Capítulo

11

Uma introdução ao SWEBOK

André Luís de L. Torres

*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 Engineers).*

**11.1. Introdução**

Em uma conferência da OTAN no ano de 1968, 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 feitos e mal interpretados, conflitos nos objetivos e mudanças intermináveis e mal controladas.

Tais necessidades aumentaram a importância e responsabilidades dos especialistas ligados a uma das áreas da computação, conhecida como Engenharia de Software. Com isso, a *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 iremos apresentar uma descrição sobre a visão da Engenharia de Software detalhada e apoiada por um processo de desenvolvimento realizado por profissionais, sociedade científica e órgãos públicos que culminou no guia tema do capítulo.

**11.2. O Projeto do SWEBOK**

O SWEBOK surgiu através de uma parceria entre a 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 de Engenharia de Software, professores e instrutores. Os objetivos do SWEBOK 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;
* 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

|  |  |
| --- | --- |
| EspecializadoPráticas usadas apenas por alguns tipos de software | Geralmente AceitasPráticas tradicionais estabelecidas recomendadas pela maior parte das organizações |
| Pesquisa AvançadaPráticas inovadoras usadas apenas por algumas organizações com conceitos a serem desenvolvidos e testados em organizações de pesquisa. |

**Tabela 11.1. Categorias do Conhecimento conforme o SWEBOK**

O guia, como é convencionalmente chamado, divide a Engenharia de Software em onze áreas de Conhecimento - *KnowLedge Areas* (KAs): requisitos, gerência de engenharia, projeto, métodos e ferramentas de engenharia, construção, processo de engenharia, testes, qualidade, manutenção, disciplinas relacionadas à gerência de configuração, 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 CSDA (Certificação de Associação no Desenvolvimento de Software);**
* **Certificação CSDP (Certificação de Desenvolvimento Profissional de Software).**

A certificação CSDA oferece os princípios fundamentais para o avanço do profissional de Engenharia de Software, disponibilizando uma forte alavanca para experiência estudantil e as reais requisições do mercado de trabalho. CSDA é o primeiro passo para se tornar um *Certified Software Development Professional* (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 utiliza uma organização hierárquica, decompondo todas as KA´s em um conjunto de temas com rótulos reconhecíveis pela área de interesse do leitor sobre a Engenharia de Software. A figura 11.1 apresenta o corpo de conhecimento do guia, como também seus níveis hierárquicos.

****

**Fonte: Adaptado do SWEBOK,2004.**

Figura 11.1. Organização do SWEBO em tópicos da Qualidade de Software.

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

11.2.2.1. Requisitos de Software

 Os requisitos expressam a necessidade 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).

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)



Figura 11.2. O que é um requisito?

(Pressman,2002) cita que “se você não analisa, é altamente provável que construa uma solução de software muito elegante que resolve o problema errado”. Esta atitude pode resultar em perda de tempo e dinheiro, pessoas frustradas e clientes insatisfeitos, conforme ilustrado na figura 11.2.

A área de requisitos de software está dividida em sete sub-áreas:

1. ***Fundamentos dos Requisitos***

O SWEBOK define requisitos como uma propriedade que deve ser observada a fim de resolver algum problema do mundo real. Logo, o problema pode ser para automatizar parte de uma tarefa a ser realizada por alguém na utilização de um software.

Nessa atividade busca-se descrever os requisitos funcionais, não funcionais, do produto, do software, quantificáveis e os requisitos de sistema e de software.

1. ***Processo de Requisitos***

O processo de requisitos de software orienta o planejamento de requisitos para se encaixar com o processo completo de planejamento de software.

1. ***Declaração de Requisitos***

A declaração de requisitos tem a preocupação com a coleta dos requisitos de software pelo engenheiro de software. Identifica as fontes dos requisitos e define as técnicas para extrair os requisitos. Primeiro estágio para o entendimento do problema disposto.

1. ***Análise de*** ***Requisitos***

Alguns cuidados devem ser tomados para descrever com precisão os requisitos. A preocupação é a de detectar e resolver conflitos entre requisitos, além de aprimorar os requisitos do sistema para requisitos de software.

1. ***Especificação de Requisitos***

Para muitos profissionais da engenharia, conforme o guia, especificação refere-se à atribuição de valores numéricos ou limites para os objetivos do projeto. A principal atividade desta fase é a confecção da documentação do sistema, especificando os componentes de software.

1. ***Validação de Requisitos***

A documentação de requisitos pode ser objeto de validação e procedimento de verificação. Tal atividade busca a conformidade do documento com os padrões da organização. Possui as etapas de revisão, prototipação e testes de aceitação.

1. ***Considerações Práticas***

Valida 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 da tarefa.

 As principais atividades e responsabilidades das sub-áreas de requisitos de software 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.

**11.2.2.2.** **Projeto de software**

Projeto de software ou Design de Software é a atividade do ciclo de vida da Engenharia de Software (ES) em que os requisitos do software 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 é durante suas atividades que podem ser detectados problemas no funcionamento do software que podem comprometer seu uso e até mesmo a conclusão do mesmo.

A área de projetos de software está dividida em sete sub-áreas:

1. ***Fundamentos do Design de Software***

São os conceitos, noções e terminologias introduzidas na forma de uma base fundamental para a compreensão do papel do design de software.

1. ***Questões Chave no Design de Software***

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

1. ***Estrutura e Arquitetura de Software***

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

1. ***Evolução e Análise de Qualidade do Design de Software***

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

1. ***Notações do Desenho de Software***

Informa notações estruturais (estática) e comportamentais (dinâmica).

1. ***Estratégias e Métodos para o Desenho de Software***

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

11.2.2.3. Construção de software

Refere-se à criação do conjunto de programas (componentes) que compõe o software através de uma combinação de codificação, verificação, testes unitários, testes de integração e depuração.

A construção de software está ligada a todas as outras KA´s, mais fortemente ao Design de Software e Teste de Software. 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:

1. ***Fundamentos da Construção de Software***

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

1. ***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.

1. ***Considerações Práticas da Construção de Software***

Nessa sub-área 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.

**11.2.2.4.** **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).

Nessa seção é feito apenas um breve comentário sobre a área de conhecimento de teste de software conforme o SWEBOK, devido ao seu embasamento teórico está detalhado no capitulo 10 desse 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).

 Essa área descrita no 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 sub-áreas dessa área de conhecimento: fundamentos, níveis de Teste de Software, técnicas, medidas, processo e considerações práticas.

É atualmente considerado 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 busca continua da qualidade de software.

### 11.2.2.5. Manutenção de software

O desenvolvimento de software requer esforços que visam a entrega de um produto que satisfaça os 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, modificações são feitas com o objetivo de corrigir falhas, melhorar o desempenho ou adaptá-las a um ambiente externo (SWEBOK, 2004).

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

O fluxo de eventos que podem ocorrer como resultado de um pedido de manutenção é ilustrado na figura 11.3. Onde, segundo Pressman 2002, 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 está num estado ruim.

****

**Figura 11.3. Manutenção estruturada *versus* não estruturada**

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

1. ***Fundamentos da Manutenção de Software***

Nesta seção 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.

1. ***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 organizações de manutenção e outsourcing), estimativas de custos e medidas (parâmetros e experiência).

1. ***Processo de Manutenção de Software***

Fornece 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 similar a IEEE 1219 (atividades de manutenção de software).

******

**Figura 11.4. Atividades do processo de manutenção** *versus* **Processo de Manutenção de software**

1. ***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.

**11.2.2.6.** **Gerência de configuração de Software**

A Gerência de Configuração de Software – *Software Configuration Management* (SCM) é a disciplina que identifica 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 todo o ciclo de vida do sistema (SWEBOK, 2004).

São atividades da SCM:

1. ***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.

1. ***Identificação da Configuração***

Esta atividade 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. Essa atividade constitui a base para as outras atividades do SCM. Suas principais atividades são: identificação dos itens a ser controlado (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).

1. ***Controle da Configuração***

Abrange a gestão durante o ciclo de vida do software, como por exemplo: processo para determinar quais as mudanças a serem feitas, a autoridade para aprovar algumas alterações, apoio a implementação dessas mudanças, entre outras atividades. Uma Ordem de mudança de 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.

1. ***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?

1. ***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?
1. ***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 freqüentemente necessário para recriar versões específicas de pacotes e os materiais corretos para entrega da versão.

A figura 11. 5 ilustra as atividades citadas como também o seu relacionamento.



**Figura 11.5. Atividades da SCM**

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 e 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 inicio, terminando quando o software é tirado de operação.

**11.2.2.7.** **Gerência de Engenharia de Software**

A área de “Gerenciamento 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 é atividades relacionadas com o gerenciamento de pessoas, levando em consideração equipe, clientes e responsáveis da própria organização. Tendo como responsabilidades o treinamento, motivação pessoal. 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 7 sub-áreas:

1. ***Iniciação e Definição de Escop******o***

O foco é determinar os requisitos através de vários métodos de levantamento, avaliação do projeto e especificação dos requisitos (comentado na seção 11.2.2.1 deste capítulo) 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.

1. ***Planejamento do Projeto***

Neste ponto, o ciclo de vida dos processos de software é avaliado para o melhor planejamento, levanto em conta fatores como: natureza do projeto, complexidade funcional e técnica, requisitos de qualidade, entre outros. Tal sub-área é um fator importante para o sucesso da qualidade de software, detalhada na seção 11.2.2.11. Os tópicos relacionados são: planejamento do processo, esforço, previsão de custo, alocação de recursos, gerenciamento de risco e da qualidade.

1. ***Declaração do Plano de Projeto***

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

1. ***Revisão e avaliação***

Essa 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.

1. ***Fechamento***

O projeto atinge seu 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.

1. ***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.

### 11.2.2.8. Processo de Engenharia de Software

(Reis e Nunes, 2002) apud (Feiler,1993) comenta que 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.

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 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 primeiro nível é coberta por outro o KA´s no Guia. O termo "processo de Engenharia de Software" pode ser interpretada de diferentes maneiras, e isso pode causar confusão

O processo de Engenharia de Software envolve vários outros processos, como o de desenvolvimento o de gerenciamento, 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, essa área importante não apenas para empresas grandes, mas também devem ser levadas em consideração para pequenas empresas, facilitando e auxiliando no crescimento dessas. 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 sub-áreas:

1. ***Mudança e Implementação do Processo***

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

1. ***Definição do Processo***

A definição do processo, conforme o guia, pode ser definido 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 útil.

1. ***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. Tais processos são detalhados no capitulo X.

1. ***Medidas de Produtos e Processos***

Embora a aplicação de medidas de Engenharia de Software pode 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 sub-área a norma ISO/IEC 15939 para descrever tais medidas e métodos para produtos e processos.

Uma das grandes dificuldades enfrentadas por está 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 e o desempenho das equipes e qualidade final dos produtos são beneficiados.

### 11.2.2.9. Métodos e ferramentas de engenharia

Ferramentas de desenvolvimento de software são ferramentas criadas para auxiliar no ciclo de vida do software. Essas ferramentas normalmente automatizam algumas atividades do processo de desenvolvimento, fazendo com que o analista concentre-se nas atividades que exigem maior trabalho intelectual (SWEBOK, 2004).

Métodos de Engenharia de Software impõe 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 de 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).

Está KA cobre os processos do ciclo de vida do projeto de software e portanto relacionados a cada KA do guia, conforme ilustrado na figura 11.6.



**Figura 11.6. Distribuição dos tópicos relacionados aos Métodos e ferramentas de engenharia**

1. ***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.

1. ***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.

### 11.2.2.10. Qualidade de software

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

A ISO 9001 define a qualidade como "o grau em que um conjunto de
características inerentes satisfaz as necessidades” (SWEBOK,2004).

Em relação à qualidade, 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, que trata da qualidade de produtos de software, abrange esses dois tópicos (Koscianski, 2006).

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 é detalhado no capítulo X.
* Norma ISO/IEC 9126, proposta pela ISO (*International Organization for* *Standardization*), tal modelo é detalhado no capítulo X.

As organizações desenvolvedoras desses modelos de qualidade fornecem selos de qualidade para as empresas que se submetem à avaliações e estiverem dentro dos padrões propostos. Esses selos são muito valorizados pelas empresas que compram software, 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 processo de 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 SWEBOK, é dividida em três tópicos que serão rapidamente descritos a seguir:

1. ***Fundamentos da Qualidade de Software***

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

Os aspectos éticos do trabalho com software têm se tornado mais evidente com nossa dependência da tecnologia; toda uma nova classe de problemas surgiu com os crimes de computador. Respostas sociais, éticas e de legislação estão sendo desenvolvidas para procurar tratar adequadamente cada caso (Koscianski, 2006).

Este tópico tem com subáreas: Engenharia de Software Cultura e Ética, valor e custo da qualidade, modelos e características da qualidade e a melhoria da qualidade.

1. ***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 inicio do projeto sejam cumpridos, se aplicando a todas as pespectivas do processo de software. Definindo processos, responsáveis, produtos e recursos.

Alguns dos processos desta subárea são definidos pelo padrão IEEE 12207, sendo eles: garantia de qualidade, verificação, validação, revisão e auditoria.

1. ***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 SQuaRE, onde os valores desejados para as medidas sejam estabelecidos no início do projeto, ao se definir os requisitos.

**11.2.2.11.** **Disciplinas relacionadas**

Para delimitar a Engenharia de Software 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 11.7.



**Figura 11.7. Disciplinas relacionadas com a Engenharia de Software**

1. ***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 computados, entre outras.

1. ***Ciências da Computação***

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

1. ***Gerenciamento***

Segue recomendações para MBA definidos pelo Conselho Europeu, que inclui como áreas de conhecimento relacionadas: contabilidade, finanças, marketing, direito, gestão de recursos humanos, entre outras.

1. ***Matemática***

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

1. ***Gestão de Projetos***

Segue as recomendações conforme o guia PMBOK Guide 2006, sendo o mesmo detalhado no capitulo X deste livro. São conhecimentos relacionados pelo guia: gestão de custos, de riscos, de qualidade, entre outros.

1. ***Gestão de Qualidade***

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

1. ***Ergonomia***

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

1. ***Engenharia de Sistemas***

O Conselho Internacional de Engenharia de Sistemas (INCOSE) afirma que “ Engenharia de Sistemas é uma abordagem interdisciplinar que permitem 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 abrangida 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);
* Redistribuição de matérias entre as áreas de conhecimento;

A atualização completa do SWEBOK Guide tem previsão para o primeiro semester de 2010.

**11.3. Tópicos de pesquisa**

Aplicação do SWEBOK

**11.4. Sugestão de leitura**

Para saber um pouco mais sobre os estudos conduzidos pela IEEE que colabora no 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, 2006) 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, comentado na seção 11.2.2.4, 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?
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 Design de Software? Como distingui-la do que comumente chamamos de Projeto de Software?
9. De que trata a área de conhecimento Construção de Software?
10. De que trata a área de conhecimento Teste de Software? Todos os tipos de teste de software estão exclusivamente tratados neta área de conhecimento?
11. De que trata a área de conhecimento Manutenção de Software?
12. Todas as manutenções de software referem-se aos erros ocorridos no software? Justifique sua resposta.
13. Como são classificadas as manutenções, de acordo com o SWEBOK?Em sua opinião, quais as categorias de manutenção devem ocorrer com maior freqüência, idealmente?
14. Descreva o ciclo de atividades dentro do processo de manutenção, segundo o SWEBOK.
15. De que trata a área de conhecimento Gerenciamento de Configuração de Software?
16. Quais as atividades do processo de Gerenciamento de Configurações, segundo o SWEBOK?
17. De que trata a área de conhecimento Gerenciamento do Processo de Software?
18. De que trata a área de conhecimento Processo de Software?
19. De que trata a área de conhecimento Ferramentas e Métodos?

Referências

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.** São Paulo: Makron Books, 1995.

Pressman, R. S. **Engenharia de Software.** 5. ed. Rio de Janeiro: McGraw-Hill, 2002.

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.

Rios, E; Moreira, T. Teste de Software, Segunda edição, Alta Books, 2006.

Feiler, P.H; Humphrey, W.S. Software Process Development and Enactment: Concepts and Definitions. In: II Internacional Conference on the Software Process, 1993. Berlin.

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