processo de software?

5. Em quais princípios se baseia a arquitetura da Norma ISO/IEC 12207?

REFLEXÃO

Estamos quase chegando ao final do nosso estudo sobre qualidade de sofware e, pudemos perceber que, apesar da qualidade de software ser uma atividade relativamente nova dentro do contexto do desenvolvimento de software, ela segue, desde o seu surgimento modelos que já podemos considerar antigos e consolidados na literatura, como o PDCA e o IDEAL.

Os novos conceitos e normas relacionados à qualidade de software caminham juntos e relacionando-se com os modelos de gestão de qualidade mais antigos, nos mostrando que nem sempre existe um único modelo ou um única regra que vai suprir as necessidades de orientações para a obtenção da qualidade, seja de um produto ou de um processo de software.

LEITURA

Ciclo de Deming ou Ciclo PDCA
https://scsampaio.files.wordpress.com/2011/12/ciclo-de-deming-ou-ciclo-pdca.pdf

Processo de Melhoria Contínua de processos para a câmera dos deputados file:///C:/Users/Mayb/Downloads/processo_melhoria_aires.pdf

Melhoria de Processos de Tecnologia da Informação Multi-Modelo
http://www.inf.ufg.br/mestrado/sites/www.inf.ufg.br.mestrado/files/uploads/Dissertacoes/FabianaFreitas.pdf

Normas ISO: 12207 – 15504
http://olaria.ucpel.tche.br/venecian/lib/exe/fetch.php?media=qs_iso12207_15504.pdf

REFERÊNCIAS BIBLIOGRÁFICAS

PRESSMAN, Roger S. **Engenharia de Software**. Rio de Janeiro: MacGraw Hill, 2002.

ROCHA, ANA REGINA CAVALCANTI; MALDONADO, JOSÉ CARLOS; WEBER, KIVAL CHAVES. **Qualidade de Software** – Teoria e Prática. 1ª edição. São Paulo: Prentice Hall, 2001.

SOMERVILLE, IAN. **Engenharia de Software.** 6ª edição. São Paulo: Addison Wesley, 2003.

NO PRÓXIMO CAPÍTULO

No último capítulo desse material, finalizando nosso assunto abordaremos alguns modelos de qualidade de software, tais como o CMMI e o MPS. BR

5

Modelos de Qualidade de Processo de *Software*

5 Modelos de Qualidade de Processo de *Software*

Neste último capítulo da nossa disciplina de qualidade de software, iremos comentar estrutura do Modelo SPICE, da Norma ISO/IEC 15504 e também da transição do Modelo SPICE para a Norma ISO/IEC 155v04.

Fecharemos esse capítulo e também nosso livro apresentado dois modelos de grande importância para a área de qualidade de software, que são o CMMI e o MPS.BR

OBJETIVOS

- Conhecer o projeto SPICE (*Software Process Improvement and Capability Determination*) e a Norma ISO/IEC 15504
- Apresentar a definição e estrutura modelos CMMI e MPS.BR
- Proporcionar uma visão geral do processo de gerência de risco
- Compreender as atividades da gerência de risco.

REFLEXÃO

Você deve estar lembrado de quando comentamos no início desse trabalho que a qualidade de software pode ser analisada do ponto de vista de três dimensões diferentes?

Pois bem, essas dimensões são a visão do usuário, do desenvolvedor e da empresa que o comercializa.

Esse conceito nos mostra que a qualidade do software necessita ser bastante abrangente para poder agradar a estas três visões, que são sempre tão exigentes.

Para sabermos que o software agrada a essas três visões o mesmo precisa ser medido. Concordam que eu não posso dizer se o software tem qualidade ou não se eu não realizar uma avaliação no mesmo?

Nesse nosso último capítulo compreenderemos o funcionamento de alguns modelos que nos ajudam a realizar essa avaliação e consequentemente fazermos as melhorias onde forem necessárias! São esses, o SPICE, o CMMI e o MPS.BR

5.1 SPICE (*Software Process Improvement and Capability Determination*)

O projeto ***SPICE*** (*Software Process Improvement and Capability dEtermination*) surgiu quando o grupo ISO se reuniu buscando estabelecer um padrão para a avaliação do processo de software, pois constatou a deficiência da Norma ISO 9001 em relação ao setor de software e a existência de vários modelos.

Criado em 1993, o projeto ***SPICE*** tinha como objetivo gerar normas que avaliassem os processos de software e, em 1995, o projeto foi concluído gerando a primeira versão. Em 1998, foi publicado o ISO/IEC TR 15504: *Information Technology – Software Process Assessment* (Tecnologia da Informação – Avaliação de Processos de Softwate), como Relatório Técnico, que durante três anos foi submetido a uma revisão com o objetivo de se transformar em uma Norma Internacional.

! ATENÇÃO

É importante comentar aqui que o Modelo SPICE pode ser utilizado tanto em organizações que desenvolvem software visando à melhoria dos processos de produção, como também em organizações que adquirem esse software visando avaliar seus fornecedores em relação ao seu perfil de capacidade.

Este modelo constitui de um conjunto padronizado de processos fundamentais e define seis níveis de capacidade para cada um desses processos, que estão descritos em sua estrutura.

Para gerenciar a utilização das várias versões da Norma ISO 15504, foram realizados os projetos ***SPICE*** *Trials*, que são diversos testes práticos baseados no modelo com o objetivo de investigar a consistência do mesmo. Na Fase 1 e Fase 2 foram acompanhadas 105 avaliações de processos de software em organizações de diversas partes do mundo, e na Fase 3, foram acompanhadas outras avaliações tendo como referência o Relatório Técnico (ISO/IEC TR 15504). Além disso, esses testes consolidaram um dos objetivos do projeto ***SPICE***, que é disponibilizar um método de avaliação que possibilitasse a comparação dos resultados obtidos na avaliação com o uso de outros modelos.

O ***SPICE*** é um modelo contínuo, ou seja, define níveis de maturidade para determinados processos de acordo com diferentes características e o usuário determina sobre qual nível de maturidade o processo será avaliado.

A avaliação pode ser realizada no contexto de **melhoria contínua**, identificando as oportunidades de melhoria dos processos de uma organização, ou para **determinação da capacidade de processo**, determinando a conformidade dos processos de uma organização em relação a requisitos ou contratos.

5.1.1 Estrutura do Modelo Spice

O projeto ***SPICE*** está organizado em nove partes, onde apenas três partes são normativas e as demais são somente informativas, conforme descritas a seguir:

- **Parte 1 (informativa): Conceitos e Guia Introdutório**
  Esta parte descreve como interagem as partes do Modelo ***SPICE*** e fornece orientação para a sua seleção e utilização.

- **Parte 2 (normativa): Um Modelo de Referência para Processos e Capacitação de Processos**
  A parte 2 define um modelo de referência bidimensional usado como base para a avaliação e descreve quais atividades fundamentais são exigidas para uma boa engenharia de software, mas não descrevem como implementá-las. Nessa parte estão descritos quarenta processos e componentes de processos organizados em cinco categorias (Cliente-Fornecedor, Suporte, Engenharia, Gerência e Organização). Esses processos são um superconjunto dos processos que estão descritos na ISO/IEC 12207. Cada processo é descrito na parte 2 basicamente por um nome, o propósito e como reconhecer uma implementação com sucesso (Weber,et.al,2001). O modelo compreende duas dimensões (dimensão de processo e dimensão da capacidade do processo) que admite agregar níveis de maturidade a qualquer etapa do processo de desenvolvimento de um software.

- **Dimensão do processo**: caracteriza o propósito dos processos relacionados ao conjunto de atividades do ciclo de vida do software e é dividida em três agrupamentos.

- **Processos Fundamentais:** definem as atividades do ciclo de desenvolvimento do software, composto pelas categorias: Cliente-Fornecedor e Engenharia;

- **Processos Organizacionais:** definem as atividades que a organização devem executar, composto pelas categorias: Gerência e Organização;

- **Processos de Apoio:** definem as atividades para garantir a qualidade do projeto de software, composto pela categoria: Suporte.

- **Processos de Engenharia:** abrange os processos relacionados ao desenvolvimento e manutenção do software especificando, implementando ou mantendo o produto de software e a sua documentação para o cliente.

- **Processos de Suporte:** consiste nos processos de apoio ou suporte que podem ser empregados por qualquer outro processo durante o ciclo de vida do software.

- **Processos de Gerência:** consiste em processos que contêm práticas que podem ser utilizadas por quem administra qualquer processo ou projeto dentro do ciclo de vida de desenvolvimento de software.

- **Processos Organizacionais:** consiste em estabelecer objetivos tanto de negócio como de recursos humanos da organização, abrange processos relacionados às atividades gerais de uma organização, que ajudam a alcançar esses objetivos.

- ***Dimensão da Capacidade do Processo:*** define os níveis do processo em uma escala de medida de seis pontos que podem ser utilizados como base na avaliação da execução de um processo de uma organização, e como guia para a melhoria da execução. Cada nível de capacidade é descrito na parte 2 basicamente por um nome, definição e atributos (Weber et.al,2001). Os níveis são apresentados a seguir:

    - **Nível 0 – Incompleto:** processo não é implementado ou não atinge seus objetivos.

    - **Nível 1 – Executado:** processo é implementado atingindo seus objetivos, mas de forma pouco planejada ou rigorosa, sem padrão de qualidade e sem controle de prazo e custo.

- **Nível 2 – Gerenciado:** processo anterior é executado e gerenciado, isto é, planejado, controlado, acompanhado, verificado e corrigido, estando em conformidade a padrões e requisitos estabelecidos (qualidade, prazo e custo).
- **Nível 3 – Estabelecido:** processo anterior é executado e gerenciado com base nos princípios de uma boa engenharia de software, de forma eficaz e eficiente.
- **Nível 4 – Previsível:** processo anterior é executado dentro dos limites de controle estabelecido para atingir os objetivos definidos do processo e com medições detalhadas de desempenho, que são analisadas chegando a um entendimento quantitativo da capacidade do processo e a uma execução gerenciada quantitativamente.
- **Nível 5 - Em otimização:** processo anterior é alterado e adaptado continuamente, visando satisfazer os objetivos do negócio da organização, estabelecendo metas quantitativas de eficiência e eficácia para o desempenho do processo.

Os atributos do processo são medidos em uma escala de porcentagem que define o grau de satisfação da capacidade dos mesmos, detalhando os aspectos de um determinado processo, conforme a apresentado na tabela 5.1

AVALIAÇÃO	PORCENTAGEM	DESCRIÇÃO
N: Não Realizado	0% a 15%	Existe pouca ou nenhuma evidência que permite concluir que o processo avaliado satisfaça os objetivos dos atributos.
P: Parcialmente Realizado	16% a 50%	Existem evidências de uma abordagem sistemática para a realização do atributo no processo avaliado.

AVALIAÇÃO	PORCENTAGEM	DESCRIÇÃO
L: Largamente Realizado	51% a 85%	Existem evidências de uma abordagem sistemática para a realização significativa do atributo no processo avaliado.
F: Totalmente Realizado	86% a 100%	Existem evidências de uma abordagem sistemática e completa para a realização total do atributo no processo avaliado.

Tabela 5.1 – Escala de Porcentagem do Atributo (ISO SPICE, 2005)

Para estar em um nível de capacidade, um processo tem que ter todos os atributos de processo dos níveis anteriores totalmente atendidos (nota "F") e os atributos do nível em que ele se encontra tem que ser, no mínimo, largamente atendidos (notas "L" ou "F") (Côrtes & Chiossi, 2001).

- **Parte 3 (normativa): Executando uma Avaliação**
  Esta parte descreve os requisitos básicos para se realizar uma avaliação, visando atingir resultados repetíveis, confiáveis e consistentes. Os requisitos para a avaliação são (Sanches, 2003):

  - **Definição da Entrada da Avaliação:** definida antes da coleta de dados de uma avaliação e de ser aprovada pelo patrocinador da mesma, esta deve especificar: o patrocinador, o objetivo e as restrições da avaliação, o modelo a ser usado com a avaliação, os assessores, o pessoal de apoio, os critérios de competência do avaliador e dados que possam apoiar a melhoria do processo.

  - **Responsabilidades:** do patrocinador, que é verificar a competência do avaliador, e do assessor, que é confirmar o comprometimento do patrocinador com a avaliação, garantir que os participantes estejam cientes sobre o objetivo da avaliação e que esta seja conduzida em conformidade aos requisitos da norma, e documentar os requisitos encontrados na conclusão da avaliação.

- **Processo de Avaliação:** a avaliação deverá ser conduzida em conformidade com um processo documentado que cumpre o objetivo da avaliação.

- **Registro de Saída da Avaliação:** os dados referentes à avaliação, que auxiliarão na compreensão dos resultados obtidos, deverão ser incluídos no registro de avaliação, guardado pelo patrocinador, e este deve conter a data e a entrada da avaliação, a identificação dos objetivos e de dados adicionais coletados que foram reconhecidos na saída da avaliação para melhoria ou determinação da capacidade do processo, a abordagem de avaliação empregada, e o conjunto de representações de processo resultante da avaliação.

- **Parte 4 (informativa):** Guia de Orientação para a Condução de uma Avaliação
  A parte 4 descreve orientações para a realização de uma avaliação de processo de software, esclarecendo os requisitos da Parte 2 e da Parte 3 para avaliações em diferentes contextos. Este guia fornece orientação para seleção e uso de um Modelo Compatível e de um Processo de Avaliação Documentado, e seleção de instrumentos e ferramentas que auxiliarão no apoio a coleta, registro, processamento, análise e apresentação de dados.

- **Parte 5 (informativa): Um Modelo de Avaliação e Guia de Indicadores**
  Esta parte descreve um modelo para auxiliar a realização de uma avaliação de processo de software compatível ao Modelo de Referência da Parte 2, e um guia de boas práticas de Engenharia de Software que visam complementar o Modelo.

- **Parte 6 (informativa): Guia para Competência de Avaliadores**
  A parte 6 descreve a competência, formação, treinamento e experiência dos avaliadores que são fundamentais para a execução da avaliação de processos. Esta parte fornece mecanismos que podem ser utilizados com o objetivo de comprovar a competência e validar a formação, treinamento e experiência dos avaliadores.

- **Parte 7 (informativa):** Guia de Orientação para Melhoria de Processo
  Essa parte descreve como definir as entradas para a execução de uma avaliação e como utilizar os resultados obtidos visando a melhoria do processo.

Visando a melhoria do processo, uma organização define um método para a escolha dos processos que estão descritos na parte 2 do Modelo ***SPICE***, que é parte de uma abordagem para a melhoria da organização.

- **Parte 8 (informativa): Um Modelo de Avaliação e Guia de Indicadores**
  A parte 8 descreve como definir as entradas para a execução de uma avaliação e como utilizar os resultados obtidos visando a determinação da capacidade do processo. Esta parte auxilia a organização em identificar os pontos fortes e fracos associados ao desenvolvimento de processos, e os riscos de firmar um contrato com um fornecedor. A determinação da capacidade de um processo é fundamental por obter informações que auxiliam na tomada de decisões, e pode ser realizado pelo adquirente (na seleção do fornecedor) e pelos fornecedores (na decisão de um contrato avaliando seus próprios processos e analisando seus riscos envolvidos).

- **Parte 9 (normativa): Vocabulário**
  A parte 9 define todos os termos, em ordem alfabética, usados por todas as partes que compõem o modelo.

5.2 Transição para a Norma ISO/IEC 15504

Publicada em 1998, como um Relatório Técnico do Tipo 2, a ISO/IEC TR 15504: *Information Technology – Software Process Assessment* (Tecnologia da Informação – Avaliação de Processos de Softwate) foi submetido a uma revisão durante três anos, com o objetivo de se transformar em uma Norma Internacional. Atualmente, tem-se a nova Norma ISO/IEC 15504 como padrão para a avaliação de processos, que visa a melhoria contínua e determinação da capacidade dos processos de uma organização.

As principais alterações que ocorreram do Projeto ***SPICE*** para nova estrutura da Norma ISO/IEC 15504 foram:

Redução do número de partes, de nove para cinco partes, sendo apenas as partes 1 e 2 normativas e as demais informativas:

- *Parte 1*: Conceitos e Vocabulário (baseada nas partes 1 e 9 do ***SPICE***);
- *Parte 2:* Realização de uma Avaliação (baseada nas partes 2 e 3 do ***SPICE***);

- *Parte 3:* Guia para a Realização de uma Avaliação (baseada nas partes 4 e 6 do ***SPICE***);
- *Parte 4:* Guia para a Utilização dos Resultados de uma Avaliação (baseada nas partes 7 e 8 do ***SPICE***);
- *Parte 5:* Um Exemplo de Modelo de Avaliação de Processos (baseada na parte 5 do ***SPICE***).

Compatibilização com a Norma ISO/IEC 12207, visto que há muita superposição entre a dimensão de processos do ISO/IEC TR 15504 e da ISO/IEC 12207, o que provoca a duplicação de esforços, inconsistências e dificuldade na utilização da descrição de processos da ISO/IEC 12207 que contêm conceito de capacidade embutido. Para solucionar esses problemas, a dimensão de processos foi removida da ISO/IEC 15504, se tornando um anexo da Norma ISO/IEC 12207 publicada como AMD1 e AMD2, e permitindo a ISO/IEC 15504 utilizar qualquer Modelo Compatível de descrição de processo como Modelo de Referência. Sendo assim, a ISO/IEC 15504 será uma Norma para avaliação da capacidade de processos.

O maior benefício dessas alterações está na abordagem comum de avaliação, não sendo mais restringida aos processos do ciclo de vida do software, e sim, podendo ser executada em qualquer tipo de processo em qualquer domínio.

CONCEITO

Atualmente, a Dimensão do Processo é definida por um Modelo de Referência externo à ISO/IEC 15504 e serve como base para o Modelo de Avaliação de Processo.

5.3 Modelo CMMI (*Capability Maturity Model*)

O CMMI (*Capability Maturity Model Integration*) foi criado pelo SEI (*Software Engineering Institute*), o qual é um órgão integrante da universidade norte-americana *Carnegie Mellon*. Trata-se de um modelo que está atualmente na versão 1.3 (Janeiro/2013), com um enfoque voltado para a capacidade de maturidade de processos de software.

O CMMI está dividido em 5 níveis de maturidade que atestam, por sua vez, o grau de evolução em que uma organização se encontra num determinado momento. Além disso, tem por objetivo principal funcionar como um guia para

a melhoria dos processos da organização, considerando para isto atividades como o gerenciamento do desenvolvimento de software, prazos e custos previamente estabelecidos. O objetivo maior, considerando o CMMI e seus diferentes conceitos, está justamente na produção de software com maior qualidade e menos propenso a erros. Para se conseguir o que este modelo propõe, a organização interessada na implantação do CMMI deverá evoluir progressivamente, considerando para isto uma sucessão de diferentes de níveis. Cada nível indica, por sua vez, o grau de maturidade dos processos num determinado instante:

- **Nível 1 - Inicial**: os processos normalmente estão envoltos num caos decorrente da não obediência ou ainda, inexistência de padrões;
- **Nível 2 - Gerenciado**: os projetos têm seus requisitos gerenciados neste ponto. Além disso, há o planejamento, a medição e o controle dos diferentes processos;
- **Nível 3 - Definido**: os processos já estão claramente definidos e são compreendidos dentro da organização. Os procedimentos se encontram padronizados, além de ser preciso prever sua aplicação em diferentes projetos;
- **Nível 4 - Gerenciado Quantitativamente**: ocorre o aumento da previsibilidade do desempenho de diferentes processos, uma vez que os mesmos já são controlados quantitativamente;
- **Nível 5 - Otimizado**: existe uma melhoria contínua dos processos.

CONEXÃO

Maiores informações sobre o modelo CMMI podem ser obtidas através do seguinte link: http://www.sei.cmu.edu/cmmi/

5.4 Modelo Mps.Br

O MPS.BR ou Melhoria de Processos do Software Brasileiro, é simultaneamente um movimento para a melhoria e um modelo de qualidade de processo voltada para a realidade do mercado de pequenas e médias empresas de desenvolvimento de software no Brasil.

Ele é baseado no CMMI, nas normas ISO/IEC 12207 e ISO/IEC 15504 e na realidade do mercado brasileiro. No Brasil, uma das principais vantagens do modelo é seu custo reduzido de certificação em relação às normas estrangeiras, sendo ideal para micro, pequenas e médias empresas.

Um dos objetivos do projeto é replicar o modelo na América Latina, incluindo o Chile,Argentina, Costa Rica, Peru e Uruguai.

O projeto tem apoio do Ministério de Ciência e Tecnologia, do FINEP e do Banco Interamericano de Desenvolvimento. No Brasil o projeto é desenvolvido pelo Softex, pelo governo e por universidades.

Os níveis de maturidade estabelecem patamares de evolução de processos,caracterizando estágios de melhoria da implementação de processos na organização.

O MR-MPS define sete níveis de maturidade: A (Em Otimização),B (Gerenciado Quantitativamente), C (Definido), D (Largamente Definido), E (Parcialmente Definido), F (Gerenciado) e G (Parcialmente Gerenciado).

O progresso e o alcance de um determinado nível de maturidade MPS se obtém quando são atendidos os propósitos e todos os resultados esperados dos respectivos processos e dos atributos de processo estabelecidos para aquele nível. 50 A divisão em estágios, embora baseada nos níveis de maturidade do CMMI-SE/SW tem uma graduação diferente, com o objetivo de possibilitar uma implementação e avaliação mais adequada às micros, pequenas e médias empresas.

A capacidade do processo é representada por um conjunto de atributos de processo descrito em termos de resultados esperados. A capacidade do processo expressa o grau de refinamento e institucionalização com que o processo é executado na organização.

O atendimento aos atributos do processo (AP), através do atendimento aos resultados esperados dos atributos do processo (RAP) é requerido paratodos os processos no nível correspondente ao nível de maturidade, embora eles não sejam detalhados dentro de cada processo.

LEITURA

Para saber mais sobre o modelo MPS.BR acesse o artigo: **Comparando CMMi x MPS. BR: As Vantagens e Desvantagens dos Modelos de Qualidade no Brasil**. Disponível em: http://www.camilaoliveira.net/Arquivos/Comparando%20CMMi%20x%20MPS.pdf

Os níveis são acumulativos, ou seja, se a organização está no nível F, esta possui o nível de capacidade do nível F que inclui os atributos de processo dos níveis G e F para todos os processos relacionados no nível de maturidade F (que também inclui os processos de nível G). Isto significa que, ao passar do nível G para o nível F,os processos do nível de maturidade G passam a ser executados no nível decapacidade correspondente ao nível F.

5.5 Gerência de Riscos na Qualidade de Software

Existem várias modelos que auxiliam o processo de gerenciamento de riscos. Alguns desses modelos são PMBOK, RUP, CMMI, ISO, entre outros.

Riscos são eventos que podem ocorrer no futuro e se concretizados, afetam o projeto de software. Os fatores relacionados ao risco são: o evento ou fato que caracteriza o risco, a probabilidade de que ele irá ocorrer (possibilidade) e a perda resultante de sua ocorrência (conseqüência).

O risco caracteriza-se pela incerteza (o evento que caracteriza o risco pode ou não ocorrer) e perda (se o evento ocorrer, conseqüências inesperadas ou perdas irão ocorrer). Ele faz parte de qualquer atividade, não podendo ser eliminado. Além disso, não é possível conhecer todos os riscos de um projeto de software.

As atividades associadas ao risco podem ser divididas em: gerenciais e técnicas. As atividades gerenciais são responsáveis pelo planejamento, organização, seleção de pessoas, controle, direção, garantia da qualidade e gerência da configuração. Esta atividade está influencia diretamente o projeto e ao processo. Já as atividades técnicas englobam a análise de requisitos, projeto, código e teste. Esta atividade influencia tanto o produto quanto o processo.

Riscos associados ao projeto são problemas relacionados a aspectos operacionais, organizacionais e contratuais do projeto (restrições de recursos, interfaces externas, relacionamento com fornecedores, restrições de contrato e falta de suporte organizacional). Os riscos mais significativos em relação ao processo são planejamento (gerencial) e processo de desenvolvimento (técnico)

O produto está relacionado à atividade técnica através da estabilidade dos requisitos (requisitos são percebidos como flexíveis, o que dificulta a gerência de riscos), desempenho, complexidade do código, especificação dos testes. Os riscos mais significativos são ligados aos requisitos.

Para um bom gerenciamento dos riscos de um projeto de software faz-se necessário a execução das seguintes atividades:

- **Identificar***:* produzir uma lista dos riscos que podem vir a comprometer o resultado esperado.
  Os riscos podem ser genéricos (ameaçam qualquer projeto) ou específicos do projeto. Para identificar os riscos específicos é necessário entender a tecnologia, a equipe e o ambiente do projeto, examinar o plano do projeto e o escopo do sistema e responder a seguinte pergunta: "que características especiais este sistema possui que podem ameaçar o plano do projeto?"

- **Analisar**: Avaliar a probabilidade da perda e a magnitude da perda, associada com cada item de risco e com a possível composição desses riscos.

- **Planejar**: definir ações para endereçar cada item de risco, priorizar as ações e criar um plano integrado de gerência de riscos. O planejamento pode ter vários objetivos:
  - Atenuar o risco.
  - Evitar o risco modificando o projeto do produto ou o processo de desenvolvimento.
  - Transferir o risco.
  - Aceitar o risco e as conseqüências, caso o evento ocorra.
  - Fazer um estudo futuro do risco para obter mais informações e compreender melhor as características do risco.

- **Monitorar***:* acompanhar o status do risco e das ações executadas

- **Resolver:** executar as ações planejadas e reportar os resultados da resolução do risco.

- **Comunicar***:* prover informações para e entre entidades e níveis organizacionais envolvidos no projeto. Inclui níveis dentro do projeto, da organização, da organização do cliente e a comunicação entre desenvolvedor, cliente e usuário.

5.5.1 Um Processo para Gerenciamento de Riscos

O risco nos projetos de software pode ser gerenciado através das seguintes atividades:

- identificação dos riscos,
- análise dos riscos,
- planejamento dos riscos,
- acompanhamento dos riscos e
- resolução dos riscos.

O processo começa com a identificação dos riscos. Tudo o que se referir a incerteza, experiências anteriores, preocupações e questões a resolver pode ser útil na identificação dos riscos. Várias fontes podem ser utilizadas nesta fase:

- **pessoas** incluem clientes, integrantes da equipe, organizações envolvidas, disponibilidade, capacidade, experiência, etc.;
- os **produtos** e processos abrangem requisitos, prazos, estimativas, receitas, despesas, orçamento, restrições de natureza legal, estilo de gerenciamento, tamanho e escopo do projeto, etc.;
- a **tecnologia** inclui mudanças, inovação, adoção e uso, integração e interfaces, experiência específica, segurança, arquitetura, escalabilidade, etc.

Uma Avaliação de Riscos (*Risk Assessment*) deve ser conduzida a fim de identificar e registrar sistematicamente os riscos do projeto. Entrevistas, reuniões, pesquisas e listas de verificação (*checklists*) são instrumentos úteis na condução de uma Avaliação de Riscos.

O *Software Engineering Institute* (SEI) oferece uma categorização de riscos muito útil nesta área , onde a análise dos riscos é iniciada agrupando-se os riscos de mesma natureza, ou semelhantes.

É importante determinar os fatores atuantes sobre os riscos, isto é, as variáveis que fazem a probabilidade de ocorrência ou o impacto (valor da perda) dos riscos flutuarem. Também devem ser determinadas as fontes de risco, ou seja, as respectivas causas, normalmente descobertas respondendo-se à pergunta "Por quê?" com relação a cada risco identificado.

Em seguida, deve-se calcular a exposição referente a cada risco, definida como o produto da probabilidade de ocorrência do risco pelo respectivo impacto. A exposição é utilizada na priorização dos riscos.

O planejamento dos riscos inclui a definição de cenários para os riscos mais importantes, a definição de alternativas de solução para esses cenários, a escolha das alternativas mais adequadas, o desenvolvimento de um Plano de Ação de Riscos, assim como o estabelecimento de limiares ou disparadores para a ação.

O acompanhamento dos riscos envolve a monitoração dos cenários de riscos, a verificação de que os limiares foram ou não atingidos, bem como a análise das medidas e indicadores referentes aos riscos.

A resolução dos riscos inclui a resposta aos eventos disparadores, a execução do Plano de Ação de Riscos, o acompanhamento da execução do plano e as eventuais correções de desvios.

A seguir apresentamos algumas atividades a serem resolvidas com o intuito de fixar os conhecimentos aprendidos nesse capítulo.

ATIVIDADE

1. O Modelo SPICE possui 9 partes. Na segunda parte encontra-se o um Um Modelo de Referência para Processos e Capacitação de Processos e, nesse modelo são definidos quatro tipos de processos, comente cada uma deles.

2. Explique as principais alterações que aconteceram na transição do Modelo SPICE para a Norma ISO/IEC 15504.

3. As principais alterações que ocorreram do Projeto SPICE para nova estrutura da Norma ISO/IEC 15504 foram:

4. O CMMI está dividido em 5 níveis de maturidade que atestam o grau de evolução em que uma organização. Comente o que lembra de cada um desses níveis.

5. Você se lembra se o MPS.BR possui alguma vantagem para o mercado brasileiro com relação aos outros modelos vistos?

REFLEXÃO

Nosso material termina por aqui, apresentando a Norma ISO/IEC 15505, os Modelos *SPICE*, CMMI e MPS.BR, mas você ainda possui um amplo universo de informações importantes

sobre a qualidade de software e seus modelos de melhoria. Espero que tenha apreciado o assunto e que o mesmo lhe seja útil na sua vida profissional!

Continue se atualizando, a qualidade de software ainda é uma área recente e sempre podemos contar com mais informações e estudos de caso para nossa aprimorar nossa formação.

LEITURA

Padrões de Qualidade de Software – MPS Br x CMMI e a realidade Brasileira.
http://www.2xt.com.br/gerencia-de-requisitos-em-processos-de-desenvolvimento-de-software/

Principais diferenças entre o MPS.BR e o CMMI
http://longhigh.wordpress.com/2008/05/15/principais-diferencas-entre-o-mps-br-e-o-cmmi/

Projetos da CE-21-007-10 e Novidades da ISO/IEC 15504 (SPICE)
http://www.softwarepublico.gov.br/file/17039869/apresISO-IEC-15504-Clenio-40min-v1p2-ABNT2009.pdf

Arquitetura da ISO/IEC 15504
http://www.micpm.com/hdk/manuais/MiCPManuaisManuais_103.html

REFERÊNCIAS BIBLIOGRÁFICAS

(CHARETTE,1989) R. N. **Software Engineering** Risk Analysis and Management, McGraw-Hill/Intertext, 1989.

(CMMI, 2014)CMMI- **Capability Maturity Model**, disponível em: http://cmmiinstitute.com/assets/CMMI-DEV_1-2_Portuguese.pdfISO

(CMMI:2000) C**MMI Model Componentes Derived from CMMI** – SE/SW, Version 1.0 Technical report CMU/SEI-00-TR24. Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2000.

(ISO, 2014) – **International Organization for Standardization**. Disponível em: http://www.iso.org

(PÁDUA, 2005) Wilson de Pádua Paula. **Engenharia de software**: fundamentos, métodos e padrões. 3ª ed. Rio de Janeiro: LTC, 2005.

(PRESSMAN, 1997) Roger S. Pressman, **Engenharia de Software**, Makron Books, McGraw-Hill, 1997

(Softex, 2012D) – Associação Para Promoção Da Excelência Do Software Brasileiro – Softex. MPS.BR – Guia de Implementação – Parte 11: Implementação e Avaliação do MR-MPS-SW:2012 em Conjunto com o CMMI-DEV v1.3, Disponível em: www.softex.br.

(*Wikipédia*, 2014) – A enciclopédia livre. Disponível em: http://www.wikipedia.org

EXERCÍCIO RESOLVIDO

Capítulo 1

1. Defina qualidade de software.

Qualidade de software possui várias definições, vejamos algumas:

- Qualidade de software é a conformidade a requisitos funcionais e de desempenho que foram explicitamente declarados, a padrões de desenvolvimento claramente documentados, e a características implícitas que são esperadas de todo software desenvolvido por profissionais.
- Um produto de software apresenta qualidade dependendo do grau de satisfação das necessidades dos clientes sob todos os aspectos do produto.
- Qualidade é a totalidade de características e critérios de um produto ou serviço que exercem suas habilidades para satisfazer as necessidades declaradas ou envolvidas.
- Qualidade é a totalidade das características de uma entidade, que lhe confere a capa-

cidade de satisfazer necessidades explícitas e implícitas.

2. Defina processo de software e comente as características de um bom processo de software.

Um processo de software pode ser definido como um conjunto de atividades, métodos, práticas e transformações que as pessoas usam para desenvolver e manter o software e os produtos associados (por exemplo: planos de projeto, documentos, código, casos de teste e manuais de usuário).

Um processo de software envolve um grande conjunto de elementos, tais como objetivos organizacionais, políticas, pessoas, comprometimentos, ferramentas, métodos, atividades de apoio e as tarefas da engenharia de software. Para que o processo de software seja eficiente ele precisa ser constantemente avaliado, medido e controlado. Quando o processo é eficiente ele possui as seguintes características:

- O processo continua a despeito de problemas inesperados (Robustez).
- Rapidez na produção do sistema (Velocidade).
- O processo é aceito por todos os envolvidos nele (Aceitabilidade)
- Os erros do processo são descobertos antes que resultem em erros no produto (Confiabilidade)
- O processo evolui para atender alterações de necessidades organizacionais (Manutenibilidade)
- O processo é compreendido (usualmente através de documentação e de treinamento), utilizado, vivo e ativo.
- O processo é bem controlado – a fidelidade ao processo é objeto de auditoria e de controle.
- Medidas do produto e do processo são utilizadas,
- Os papéis e responsabilidades no processo estão claros ao longo de todo o projeto e por toda a organização

3. Comente sobre a categorização das métricas.

Com relação à categorização das métricas de software podemos citar:

- Métricas diretas (fundamentais ou básicas): medida realizada em termos de atributos observados (usualmente determinada pela contagem). Ex.: custo, esforço, n° linhas de código, capacidade de memória, n° páginas, n° diagramas, etc.
- Métricas indiretas (derivadas): medidas obtidas a partir de outras métricas. Ex.: complexidade, eficiência, confiabilidade, facilidade de manutenção.
- Métricas orientadas a tamanho: são medidas diretas do tamanho dos artefatos de software associados ao processo por meio do qual o software é desenvolvido. Ex.: esforço, custo, no. KLOC, n° páginas de documentação, no. Erros.
- Métricas orientadas por função: consiste em um método para medição de software do ponto de vista do usuário, determinando de forma consistente o tamanho e a complexidade de um software.
- Métricas de produtividade: concentram-se na saída do processo de engenharia de software. Ex.: n° de casos de uso/iteração.
- Métricas de qualidade: Oferecem uma indicação de quanto o software se adequa às exigências implícitas e explícitas do cliente. Ex.: erros/fase.
- Métricas técnicas: concentram-se nas características do software e não no processo por meio do qual o software foi desenvolvido. Ex.: complexidade lógica e grau de manutenibilidade.

4. Quando falamos de revisões de software, o que é importante que o engenheiro considere no planejamento?

Devem ser consideradas as seguintes questões:

- quem participa?
- qual informação é requerida antes da revisão?
- quais pré-condições que devem ser satisfeitas antes que a revisão possa ser conduzida?
- Como Organizar?
- Gerar checklists ou outra indicação do que deve ser coberto na revisão;

- Determinar as condições de término ou critérios que devem ser satisfeitos para que a revisão termine;
- Gerar registros e documentos que devem ser produzidos.

Capítulo 2

1. Comente algumas vantagens da realização da garantia de qualidade de software.

Alguns aspectos positivos da garantia de qualidade de software que podemos mencionar são:

- o software terá menos defeitos latentes resultando em redução do esforço e do tempo gasto durante as atividades de teste e manutenção
- a maior confiabilidade resultará em maior satisfação do cliente
- os custos de manutenção podem ser reduzidos
- o custo do ciclo de vida global do software é reduzido

2. Explique as subcaracterísticas da característica FUNCIONALIDADE da norma ISO 9126.

- Funcionalidade: capacidade do produto de software de prover funções que atendam às necessidades explícitas e implícitas, quando o software estiver sendo utilizado sob condições específicas.
- Adequação: capacidade do produto de software de prover um conjunto apropriado de funções para tarefas e objetivos do usuário especificados.
- Acurácia: capacidade do produto de software de prover, com o grau de precisão necessário, resultados ou efeitos corretos ou conforme acordados.
- Interoperabilidade: capacidade do produto de software de interagir com um ou mais sistemas especificados.
- Segurança de Acesso: capacidade do produto de software de proteger informações e dados, de forma que pessoas ou sistemas não autorizados não possam lê-los nem modificá-los e que não seja negado o acesso às pessoas ou sistemas autorizados.

- Conformidade: capacidade do produto de software de estar de acordo com normas, convenções ou regulamentações previstas em leis e prescrições similares relacionadas à funcionalidade.

3. O que é um pacote de software? Dê alguns exemplos.

Pacote de Software: trata-se de um produto de software que envolve um conjunto completo e documentado de programas fornecidos a diversos usuários para uma aplicação ou função genérica.

Um Pacote de Software envolve todos os componentes do produto disponíveis aos usuários, tais como documentação, manual de instruções e guia para instalação.

Exemplos: Processadores de Texto, Planilhas Eletrônicas, Gerenciadores de Banco de Dados, Software Gráficos, Programas para Funções Técnicas ou Científicas e Programas Utilitários

4. Quais são os pré-requisitos de testes da norma ISO 12119?

Os pré-requisitos de teste são:

- Presença de itens de produto,
- Presença do sistema necessário e,
- Treinamento (se mencionado na descrição do produto)

5. O que deve ser testado de acordo com a norma ISO 12119?

As atividades de teste consistem em testar se estão de acordo com os requisitos de qualidade:

- Descrição do produto
- Documentação do usuário
- Programas e dados

Capítulo 3

1. Comente uma das vantagens de se usar a Norma ISO 9241?

Esclarecer os benefícios de medir usabilidade em termos de desempenho e satisfação do usuário, lembrando sempre que a usabilidade dos computadores depende do contexto de uso e que o nível de usabilidade alcançado dependerá das circunstâncias específicas nas quais o produto é usado.

2. **Como a ISO 9241-11 redefine usabilidade?**

   Como a capacidade de um produto ser usada por usuários específicos para atingir objetivos específicos com eficácia, eficiência e satisfação em um contexto específico de uso.

3. **De acordo com a ISO 14598, o que significa avaliar a qualidade de um produto de software?**

   Significa verificar, através de técnicas e atividades operacionais, o quanto os requisitos são atendidos. - Tais requisitos, de uma maneira geral, expressam as necessidades explicitadas em termos quantitativos ou qualitativos e apresentam como objetivo a definição das características de um software, a fim de permitir o exame de seu entendimento.

4. **Você se lembra de quantas e quais são as partes da ISO/IEC 9241? São várias, mas faça um esforço.**

   A ISO/IEC 9241 é dividida em 14 partes, a saber:

   - Parte 1: Introdução Geral
   - Parte 2: Orientações sobre requisitos da tarefa
   - Parte 3: Requisitos para apresentação visual
   - Parte 4: Requisitos para teclado
   - Parte 5: Requisitos posturais e de layout para posto de trabalho
   - Parte 6: Requisitos para ambiente
   - Parte 7: Requisitos para monitores quanto à reflexão
   - Parte 8: Requisitos para apresentação de cores
   - Parte 9: Requisitos para outros dispositivos de entrada que não o teclado
   - Parte 10: Princípios de diálogo
   - Parte 11: Orientações sobre usabilidade
   - Parte 12: Apresentação da informação
   - Parte 13: Orientações ao usuário
   - Parte 14: Diálogos por menu

Capítulo 4

1. **Comente as atividades da etapa de Ação do Ciclo PDCA.**

   Consiste em atuar no processo em função dos resultados obtidos. Se o sucesso é obtido em uma escala menor, pode significar que a mudança pode fazer parte das atividades da organização, envolvendo outras pessoas, outros departamentos, fornecedores e clientes afetados pelas mudanças. O envolvimento de outras pessoas é importante para ajudar na implementação das mudanças numa escala maior, ou mesmo para a

conscientização dos novos benefícios que podem adquirir com as mudanças. Existem duas formas de atuação possíveis:

- adotar o plano como padrão, caso a meta tenha sido alcançada
- agir sobre as causas que não permitiram que as metas sejam efetivas.

2. Qual a proposta da fase de Estabelecimento do Modelo IDEAL?

A proposta dessa fase é definir prioridades para as alterações, desenvolver uma estratégia para realização do trabalho, identificar recursos disponíveis e desenvolver um plano de implementação detalhado, o qual contém horários, tarefas, pontos de decisão, recursos, responsabilidades, estratégias de risco e qualquer outro elemento requerido pela organização.

3. Quais são as diretrizes da ISO 9000-3?

As diretrizes são divididas em 3, a saber:

- Estrutura – descreve aspectos organizacionais relacionados ao sistema de qualidade.
- Atividades do ciclo de vida: descreve atividades de desenvolvimento de software.
- Atividades de suporte: descreve atividades que apoiam as atividades do ciclo de vida.

4. A Norma ISO 12207 está relacionada ao processo de ciclo de vida do software. Quais as vantagens para e empresa em gerenciar os processos de software?

Podemos citar as seguintes vantagens:

- Alinha estrategicamente a organização.
- Foca a organização no cliente.
- Obriga a organização a prestar contas pelo desempenho dos seus processos.
- Alinha a força de trabalho com os processos.
- Evidencia a necessidade de alocação de recursos.
- Melhora a eficiência.

5. Em quais princípios se baseia a arquitetura da Norma ISO/IEC 12207?

A arquitetura segue dois princípios básicos:

- Modularidade - Os processos têm alta coesão e baixo acoplamento, ou seja, todas as partes de um processo são fortemente relacionadas e o número de interfaces entre os processos é mantido ao mínimo

- Responsabilidade - Cada processo na Norma é de responsabilidade de uma "parte envolvida". Uma "parte envolvida" pode ser uma organização ou parte dela. As partes envolvidas podem ser da mesma organização ou de organizações diferentes.

Capítulo 5

1. **O Modelo SPICE possui 9 partes. Na segunda parte encontra-se o um Um Modelo de Referência para Processos e Capacitação de Processos e, nesse modelo são definidos quatro tipos de processos, comente cada um deles.**
   - Processos de Engenharia: abrange os processos relacionados ao desenvolvimento e manutenção do software especificando, implementando ou mantendo o produto de software e a sua documentação para o cliente.
   - Processos de Suporte: consiste nos processos de apoio ou suporte que podem ser empregados por qualquer outro processo durante o ciclo de vida do software.
   - Processos de Gerência: consiste em processos que contêm práticas que podem ser utilizadas por quem administra qualquer processo ou projeto dentro do ciclo de vida de desenvolvimento de software.
   - Processos Organizacionais: consiste em estabelecer objetivos tanto de negócio como de recursos humanos da organização, abrange processos relacionados às atividades gerais de uma organização, que ajudam a alcançar esses objetivos.

2. **Explique as principais alterações que aconteceram na transição do Modelo SPICE para a Norma ISO/IEC 15504.**
   As principais alterações que ocorreram do Projeto SPICE para nova estrutura da Norma ISO/IEC 15504 foram:

   - Redução do número de partes, de nove para cinco partes, sendo apenas as partes 1 e 2 normativas e as demais informativas:
     - Parte 1: Conceitos e Vocabulário (baseada nas partes 1 e 9 do SPICE);
     - Parte 2: Realização de uma Avaliação (baseada nas partes 2 e 3 do SPICE);
     - Parte 3: Guia para a Realização de uma Avaliação (baseada nas partes 4 e 6 do SPICE);
     - Parte 4: Guia para a Utilização dos Resultados de uma Avaliação (baseada nas partes 7 e 8 do SPICE);
     - Parte 5: Um Exemplo de Modelo de Avaliação de Processos (baseada na parte 5 do SPICE).

- Compatibilização com a Norma ISO/IEC 12207, visto que há muita superposição entre a dimensão de processos do ISO/IEC TR 15504 e da ISO/IEC 12207, o que provoca a duplicação de esforços, inconsistências e dificuldade na utilização da descrição de processos da ISO/IEC 12207 que contêm conceito de capacidade embutido. Para solucionar esses problemas, a dimensão de processos foi removida da ISO/IEC 15504, se tornando um anexo da Norma ISO/IEC 12207 publicada como AMD1 e AMD2, e permitindo a ISO/IEC 15504 utilizar qualquer Modelo Compatível de descrição de processo como Modelo de Referência. Sendo assim, a ISO/IEC 15504 será uma Norma para avaliação da capacidade de processos.

O maior benefício dessas alterações está na abordagem comum de avaliação, não sendo mais restringida aos processos do ciclo de vida do software, e sim, podendo ser executada em qualquer tipo de processo em qualquer domínio.

3. O CMMI está dividido em 5 níveis de maturidade que atestam o grau de evolução em que uma organização. Comente o que lembra de cada um desses níveis.

- Nível 1 - Inicial: os processos normalmente estão envoltos num caos decorrente da não obediência ou ainda, inexistência de padrões;
- Nível 2 - Gerenciado: os projetos têm seus requisitos gerenciados neste ponto. Além disso, há o planejamento, a medição e o controle dos diferentes processos;
- Nível 3 - Definido: os processos já estão claramente definidos e são compreendidos dentro da organização. Os procedimentos se encontram padronizados, além de ser preciso prever sua aplicação em diferentes projetos;
- Nível 4 - Gerenciado Quantitativamente: ocorre o aumento da previsibilidade do desempenho de diferentes processos, uma vez que os mesmos já são controlados quantitativamente;
- Nível 5 - Otimizado: existe uma melhoria contínua dos processos.

4. **Você se lembra se o MPS.BR possui alguma vantagem para o mercado brasileiro com relação aos outros modelos vistos?**
   O MPS.BR é baseado no CMMI, nas normas ISO/IEC 12207 e ISO/IEC 15504 e também na realidade do mercado brasileiro.

   No Brasil, uma das principais vantagens do modelo é seu custo reduzido de certificação em relação às normas estrangeiras, sendo ideal para micro, pequenas e médias empresas