

Capítulo

11

Uma introdução ao SWEBOK

André Luís de Lucena Torres¹

Nas últimas décadas, a computação tem se desdobrado em uma extensa lista de subáreas de estudo. A quantidade de informação aumentou de tal modo que a especialização profissional tornou-se comum, de modo a alcançar o nível de excelência desejado. Este capítulo tem o objetivo de apresentar o SWEBOK (*Software Engineering Body of Knowledge*), um guia para o corpo de conhecimento em Engenharia de Software, patrocinado pelo IEEE (*Institute of Electrical and Electronics Engineering*).

11.1. Introdução

O termo Engenharia de Software foi usado pela primeira vez como tema na conferência da OTAN no ano de 1968, evento esse que foi motivado pela crise no desenvolvimento de software. Nesta época, a demanda por métodos padronizados para a qualidade de software era muito grande, visto que a maioria das aplicações era produzida sem nenhuma padronização. John Ronald Graham comentou: "construímos sistemas como os irmãos Wright construíam aviões - constrói-se de uma só vez, empurra-se para o despenhadeiro, deixa bater e começa tudo outra vez" [Naur & Randell, 1968].

O aumento gradual e crescente da capacidade de processamento dos computadores revelou a necessidade de se criar processos que orientassem e organizassem a atividade de desenvolvimento de software, deixando de ser uma atividade que até então supria apenas as necessidades do hardware.

Desde os primeiros computadores comerciais, os softwares implantados ou lançados no mercado se caracterizam, na sua maioria, pela presença de erros encontrados nas fases de verificação e validação, como por exemplo: erros em estimativas, dificuldade no domínio da área de conhecimento específica do software proposto, especificações obscuras, requisitos mal elaborados e mal interpretados, conflitos nos objetivos e mudanças intermináveis e mal controladas.

¹ atorres@dce.ufpb.br

Tais erros aumentaram a importância e responsabilidade dos especialistas ligados a uma das áreas da computação, conhecida como Engenharia de Software. Com isso, o *Institute of Electrical and Electronics Engineers* (IEEE) e a *Association for Computing Machinery* (ACM) conduziram estudos de modo a promover ativamente a Engenharia de Software como uma profissão desde 1993, definindo as fronteiras que delimitam a Engenharia de Software, através do Corpo de Conhecimento em Engenharia de Software - *Software Engineering Body of Knowledge* (SWEBOK). [Swebok, 2004].

Neste capítulo será apresentada uma descrição sobre as áreas do SWEBOK e suas idéias gerais, que embasam, através do seu guia, profissionais, sociedade científica e órgãos públicos ao conhecimento da Engenharia de Software.

11.2. O Projeto do SWEBOK

O SWEBOK surgiu através de uma parceria entre o IEEE, a Computer Society e ACM a fim de promover a profissionalização da Engenharia de Software e criar um consenso sobre as áreas de conhecimento da Engenharia de Software e seu escopo. Sendo iniciado em 1998 pelo Software Engineering Coordinating Committee (SWECC) e financiado por organizações como a ACM, a Boeing, o Conselho Canadense de Engenheiros Profissionais, Construx Software, MITRE Corporation, entre outras.

O SWEBOK é recomendado para diversos tipos de público, em todo o mundo, com o objetivo de ajudar organizações a terem uma visão consistente da Engenharia de Software. É endereçado a gerentes, engenheiros de software, às sociedades profissionais, estudantes, professores e instrutores desta área de conhecimento.

Seus objetivos, conforme SWEBOK (2004) são:

- Oferecer uma visão consistente da Engenharia de Software no âmbito mundial;
- Deixar claros os limites da Engenharia de Software com respeito a outras disciplinas como ciência da computação, gerência de projetos, engenharia da computação, matemática, entre outros;
- Caracterizar o conteúdo da disciplina de Engenharia de Software;
- Prover acesso aos tópicos do corpo de conhecimento da Engenharia de Software;
- Prover uma base para desenvolvimento curricular e para certificação individual;
- Servir como material de apoio.

11.2.1. Categorias do Conhecimento da Engenharia de Software

São consideradas três bases de conhecimento para categorizar o guia SWEBOK, conforme a Tabela 11.1.

Tabela 11.1: Categorias do Conhecimento conforme o SWEBOK

Fonte: [adaptado de SWEBOK, 2004].

Especializado Práticas usadas apenas por alguns tipos de software	Geralmente Aceitas Práticas tradicionais estabelecidas recomendadas pela maior parte das organizações.
	Pesquisa Avançada Práticas inovadoras usadas apenas por algumas organizações com conceitos a serem desenvolvidos e testados em organizações de pesquisa.

O guia, como é convencionalmente chamado, divide a Engenharia de Software em dez áreas de Conhecimento - *KnowLedge Areas* (KAs): (1) requisitos, (2) Projeto, (3) Construção, (4) testes, (5) manutenção, (6) configuração, (7) gerenciamento, (8) processo, (9) ferramentas e métodos, (10) Qualidade além de disciplinas relacionadas à Engenharia de Software, as quais serão explicadas nas subseções a seguir.

É importante comentar que o IEEE oferece duas modalidades de certificação sobre o SWEBOK, disponibilizadas para engenheiros e desenvolvedores.

São elas:

- **Certificação de associados para o desenvolvimento de Software - *Certified Software Development Associate (CSDA)*;**
- **Certificação Profissional para desenvolvimento de Software - *Certified Software Development Professional (CSDP)*.**

Depois de pesquisa extensa, os líderes do IEEE reconheceram a necessidade por um endereçamento do programa de certificação aos iniciantes e profissionais no desenvolvimento de software. O propósito do programa de CSDA é estabelecer um nível mínimo de habilidades e conhecimento, sendo o primeiro passo para se tornar um Profissional de Desenvolvimento de Software Certificado (CSDP).

A certificação CSDP é recomendada para profissionais mais experientes e tem como objetivo ampliar as habilidades e conhecimentos técnicos especializados relevantes sobre o SWEBOK.

11.2.2. Áreas de Conhecimento

O SWEBOK está organizado em dez áreas de conhecimento e um tópico referente às disciplinas relacionadas com a Engenharia de Software. A Figura 11.1 apresenta o corpo de conhecimento do guia, como também seus níveis hierárquicos.

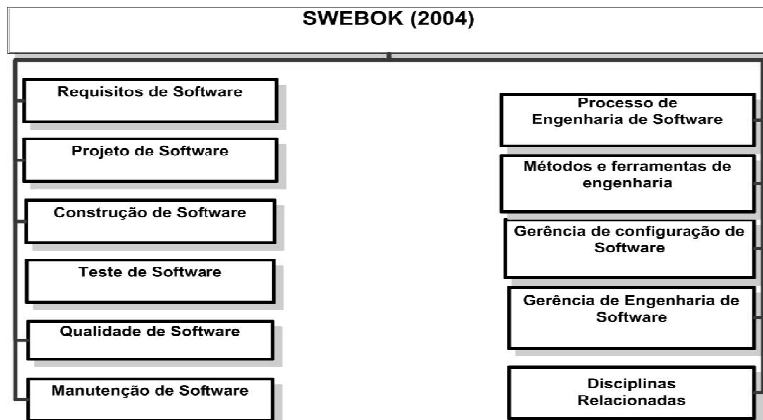


Figura 11. 1: Organização do SWEBOK

Fonte: [adaptado de SWEBOK, 2004].

A seguir serão descritas cada uma das áreas de conhecimento do SWEBOK.

Requisitos de Software

Os requisitos expressam as necessidades e restrições colocadas sobre o produto de software que contribuem para a solução de algum problema do mundo real. Esta área envolve elicitação, análise, especificação e validação dos requisitos de software [Swebok, 2004].

As principais falhas verificadas em projetos de software são relativas aos requisitos, devidas as dificuldades no entendimento das necessidades do usuário. Portanto, realizar corretamente o levantamento e administração de requisitos é essencial para a qualidade de software [Koscianski e Soares, 2007].

A área de requisitos de software está dividida em sete subáreas:

a. Fundamentos dos Requisitos de Software

O SWEBOK define requisitos como propriedades que devem ser exibidas, a fim de resolver algum problema no mundo real [Swebok, 2004].

Esta subárea inclui as definições dos próprios requisitos de software e também dos tipos principais de requisitos, são eles:

- **Requisitos de produto:** são os requisitos que descrevem as propriedades de um sistema ou produto que buscam satisfazer os requisitos do negócio.
- **Requisitos de processo:** são “regras para o desenvolvimento do software”
- **Requisitos funcionais:** são aqueles que descrevem funções que o sistema deve executar, como por exemplo: formatação de textos, adaptação de sinais e cadastros de “entidades”.
- **Requisitos não-funcionais:** são aqueles que descrevem restrições para a **solução**, exemplos: requisitos de performance, requisitos de manutenção, requisitos de segurança; e requisitos de confiança (estabilidade).
- **Requisitos Emergentes:** não são identificados por simples componentes, mas a partir da satisfação de várias dependências de como o software deve operar. Característica: São extremamente dependentes da arquitetura do sistema.

Esta subárea também descreve a importância de requisitos quantitativos e distingue-os entre requisitos de sistema e requisitos de software.

b. Processo de Requisitos

O processo de requisitos de software descreve o planejamento de requisitos, de forma a efetuar o entendimento das outras cinco subáreas com o processo completo de planejamento de software. Esta subárea mostra como os processos de requisitos se integram com os processos de Engenharia de Software, sendo dividida em quatro subáreas:

- **Modelos de processos:** o tema está preocupado com a forma que as atividades de análise de levantamento de requisitos, especificação e validação estão configurados para diferentes tipos de projetos e restrições. O tema inclui também atividades que contribuem para o processo de requisitos, tais como marketing e estudos de viabilidade.
- **Atores dos processos:** este tópico apresenta os papéis das pessoas que participam no processo de requisitos. Há frequentemente muitas pessoas envolvidas, além do especialista, mas sempre incluem usuários, operadores e clientes (que não precisa ser o mesmo).
- **Processo de suporte e gestão:** este tópico apresenta os recursos do gerenciamento de projetos necessários e consumidos pelo processo de requisitos. Seu objetivo principal é fazer a conexão entre as atividades do processo identificado na subárea, modelos de processo, e as questões de custos, recursos humanos, treinamento e ferramentas.
- **Processos de Qualidade e Melhoria:** Esta subárea preocupa-se com a avaliação da qualidade e da melhoria do processo de requisitos. Seu objetivo é enfatizar o papel fundamental do processo de execuções dos requisitos em termos de custo e oportunidade de um produto de software e de satisfação do cliente.

c. Elicitação de Requisitos

A elicitação de requisitos efetua a captura, descoberta e aquisição dos requisitos de software pelo engenheiro de software. Identifica as fontes dos requisitos e define as técnicas para extraí-los. É o primeiro estágio para o entendimento do problema disposto. Está dividida em duas subáreas:

- **Fontes de requisitos:** promove a conscientização das diversas fontes de requisitos de software e dos quadros de gestão. Os principais pontos abordados são: Objetivos, Domínio do conhecimento, *Stakeholders*², ambiente operacional e organizacional
- **Técnicas de elicitação:** este tópico concentra-se em técnicas para obter das partes interessadas as suas necessidades.

d. Análise de Requisitos

A análise de requisitos é uma tarefa da engenharia de software que efetua a ligação entre a alocação de software em nível de sistema e o projeto de software, possibilitando que o engenheiro aprimore e construa modelos do processo, dos dados e dos domínios comportamentais que serão tratados pelo software. [Pressman, 1995].

² Termo usado em administração que se refere a qualquer pessoa ou entidade que afeta ou é afetada pelas atividades de uma empresa.

(Pressman,2002) cita que “se você não analisa, é altamente provável que construa uma solução de software muito elegante que resolve o problema errado”. Uma vez que cada pessoa envolvida na construção do software contribui efetuando uma “pequena” modificação, a cada estágio a especificação do produto acabou diferente da existente no estágio anterior [Koscianski e Soares, 2007]. Esta atitude pode resultar em perda de tempo e dinheiro, pessoas frustradas e clientes insatisfeitos, conforme ilustrado na Figura 11.2.

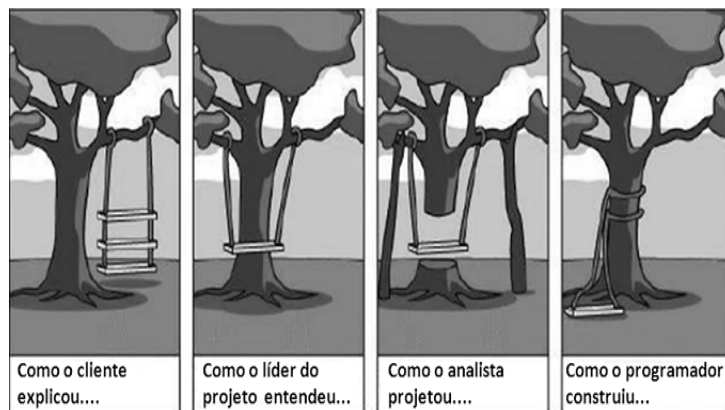


Figura 11. 2: Caricatura dos problemas de comunicação na produção de software

Fonte: [Autoria desconhecida].

Alguns cuidados devem ser tomados para descrever com precisão os requisitos. Esta subárea preocupa-se em detectar e resolver conflitos entre requisitos, descobrir as fronteiras do software e como interagirá com o ambiente, além de aprimorar os requisitos do sistema para derivar requisitos de software. Está dividida em quatro subáreas:

- **Classificação de requisitos:** classifica os requisitos conforme sua exigência, podendo ser: funcional, não funcional, produto, processo, prioridade, escopo e volatilidade/estabilidade.
- **Modelagem conceitual:** sua finalidade é auxiliar na compreensão do problema, ao invés de dar início ao projeto. Os modelos conceituais incluem modelos de entidades do domínio do problema configurado para refletir as suas relações no mundo real e suas dependências.
- **Projeto arquitetural e Distribuição de requisitos:** é o ponto no qual são descritos os requisitos do projeto de software, ilustrando as diferenças entre estas duas tarefas.
- **Negociação de requisitos:** também conhecida como “resolução de conflitos”. Trata-se de resolver os problemas com as exigências, onde ocorrem conflitos entre as duas partes que requerem recursos incompatíveis entre si, entre as necessidades e recursos, ou entre funcionais e requisitos não-funcionais, por exemplo.

e. Especificação de Requisitos

Para muitos profissionais da engenharia, conforme o SWEBOK, especificação refere-se à atribuição de valores numéricos ou limites para os objetivos do projeto. No entanto, a principal atividade desta fase é a confecção da documentação do sistema, que pode ser sistematicamente revisada, validada e aprovada, especificando os componentes de software. Está é dividida em três subáreas:

- **Documentação de definição do sistema:** também conhecido como documento de requisitos do usuário, sendo descrito a definição dos requisitos do sistema de alto nível a partir da perspectiva de domínio. Seus leitores incluem os representantes dos usuários do sistema (clientes). O documento lista os requisitos de sistema, juntamente com informações básicas sobre os objetivos gerais para o sistema, seu ambiente de destino e um mapa de restrições, premissas e requisitos não-funcionais.
- **Especificação dos requisitos do sistema:** Neste tópico os requisitos do sistema são especificados, O detalhamento deste tópico está fora do escopo do guia, sendo descrito como uma atividade da engenharia de sistemas.
- **Especificação dos requisitos do software:** estabelece as bases para um acordo entre os clientes e fornecedores para o produto de software em desenvolvimento, bem como o que não faz parte do seu escopo. Para o documento de especificação de requisitos de software é recomendado um documento de definição de requisitos de software, fornecendo uma base realista para estimar os custos dos produtos, riscos e cronogramas.

f. Validação de Requisitos

A documentação de requisitos pode ser objeto de validação. Tal atividade busca a conformidade do documento com os padrões da organização. Sendo o seu objetivo o de enumerar todos os problemas antes de alocar recursos para resolver os requisitos.

Esta subárea possui as etapas de:

- **Revisão dos requisitos:** é o meio mais comum de validação. Nesta atividade um grupo de revisores efetua uma avaliação preliminar de erros, com o desenvolvimento de *checklists*³ de revisão.
- **Prototipação:** prototipagem é normalmente um meio para validar a interpretação do desenvolvedor sobre os requisitos de software. Neste tópico é realizado o levantamento da técnica de prototipagem e dos pontos no processo onde a validação do protótipo pode ser apropriada.
- **Validação de modelos:** este tópico é necessário para validar a qualidade dos modelos desenvolvidos durante a análise.
- **Testes de aceitação:** é realizado o projeto para os testes dos requisitos de software para aceitação. Informações adicionais são comentadas na área de conhecimento Teste de Software.

³ Documento com itens de verificação e acompanhamento detalhado de ações num projeto.

g. Considerações Práticas

Esta subárea descreve os tópicos que necessitam ser compreendidos na prática, validando os atributos dos requisitos como também tem o papel de avaliar o tamanho das mudanças nos requisitos e estimar os custos do desenvolvimento e manutenção.

O primeiro tópico é a natureza iterativa do processo dos requisitos. Os três tópicos seguintes são fundamentalmente sobre gerência da mudança e a manutenção dos requisitos em um estado que espelhe exatamente o software a ser construído. Incluem na gerência da mudança, os atributos dos requisitos e a rastreabilidade dos requisitos. O tópico final é sobre processos de medição dos requisitos.

As principais responsabilidades das subáreas de requisitos de software, segundo o SWEBOK (2004) são:

- Preocupação com a origem dos requisitos e como os engenheiros de software podem coletá-los. Inclui fontes e técnicas de levantamento de requisito.
- Descrever a importância dos requisitos quantificáveis, e distinguir entre sistemas e requisitos de software.
- Demonstração de como o planejamento de requisitos se encaixa com o processo completo de planejamento de software.
- Preocupação com os modelos de processo, atores, suporte, gerenciamento de requisitos, melhoria e qualidade do processo.

Projeto de software

O Projeto de software ou Design de Software é a atividade do ciclo de vida da Engenharia de Software em que os requisitos do são analisados a fim de produzir uma descrição da estrutura interna do software que servirá como base para sua construção [Swebok, 2004].

Esta é uma área de grande importância, pois compreende tanto a definição da arquitetura, componentes, relações, e outras características de um sistema ou de um componente quanto o resultado do próprio processo [IEEE, 1990].

A área de projetos de software está dividida em seis subáreas:

a. Fundamentos do Projeto de Software

São os conceitos, noções e terminologias introduzidas como base fundamental para a compreensão do papel do projeto de software.

b. Questões Chave no Design de Software

Trata dos assuntos que devem ser abordados no projeto de software, tais como: controle e tratamento de eventos, concorrência, tratamentos de erros e de exceções, entre outros.

c. Estrutura e Arquitetura de Software

Descreve a estrutura, estilo, padrões e frameworks utilizados para a arquitetura funcional do software.

d. Evolução e Análise de Qualidade do Projeto de Software

Descreve tópicos relacionados com a qualidade de software, como: métricas, avaliação de ferramentas e características de qualidade.

e. Notações do Projeto de Software

Apresenta notações estruturais (estáticas) e comportamentais (dinâmicas).

f. Estratégias e Métodos para o Projeto de Software

Descrição de método para o projeto de software, bem como de um conjunto de orientações na utilização de tais métodos. São eles: métodos orientados a funções, objetos, formais e transformacionais.

Construção de software

Área envolvida com todas as áreas de conhecimento do guia, entretanto existe um relacionamento maior com o projeto de software e com teste de software. Fazem parte desta área as atividades de implementação do software, verificação, testes de unidade, teste de integração e depuração. Isso ocorre porque o processo de construção do próprio design de software envolve vários testes das suas atividades [Swebok, 2004].

As áreas correlatas à construção de software, segundo o SWEBOK (2004) são:

a. Fundamentos da Construção de Software

Tem como objetivos principais: minimizar a complexidade, antecipar as mudanças, efetuar a verificação e definir os padrões para a construção de software.

b. Gerenciamento da Construção de Software

Efetua o planejamento e avaliação da construção do software, como também informa os modelos para tal atividade. Exemplos dos modelos: Prototipação, Cascata, Espiral, etc.

c. Considerações Práticas da Construção de Software

Nessa subárea são descritas atividades práticas como: Projeto de Construção, Linguagem Própria, Codificação, Teste, Construção, Reuso, Qualidade, Integração.

O guia SWEBOK recomenda que nesta etapa as funcionalidades do software sejam testadas durante todo o processo de desenvolvimento, não deixando apenas para a etapa de testes.

Teste de software

Nas décadas de 60 e 70, os desenvolvedores dedicavam a maior parcela dos seus esforços nas atividades de codificação e nos testes unitários. Sendo apenas uma parcela menor dedicada à integração dos programas e nos testes dos sistemas, pois as atividades de testes eram consideradas um mal necessário e não eram tratadas como um processo formal alinhado ao processo de desenvolvimento dos sistemas. [Rios e Moreira, 2006].

Nesta seção é feito apenas um breve comentário sobre a área de conhecimento de teste de software conforme o SWEBOK, devido ao seu embasamento teórico já está detalhado no capítulo 10 deste livro.

O teste é uma atividade realizada para avaliação da qualidade do produto, efetuando sua melhoria através da identificação de defeitos e problemas [Swebok, 2004].

O destaque crescente do software como elemento de sistema e os “custos” envolvidos associados às falhas de software são forças propulsoras para uma atividade de teste cuidadosa e bem planejada [PRESSMAN, 1995].

Esta área, conforme o SWEBOK consiste na verificação dinâmica do comportamento de um programa com um conjunto finito de casos de testes, selecionados de um domínio geralmente infinito de execuções, para confirmar o comportamento especificado esperado.

São subáreas dessa área de conhecimento: (1) Fundamentos do teste de software, (2) Níveis de Teste de Software, (3) Técnicas de teste de software, (4) Medidas de teste de software, (4) Processo de teste e (5) Considerações práticas.

Considera-se atualmente que a atitude certa para qualidade é a de prevenção, sendo muito melhor evitar problemas do que corrigi-los. Mas ao se debater com falhas em software já concluído, um bom plano de manutenção corretiva é um remédio a ser planejado pela equipe. Na próxima seção é abordada a manutenção de software como uma estratégia descrita no SWEBOK para uma busca contínua da qualidade de software.

Manutenção de software

O desenvolvimento de software requer esforços que visam à entrega de um produto que satisfaça aos requisitos do cliente. Nesta área de conhecimento sua principal responsabilidade é totalizar as atividades requeridas para fornecer suporte custo-efetivo a um sistema de software, que pode ocorrer antes ou depois da entrega. Antes da entrega do software são realizadas atividades de planejamento e depois, as modificações são feitas com o objetivo de corrigir as falhas, melhorar o desempenho ou adaptá-las a um ambiente externo [Swebok, 2004].

No guia, a manutenção de software está relacionada com todos os aspectos da Engenharia de Software, sendo assim, ligada a todas as áreas do SWEBOK. Sendo seu objetivo principal sustentar o produto ao longo do seu ciclo de vida operacional.

O fluxo de eventos que pode ocorrer como resultado de um pedido de manutenção é ilustrado na figura 11.3. Onde, se o único elemento disponível de uma configuração de software for o código-fonte, a atividade de manutenção inicia-se com

uma penosa avaliação do código, etapa essa dificultada pela freqüente documentação interna encontra-se num estado ruim [Pressman, 2002].

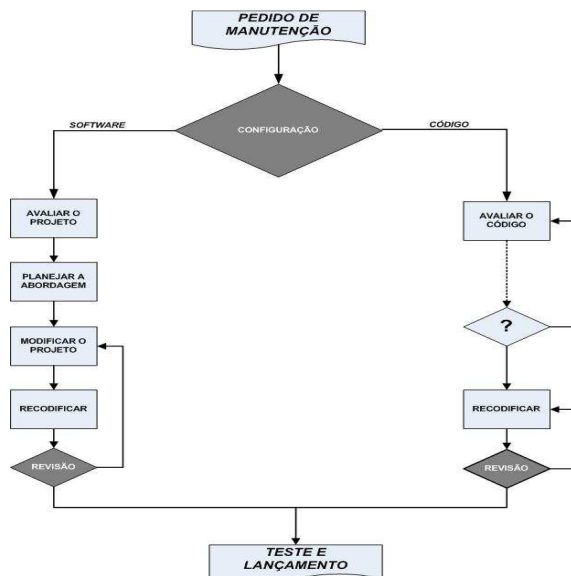


Figura 11. 3: Manutenção estruturada *versus* não estruturada

Fonte: [Pressman, 2002].

As áreas correlatas à manutenção de software, segundo o SWEBOK (2004) são:

a. Fundamentos da Manutenção de Software

Apresenta os conceitos e terminologias que formam a base de conhecimento para compreensão do papel da manutenção de software, como por exemplo, a natureza, categorias, custos, evolução de software, entre outras.

b. Questões Chave sobre Manutenção de Software

Apresenta as questões chaves relacionadas com a manutenção de software, agrupadas como: problemas técnicos (compreensão limitada, teste, análise de impactos e manutenção), questões de gestão (alinhamento com os objetivos organizacionais, processo, aspectos organizacionais de manutenção e outsourcing⁴), estimativas de custos e medidas (parâmetros e experiência).

c. Processo de Manutenção de Software

Fornecer referências e padrões utilizados para implementar a manutenção de software, relacionando com outras atividades da Engenharia de Software. A figura 11.4 ilustra o processo de manutenção de software conforme a ISO/IEC 14764, que é similar a IEEE 1219 (atividades de manutenção de software).

⁴ Uso estratégico de recursos externos para a realização de atividades tradicionalmente realizadas pelos recursos e equipes internos.

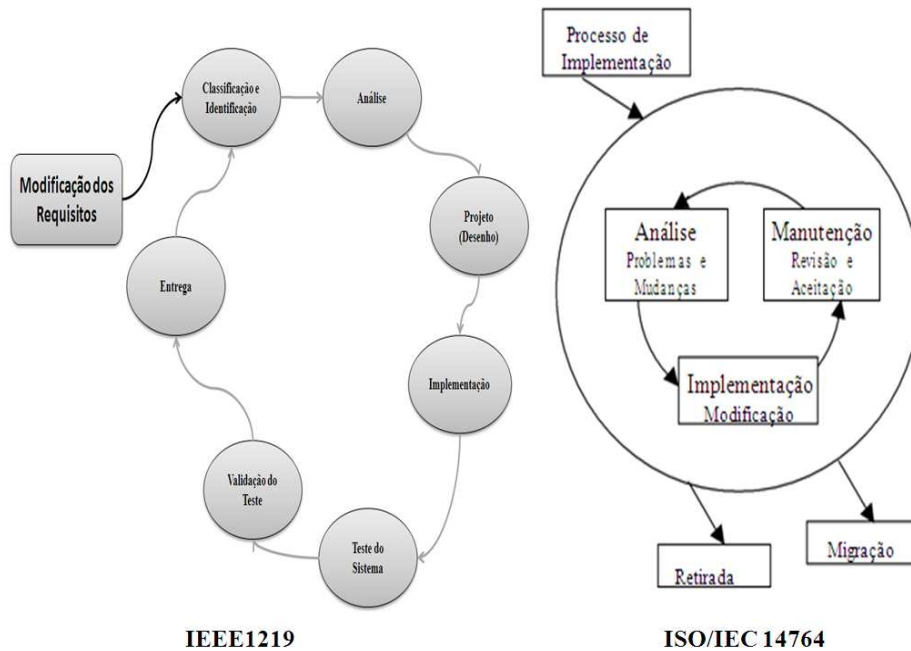


Figura 11. 4: Atividades do processo de manutenção (IEEE1219) *versus* Processo de Manutenção de software (ISO/IEC 14764).

Fonte: [Adaptado SWEBOK, 2004].

d. Técnicas de Manutenção de Software

Descreve técnicas de manutenção, por exemplo, compreensão do programa, reengenharia, engenharia reversa.

Em última análise, algumas organizações de software podem permanecer voltadas à manutenção, incapazes de embarcar em novos projetos, porque todos os seus recursos são dedicados à manutenção de velhos programas.

Gerência de configuração de Software

A Gerência de Configuração de Software – Software Configuration Management (SCM), conforme o SWEBOK(2004) tem o papel de identificar a configuração de um sistema em pontos distintos no tempo com a finalidade de controlar sistematicamente as mudanças para configurar e manter a integridade e rastreabilidade de todos os artefatos ao longo do ciclo de vida do sistema [Swebok, 2004].

São atividades da SCM:

a. Processo de Gerenciamento de Configurações

Controla a evolução e integridade de um produto. Do ponto de vista do engenheiro de software, uma implementação bem sucedida exige um SCM cuidadoso, sendo preciso o seu processo.

b. Identificação da Configuração

Esta subárea tem papel de controlar e administrar itens de configuração de software (Informação que é criada como parte do processo de Engenharia de Software), cada um deve ser nomeado separadamente. Suas principais atividades são: identificação dos itens a serem controlados (configuração de software, itens de configuração do software, relacionamento entre os itens de configuração, versões de software, Baseline e aquisição dos itens de configuração de software).

c. Controle da Configuração

Abrange a gestão durante o ciclo de vida do software, como por exemplo: processo para determinar quais as mudanças a serem realizadas, a autoridade para aprovar algumas alterações, apoio a implementação dessas mudanças, entre outras atividades. Uma Ordem de Mudança da Engenharia – Engineering Change Order (ECO) é gerada para cada mudança aprovada. A ECO descreve as mudanças a serem feitas, as restrições que devem ser respeitadas e os critérios de revisão e auditoria. Informações obtidas a partir dessas atividades são úteis para a medição do tráfego da mudança e dos aspectos do retrabalho.

d. Registro do Estado da Configuração

É o registro e comunicação das informações necessárias para uma gestão eficaz da configuração de software. Desempenha um papel vital no sucesso de um grande projeto de desenvolvimento de software. São perguntas importantes dessa fase: O que aconteceu? Quem o fez? Quando aconteceu? O que mais será afetado?

e. Auditoria da Configuração

O SWEBOK detalha essa atividade baseado na norma IEEE 1028 e tem o objetivo de avaliar a conformidade dos produtos de software e processos de regulamento. As auditorias são conduzidas pelo processo que consiste na descrição dos papéis, contas e responsabilidades. Conforme (Pressman 1995), uma auditoria de configuração de software pergunta e responde às seguintes questões:

- A mudança especificada na ECO foi feita? Outras modificações adicionais foram incorporadas?
- Uma revisão técnica formal foi realizada para avaliar a exatidão técnica?
- Os padrões de Engenharia de Software foram adequadamente seguidos?
- A mudança foi “realçada” no SCI? A data da mudança e o autor da mudança foram especificados? Os atributos do objeto de configuração refletem a mudança.
- Os procedimentos de SCM para anotar a mudança, registrá-la e relatá-la foi seguido?
- Todos os SCIs relacionados foram adequadamente atualizados?

f. Entrega e Gerenciamento de Versões

A liberação é utilizada neste contexto para se referir a entrega de um item de configuração, incluindo a liberação interna. Quando as versões de diferentes itens de software estão disponíveis para entrega é frequentemente necessário recriar versões específicas de pacotes e os materiais corretos para entrega da versão.

É importante fazer uma distinção entre manutenção de software e gerenciamento de configuração de software. A manutenção é um conjunto de atividades de Engenharia de Software que acontece depois da entrega do software enquanto ele é colocado em operação, diferentemente do gerenciamento de configuração de software, que já efetua o controle e rastreamento do projeto de desenvolvimento de software desde o início, terminando quando o software é tirado de operação.

Gerência de Engenharia de Software

A área de Gerenciamento de Software é uma área que influencia e recebe influência das outras áreas da Engenharia de Software, sendo desta forma fundamental para atingir um bom resultado ao final do projeto. É definida como a aplicação da gestão de atividades de planejamento, coordenação, medição, monitoramento, controle e comunicação de forma a garantir a manutenção sistemática e desenvolvimento dos projetos de software [Swebok, 2004].

No que se diz respeito à Engenharia de Software, o gerenciamento de software ocorre em três níveis: gerenciamento organizacional, gerenciamento de projeto e controle e planejamento de programas de medição. Sendo apenas, os dois últimos abordados em detalhes nesta seção.

Outro aspecto importante do gerenciamento são as atividades relacionadas com o gerenciamento de pessoas, levando em consideração equipe, clientes e responsáveis da própria organização. Além disso, são necessários, os planejamentos e gerenciamentos da comunicação entre essas pessoas para melhorar o entendimento de todos, fator primordial para um bom resultado final.

A área de “Gerenciamento da Engenharia de Software” se divide em sete subáreas:

a. Iniciação e Definição de Escopo

O foco é determinar os requisitos através de vários métodos de levantamento, avaliação do projeto e especificação dos requisitos e procedimentos para validação de mudanças. O guia destaca os seguintes tópicos relacionados a esta subárea: determinação e negociação dos requisitos, análise de viabilidade, revisão dos requisitos e do processo de software.

b. Planejamento do Projeto

Neste ponto, o ciclo de vida dos processos de software é avaliado para o melhor planejamento, levando em conta fatores como: natureza do projeto, complexidade funcional e técnica, requisitos de qualidade, entre outros. Tal subárea é um fator importante para o sucesso da qualidade de software. Os tópicos relacionados são: planejamento do processo, esforço, previsão de custo, alocação de recursos, gerenciamento de risco e da qualidade.

c. Declaração do Plano de Projeto

A declaração do plano de projeto tem o papel de descrever o plano a ser implementado e os processos a ele incorporados, com a expectativa que a sua adesão poderá levar ao sucesso e satisfação dos requisitos do cliente. São tópicos abordados nesta subárea: implementação do plano, contrato dos fornecedores da gestão, processo de medição, monitoramento e controle dos processos.

d. Revisão e avaliação

Esta atividade descreve as avaliações na busca da eficácia do processo global do projeto, contendo informações como: datas, pessoas envolvidas, ferramentas e métodos utilizados. Os objetivos principais são: determinar a satisfação dos requisitos, rever e avaliar o desempenho.

e. Fechamento

O projeto é caracterizado como concluído ou passa para o status de fechamento, quando todos os planos e processos foram homologados e completados. A se estabelecer o fechamento é iniciada a execução das atividades de melhoria dos processos.

Fazem parte das atividades desta subárea as atualizações de dados e arquivamento de documentos.

f. Evolução da Engenharia de Software

A importância da medição e seu papel nas melhor práticas de gestão são amplamente reconhecidos, e assim a sua importância só pode aumentar nos próximos anos. Medir a eficácia tornou-se um dos pilares da maturidade organizacional.

Um fator importante com relação a esta área é que, como pôde ser observada, ela possui atividades distribuídas durante todo o ciclo de vida do projeto. Isto significa que ela está ligada a praticamente todas as atividades que acontecem durante o projeto e, caso as atividades do gerenciamento estejam comprometidas, as atividades de qualquer fase do ciclo de vida podem ser afetadas reduzindo a qualidade de seus resultados finais.

Processo de Engenharia de Software

A Tecnologia de Processo de Software surgiu no final da década 80 e representou um importante passo em direção à melhoria da qualidade de software através de mecanismos que proporcionam o gerenciamento automatizado do desenvolvimento de software. Diversas teorias, conceitos, formalismos, metodologias e ferramentas surgiram nesse contexto, enfatizando a descrição de um modelo de processo de software que é automatizado por um ambiente integrado de desenvolvimento de software [Reis e Nunes, 2002].

A área de conhecimento do processo de Engenharia de Software está relacionada com a definição, implementação, controle, e proposta de mudança no próprio processo. Esta área pode ser visualizada em dois níveis. O primeiro nível engloba as atividades técnicas e gerenciais executadas durante a aquisição, desenvolvimento e manutenção do software. O segundo nível, tratado nesta área de conhecimento, considera as definições, implementações, gerenciamento e mudanças no próprio processo.

O processo de Engenharia de Software envolve vários outros processos, como o de desenvolvimento o de gerenciamento, e o de qualidade. Esta área está ligada com qualquer parte do gerenciamento do processo de ciclo de vida do software, onde mudanças são propostas com o intuito de melhorar o produto ou até mesmo o processo de produção.

Ao contrário do que é suposto, o processo de engenharia de software é importante não apenas para empresas grandes, mas também devem ser levadas em consideração para pequenas empresas, auxiliando no seu desenvolvimento. O objetivo desse gerenciamento do processo é implementar novas práticas individuais, de projeto ou até mesmo organizacionais.

A área de processos de Engenharia de Software está dividida em quatro subáreas:

a. Mudança e Implementação do Processo

Esta atividade descreve a infra-estrutura, atividades, modelos, práticas e considerações sobre a implementação do processo e da mudança.

b. Definição do Processo

O guia define como uma política, um procedimento ou um padrão. Variáveis importantes a considerar incluem a natureza do trabalho, como por exemplo, a manutenção ou desenvolvimento. Importante salientar que o contexto do projeto e a organização irão determinar o tipo de processo que é mais adequado.

c. Avaliação do Processo

Existem duas formas de avaliação para fazer as suposições sobre as ordens dos processos (contínuos ou escalonados), onde a organização define qual a mais pertinente para suas necessidades e objetivos.

d. Medidas de Produtos e Processos

Embora a aplicação de medidas de Engenharia de Software possa ser complexa, especialmente em termos de modelagem e métodos de análise, existem vários aspectos de medição que são fundamentais por trás de muitas medidas avançadas e processos de análise. O guia traz como palavra chave para essa subárea a norma ISO/IEC 15939⁵ para descrever tais medidas e métodos para produtos e processos.

Uma das grandes dificuldades enfrentadas por esta área é que a implementação das suas práticas geralmente não traz benefícios em curto prazo, porém com a evolução dos processos, a empresa vai aumentando o seu nível de maturidade, permitindo que o desempenho das equipes e a qualidade final dos produtos sejam beneficiados.

Métodos e ferramentas de engenharia

No SWEBOK, as ferramentas de desenvolvimento de software são ferramentas criadas para auxiliar o ciclo de vida do software. Essas ferramentas normalmente automatizam algumas atividades do processo de desenvolvimento, fazendo com que o

⁵ Norma referente ao processo de medição na Engenharia de software e de sistemas.

analista concentre-se nas atividades que exigem maior trabalho intelectual [Swebok, 2004].

A subárea Métodos de Engenharia de Software impõe uma estrutura sobre a atividade de desenvolvimento e manutenção de software com o objetivo de torná-la sistemática e mais propensa ao sucesso [Swebok, 2004].

O objetivo desta área de conhecimento é pesquisar ferramentas e métodos que aumentem a produtividade dos desenvolvedores enquanto reduzem a ocorrência de falhas no desenvolvimento [Fernandes, 2003].

Embora existam manuais detalhados de ferramentas e inúmeros trabalhos de instrumentos inovadores, uma das dificuldades é a elevada taxa de mudança de ferramentas de software em geral [Swebok, 2004].

a. Ferramentas de Engenharia de Software

Cada tópico ilustrado na Figura 11.6 é referente em todos os KA's do guia, possuindo um tema adicional que aborda técnicas e ferramentas de integração, que são potencialmente aplicáveis a todas as classes de ferramentas.

b. Métodos de Engenharia de Software

É dividida em três temas:

- Heurísticos: abordagem informal;
- Matemáticos: abordagem formal;
- Protótipos: para a abordagem do software baseado em telas.

Qualidade de Software

O SWEBOK faz uma distinção entre técnicas estáticas e dinâmicas. As primeiras aparecem sob a área de conhecimento Qualidade, enquanto as últimas figuram na área de Testes. A norma internacional ISO/IEC 25000 SQuaRE, detalhada no capítulo 10, trata da qualidade de produtos de software, abrangendo estes dois tópicos [Koscianski, 2006].

A qualidade é relativa. O que é qualidade para uma pessoa pode ser falta de qualidade para outra. [Weinberg, 1994].

A ISO 9001, conforme comentado no capítulo 7, define a qualidade como "o grau em que um conjunto de características inerentes satisfaz as necessidades" [Swebok, 2004].

Um dos principais objetivos da Engenharia de Software é melhorar a qualidade dos produtos de software, ela visa estabelecer métodos e tecnologias para construir produtos de software de qualidade dentro dos limites de tempo e recursos disponíveis. A qualidade de software está diretamente ligada com a qualidade do processo através do qual o software é desenvolvido, portanto, para se ter qualidade em um produto de software é necessário ter um processo de desenvolvimento bem definido, que deve ser documentado e acompanhado [Swebok, 2004].

A avaliação da qualidade de produtos de software normalmente é feita através de modelos de avaliação de qualidade. Esses modelos descrevem e organizam as

propriedades de qualidade do produto em avaliação. Os modelos de avaliação mais aceitos e usados no mercado são:

- CMMI (*Capability Maturity Model Integration*), proposto pelo CMM (*Capability Maturity Model*), tal modelo foi detalhado no capítulo 8.
- Norma ISO/IEC 9126, proposta pela ISO (*International Organization for Standardization*), tal modelo foi detalhado no capítulo 10.

As organizações desenvolvedoras destes modelos de qualidade fornecem selos para as empresas que se submetem as avaliações e que estejam dentro dos padrões propostos. Esses selos são muito valorizados pelas empresas que buscam novas soluções e software com qualidade, e representam um diferencial competitivo no mercado. Porém, nem todas as empresas têm condições financeiras de bancar os custos de uma aquisição de um selo de qualidade, pois implantar um plano de gestão da qualidade em uma empresa envolve custos elevados. Contudo, é possível implantar boas práticas e desenvolver um processo de desenvolvimento organizado adaptando modelos de desenvolvimento conhecidos, despendendo menos recursos e provendo um mínimo de sistematização no desenvolvimento de software, a fim de se ter maior qualidade.

A área de Qualidade de Software, segundo o SWEBOOK, é dividida em três tópicos que serão rapidamente descritos a seguir:

a.Fundamentos da Qualidade de Software

Este tópico abrange a definição de qualidade, buscando acordar os requisitos da qualidade, bem como efetuar uma comunicação clara com o engenheiro de software sobre tais requisitos.

Este tópico tem com subáreas:

- **Engenharia de Software Cultura e Ética:** é detalhado o código de ética e prática profissional desenvolvido pelo IEEE *Computer Society* e a ACM, com base em oito princípios para ajudar engenheiros de software e reforçar atitudes relacionadas à qualidade e à independência do seu trabalho.
- **Valor e custos da qualidade:** o tópico detalha as diferenças dos tipos de custo da Qualidade, podendo ser: prevenção custos, custos de avaliação, custos de falhas internas e externas insuficiência de custo.
- **Modelos e características da qualidade:** a norma ISO / IEC 9126 e ISO/IEC14598 definem três modelos relacionados à qualidade dos produtos de software (interno de qualidade de qualidade, externa e qualidade em uso) e um conjunto de partes relacionadas, tais normas são detalhadas no capítulo 10.
- **Melhoria da qualidade:** o tópico descreve um processo iterativo de melhoria contínua, que exige controle da gestão, coordenação e feedback de muitos processos simultâneos: (1) o ciclo de vida do processo de software, (2) o processo de detecção dos erros / defeitos e (3) o processo de melhoria da qualidade.

b. Processo de Gerenciamento da Qualidade de Software

Tal tópico abrange todos os aspectos de construção do produto. São abordados todos os elementos de um projeto, como: ferramentas para controle de versão e linguagens, metodologias para revisão do produto, técnicas organizacionais e de administração de pessoas etc.

O propósito da subárea é assegurar que os objetivos planejados no início do projeto sejam cumpridos, se aplicando a todas as perspectivas do processo de software. Definindo processos, responsáveis, produtos e recursos.

Alguns dos processos desta subárea são definidos pela norma ISO/IEC 12207, sendo eles: garantia de qualidade, verificação, validação, revisão e auditoria.

c. Considerações Práticas sobre a Qualidade de Software

Neste tópico são apresentadas as recomendações gerais sobre como transcorre a execução das atividades relacionadas com qualidade.

Na estrutura deste tópico estão as subáreas:

- Requisito de qualidade de software: menciona itens como “fatores de influência” sobre requisitos, segurança do funcionamento e as conseqüências que as falhas podem causar;
- Caracterização de defeitos: verifica a não-conformidade aos requisitos;
- Técnicas de gestão de qualidade de software: podem ser orientadas a pessoas (revisões e auditorias), estáticas (não envolvem execução do produto), dinâmicas (efetuados durante a execução do produto) e técnicas analíticas (utilização de métodos formais).
- Medição da qualidade: inclusão de medidas para determinar o grau de qualidade atingido pelo produto. Como por exemplo, o proposto pela norma SQuaRE⁶, onde os valores desejados para as medidas sejam estabelecidos no início do projeto, ao se definir os requisitos.

Disciplinas relacionadas

Para delimitar a Engenharia de Software se faz necessário identificar as disciplinas com que ela compartilha uma fronteira comum. Nesta seção é feita a identificação em ordem alfabética das disciplinas relacionadas, conforme ilustrada na Figura 11.5.

⁶ Acrônimo de Software Quality Requirements and Evaluation, foi desenvolvido pelo grupo de trabalho WG6 do Subcomitê de Sistemas e Software (SC7) da ISO/IEC.



Figura 11. 5: Disciplinas relacionadas com Engenharia de Software

Fonte: [adaptado de SWEBOK, 2004].

a. Engenharia da Computação

Conforme relatório do Computing Curricula 2001 Computer Science a engenharia da computação incorpora a tecnologia e ciências de concepção, construção, implementação e manutenção de componentes de software e hardware dos sistemas de computação e controlados por computador. Destacam-se como áreas de conhecimento: algoritmos, arquitetura e organização de computadores, engenharia de sistemas de computadores, entre outras.

b. Ciências da Computação

O relatório final Computing Curricula 2001 Computer Science identifica diversas áreas de conhecimento que estão relacionadas com a ciência da computação, tais como: sistemas operacionais, linguagem de programação, computação gráfica, engenharia de software, entre outras.

c. Gerenciamento

Segue recomendações para *Master of Business Administration (MBA)*⁷ definidos pelo Conselho Europeu, que inclui como áreas de conhecimento relacionadas: contabilidade, finanças, marketing, direito, gestão de recursos humanos, entre outras.

⁷ Curso de formação de executivos, nas diversas disciplinas da administração.

d. Matemática

É recomendado para o engenheiro de software, conforme SWEBOK, os conhecimentos de álgebra linear, equações diferenciais, probabilidade, estatística, entre outras.

e. Gestão de Projetos

Segue as recomendações conforme o guia PMBOK⁸ Guide 2006. São conhecimentos relacionados pelo guia: gestão de custos, de riscos, de qualidade, entre outros.

f. Gestão de Qualidade

A gestão da qualidade é definida na norma ISO/IEC 9000. O guia recomenda o conhecimento para a ISO/IEC 9000, ISO/IEC 9001, ISO/IEC 9004 para fábricas de software.

g. Ergonomia

É uma disciplina científica relacionada com a compreensão das interações entre humanos e outros elementos de um sistema. Tem como conhecimentos relacionados: computação gráfica, processo de desenvolvimento, técnicas de aplicação, entre outros.

h. Engenharia de Sistemas

O Conselho Internacional de Engenharia de Sistemas (INCOSE) afirma que “Engenharia de Sistemas é uma abordagem interdisciplinar que permite a realização de sistemas bem sucedidos”. São conhecimentos relacionados: verificação das necessidades dos clientes, funcionalidade necessárias no início do ciclo de desenvolvimento, documentação de requisitos, entre outros.

11.2.2. SWEBOK 2010

Novos estudos estão sendo realizados para a atualização do Guia de Conhecimento da Engenharia de Software - Guide to the Software Engineering Body of Knowledge (SWEBOK) com a intenção de incluir novas área de conhecimento e rever outras.

O objetivo principal da atual revisão do Guia SWEBOK é a adição de um KA sobre as práticas profissionais, um assunto atualmente abrangido pela certificação CSDP, além de acrescentar KA's sobre assuntos relacionados com os engenheiros de software e educação para graduação. Além de:

- Remoção de três disciplinas relacionadas: Ciência da Computação, Matemática e Ergonomia;
- Adicionado material sobre Interfaces Humano-Computador no design de software e Teste de Software;
- Remoção da seção Ferramentas e métodos de Engenharia de Software (distribuídos para outras áreas de conhecimento);

⁸ Guia do conjunto de conhecimentos em gerenciamento de projetos desenvolvido pelo PMI

- Redistribuição de matérias entre as áreas de conhecimento;

A atualização completa do SWEBOK Guide 2010 ainda não foi concluída, mas tem previsão para o final de 2010.

11.3. Tópicos de pesquisa

Os temas de pesquisa relacionados a aplicação do SWEBOK para a formação de graduados em Engenharia de Software, como por exemplo: Universidade de New South Wales (Austrália), Universidade McMaster (Canadá), o Instituto de Tecnologia de Rochester (E.U.) e a Universidade de Sheffield (Reino Unido).

Dentro das aplicações do SWEBOK por instituições pelo mundo, podemos citar:

- Canadian Information Processing Society: publicação dos critérios de aceitação da Engenharia de Software por programas universitários de graduação na área.
- A norma ISO 9000 (Gestão da Qualidade) aplicados à Engenharia de Software (ISO/IEC 90003).
- Efetivação de licenças de engenheiros de software baseados no guia: Conselho de Engenheiros Profissionais do Texas, Associação de Engenheiros Profissionais e geocientistas da Columbia Britânica (APEGBC).
- A ACM e IEEE desenvolveram conjuntamente e aprovaram um Código de Ética e Exercício Profissional da Engenharia de Software.

11.4. Sugestão de leitura

Para saber um pouco mais sobre os estudos conduzidos pela IEEE que colabora com o incremento da prosperidade mundial, promovendo a engenharia de criação, desenvolvimento, integração, compartilhamento e o conhecimento aplicado no que se refere à ciência e tecnologias da eletricidade e da informação, em benefício da humanidade e da profissão, consulte: <http://www.computer.org/portal/web/guest/home>.

Com interesse em consultar o guia completo do SWEBOK, suas certificações e suas áreas de conhecimento, consulte: <http://www.swebok.org/index.html>, onde poderá baixar a versão digital do guia 2004.

O livro de Koscianski e Soares (Qualidade de Software, Novatec, 2007) oferece uma cobertura detalhada das métricas da qualidade, metodologias e técnicas mais modernas para o desenvolvimento de software. O livro aborda a definição de qualidade de software, normas e organismos normativos, métricas e conteúdo que se refere principalmente aos processos de desenvolvimento de software.

Uma excelente fonte de informação sobre testes de software pode ser consultada no livro de Rios e Moreira (Teste de Software, segunda edição, Alta Books, 2006), para aqueles que estão se iniciando no assunto de teste de software.

11.5. Exercícios

1. O que é o *Guide to the SWEBOK* e por qual entidade é produzido?
2. Quais são os cinco principais objetivos do SWEBOK?
3. Quantas e quais são as áreas de conhecimento em que o SWEBOK é dividido?

Código de campo alterado

4. Como o objetivo “Acesso por Tópicos às Referências” é tratado em cada Área de Conhecimento?
5. O que é um Requisito de Software?
6. De que trata a área de conhecimento Requisitos de Software? Qual a sua relação com o problema que deve ser resolvido pelo software?
7. Qual a importância de uma boa Especificação de Requisitos para a qualidade do software?
8. De que trata a área de conhecimento Construção de Software?
9. Todos os tipos de teste de software estão exclusivamente tratados nesta área de conhecimento?
10. Descreva as subáreas de conhecimento Manutenção de Software?
11. Todas as manutenções de software referem-se aos erros ocorridos no software? Justifique sua resposta.
12. Como são classificadas as manutenções, de acordo com o SWEBOK? Em sua opinião, quais as categorias de manutenção devem ocorrer com maior frequência, idealmente?
13. Descreva o ciclo de atividades dentro do processo de manutenção, segundo o SWEBOK.
14. De que trata a área de conhecimento Gerenciamento de Configuração de Software?
15. Quais as atividades do processo de Gerenciamento de Configurações, segundo o SWEBOK?
16. De que trata a área de conhecimento Gerenciamento do Processo de Software?
17. De que trata a área de conhecimento Processo de Software?
18. De que trata a área de conhecimento Ferramentas e Métodos?

11.6 Referências

- Feiler, P.H; Humphrey, W.S. Software Process Development and Enactment: Concepts and Definitions. In: II Internacional Conference on the Software Process, 1993. Berlin.
- J. Goguen and C. Linde, “Techniques for Requirements Elicitation,” presented at International Symposium on Requirements Engineering, 1993.
- Koscianski, A. and Soares, M.S. (2006). “Qualidade de Software” - 2ª edição, Novatec. São Paulo – SP
- Naur, P. and Randell, B. E. (1968). “Software Engineering: Report on a Conference Sponsored by the NATO Science Committee. Technical report”, NATO, Garmisch, Germany.
- Pressman, R. S. Engenharia de Software. 5. ed. Rio de Janeiro: McGraw-Hill, 2002.
- Pressman, R. S. Engenharia de Software. São Paulo: Makron Books, 1995.
- Reis, R.Q; Reis, C.A; Nunes, D. J. Automação no Gerenciamento do processo de Engenharia de Software. Departamento de Informática, Universidade Federal do Pará (UFPA), Instituto de Informática, Universidade Federal do Rio Grande do Sul (UFRGS), Belém (PA), outubro de 2002.
- Ribeiro, D.A. Escolha de uma das áreas de Engenharia de Software do SWEBOK Centro de Informática – Universidade Federal de Pernambuco (UFPE), Recife – PE – Brasil.
- Rios, E; Moreira, T. Teste de Software, Segunda edição, Alta Books, 2006.
- SWEBOK. Guide to the Software Engineering Body of Knowledge. 2004 Version. project of the IEEE Computer Society Professional Practices Committee. Disponível em: <<http://www.swebok.org/>>. Acesso em: 22 Ago. 2009.
- Tavares, A. L. O. and Eckel, A. P. and Scarpa, C. and Vedrame R. “ Engenharia de Software: Uma Visão Geral”. Curso de especialização em Engenharia de Software de Projetos de Software - Universidade do Sul de Santa Catarina (UNISUL) – Palhoça, SC – Brasil
- Weinberg, G.M. *Software com Qualidade*. Makroon Books, 1994.