Engenharia de Requisitos - SWEBOK

Profª. Ana Flávia

ROTEIRO

- O que é SWEBOK;
- Objetivos do SWEBOK;
- Áreas de Conhecimento do SWEBOK

O QUE É SWEBOK?

- É um guia para o Corpo de Conhecimento de Engenharia de Software, ou seja, fornece a definição do corpo de conhecimento aceito e práticas recomendadas;
- É um guia que descreve a soma de conhecimentos inerente, referentes à Engenharia de Software, ou seja, abrange práticas já aprovadas e de uso comum.

OBJETIVOS DO SWEBOK

- Promover uma visão consciente da Engenharia do Software no mundo inteiro;
- Esclarecer o lugar "e estabelecer o limite" da Engenharia de Software com respeito a outras disciplinas como Ciência da Computação, Gerência de Projetos, Engenharia de Computação e Matemática;

OBJETIVOS DO SWEBOK

- Caracterizar os conteúdos da disciplina de Engenharia de Software;
- Fornecer um acesso ao Corpo de Conhecimento de Engenharia de Software;
- Fornecer uma base para desenvolvimento de currículo, certificação individual e licenciamento de material.

ÁREAS DE CONHECIMENTO

1. Requisitos de Software;
2. Design de Software;
3. Construção de Software;
4. Teste de Software;
5. Manutenção de Software;

ÁREAS DE CONHECIMENTO

6. Gerência de Configuração de Software;
7. Gerência da Engenharia de Software;
8. Processo de Engenharia de Software;
9. Ferramentas e Métodos da Engenharia de Software;
10. Qualidade de Software.

REQUISITOS DE SOFTWARE

- Um requisito é definido como uma propriedade que deve ser exposta para resolver algum problema do mundo real;
- Essa área é subdividida em mais 7 subáreas:

1. Fundamentos de Requisitos de Software;
2. Processo de Requisitos;
3. Elicitação de Requisitos;
4. Análise de Requisitos;
5. Especificação de Requisitos;
6. Validação de Requisitos;
7. Considerações Práticas.

REQUISITOS DE SOFTWARE

- Fundamentos de Requisitos de Software:
  - Definição de requisito de software – propriedade que deve ser apresentada pelo software para resolver um problema do mundo real;
  - **Requisitos de produtos** e **processo** – o primeiro **são as regras de negócio do software** e segundo **são as restrições de desenvolvimento impostas para a criação do software**;
  - Requisitos funcionais e não funcionais – o primeiro descreve as funções que o sistema deve executar e o segundo são restrições ou requisitos de qualidade;

REQUISITOS DE SOFTWARE

- Fundamentos de Requisitos de Software:
  - Propriedades emergentes – são propriedades que não são endereçadas por um componente, mas cuja satisfação depende da inter operação de diversos componentes do sistema;
  - Requisitos quantificáveis – requisitos precisam ser definidos com clareza e sem ambiguidade e quando for o caso quantitativamente;
  - Requisitos de sistema e requisitos de software – o primeiro corresponde as necessidades para um sistema como um todo e o segundo é derivado do primeiro;

REQUISITOS DE SOFTWARE

- Processo de Requisitos:
  - Modelos de processo – identifica os requisitos de software como itens de configuração e os gerencia usando as práticas de GC;
  - Atores de processo – define os papéis dos participantes do processo;

REQUISITOS DE SOFTWARE

- Elicitação de Requisitos:
  - Fontes de requisitos – objetivo do negócio, conhecimento do domínio da aplicação, pontos de vista dos diferente tipos de *stakeholders*;
  - Técnicas de elicitação – entrevistas, cenários, protótipos, reuniões com facilitadores, observação entre outros;

REQUISITOS DE SOFTWARE

- Análise de Requisitos:
  - Classificação de requisitos – por exemplo se o requisito é funcional ou não funcional ou a prioridade do requisito;
  - Modelagem conceitual – desenvolvimento de modelos de um problema do mundo real para auxiliar no entendimento deste;
  - Projeto de arquitetura e alocação de requisitos – é o ponto onde o processo de requisitos sobrepõe-se com o projeto de software ou sistema;
  - Negociação de requisitos – resolver conflitos entre requisitos.

REQUISITOS DE SOFTWARE

- Especificação de Requisitos:
- Documento de definição do sistema – registra em alto nível os requisitos do sistema;
- Especificação de requisitos do sistema – aplica-se a sistemas que possuem número considerável de componentes;
- Especificação de requisitos de software – define o que o produto deve ser e o que ele não deve ser, estabelecendo base para acordo entre os *stakeholders*;

REQUISITOS DE SOFTWARE

- Validação de Requisitos:
- Revisões de requisitos – busca por erros, contradições, falta de clareza e desvios de práticas padrões;
- Prototipagem – meio pelo qual é possível validar a interpretação que o engenheiro faz do requisito;
- Validação do modelo – é necessário validar a qualidade dos modelos desenvolvidos durante a análise;
- Testes de aceitação – o produto deve ser validado nos seus requisitos.

REQUISITOS DE SOFTWARE

- Considerações Práticas:
  - Natureza interativa do processo de requisito;
  - Gerência de mudança;
  - Atributos de requisitos;
  - Rastreamento de requisitos;
  - Medição de requisitos.

DESIGN DE SOFTWARE

- Design é o processo de definir a arquitetura, componentes, interfaces, e outras características de um sistema ou componente;
- Essa área é subdividida em mais 6 subáreas:
  1. Fundamentos de Design de Software;
  2. Questões-Chaves em Design de Software;
  3. Estrutura e Arquitetura de Software;
  4. Análise de Qualidade do Design e Avaliação do Software;
  5. Notações de Design de Software;
  6. Estratégias de Design de Software e Métodos

DESIGN DE SOFTWARE

- Fundamentos de Design de Software:
  - **Conceitos gerais de design** – é uma forma de se resolver problemas, pois oferece objetivos, restrições, alternativas, representações e soluções;
  - **Contexto do design de software** – ligado fortemente ao contexto em que se enquadra o ciclo de vida de software
  - **Processo de design de software** – dividido em design arquitetural e design detalhado. O primeiro descreve como o software é composto e organizado dentro de componentes e o segundo o comportamento desses componentes;
  - **Técnicas ativas** – são os princípios do design de software, ou seja, noções chaves consideradas fundamentais no desenvolvimento do software.

DESIGN DE SOFTWARE

- Questões-Chaves em Design de Software:
  - **Concorrência** – como decompor o software em processos, tarefas e serviços e desejar eficiência, sincronização entre outros;
  - **Controle e transporte de eventos** – como organizar dados e controlar o fluxos de dados;
  - **Distribuição de componentes** – como distribuir o software baseado em hardware;
  - **Transporte de exceção, erro e tolerância a falha** – como prevenir e tolerar falhas;
  - **Interação e apresentação** – como estruturar e organizar interações com usuários e apresentação de informações;
  - **Persistência dos dados** – como dados de longa vida podem ser transportados.

DESIGN DE SOFTWARE

- Estrutura e Arquitetura de Software:
  - **Estruturas arquitetônicas e viewpoints** – diferentes facetas de alto nível de design de software podem e devem ser descritos e documentados;
  - **Padrões de design** – corresponde a uma solução comum para um problema comum em determinado contexto;
  - **Famílias de programas e frameworks** – permite a reutilização de design de softwares e de componentes;

DESIGN DE SOFTWARE

- Análise de Qualidade do Design e Avaliação do Software:
  - **Atributos de qualidade** – proporcionam a obtenção de manutenibilidade, portabilidade, testabilidade, rastreabilidade, correção entre outros;

DESIGN DE SOFTWARE

- Análise de Qualidade do Design e Avaliação do Software:
  - **Análise de qualidade e técnicas de avaliação** – é a revisão do design de software de forma formal ou informal permitindo a avaliação do projeto;
  - **Medidas** – utilizadas para avaliar quantitativamente o tamanho do design de software, a estrutura ou a qualidade;

DESIGN DE SOFTWARE

- Notações de Design de Software:
  - **Descrições estruturais** – fornece uma visão estática, isto é, descreve os componentes principais e como eles são interligados;
  - **Descrições comportamentais** – fornece uma visão dinâmica, isto é, descreve o comportamento dinâmico entre componentes de software;

DESIGN DE SOFTWARE

- Estratégias de Design de Software e Métodos:
  - Estratégias gerais;
  - Design orientado por função;
  - Design orientado a objeto;
  - Design centrado por estruturas de dados;
  - Design baseado em componente;

CONSTRUÇÃO DE SOFTWARE

- Se refere à criação detalhada de trabalho, ou seja, criação do software através da combinação de codificação, verificação, testes de unidades, testes de integração e depuração.
- Essa área é subdividida em mais 3 subáreas:
  1. Fundamentos de Construção de Software;
  2. Gerenciamento da Construção;
  3. Considerações Práticas.

CONSTRUÇÃO DE SOFTWARE

- Fundamentos de Construção de Software:
  - **Minimizar complexidade** – permitir que as pessoas realizem tarefas complexas diminuindo suas limitações;
  - **Antecipação de mudança** – a maior parte dos softwares irá mudar com o tempo;
  - **Construção para verificação** – meio de construir software de forma que as falhas podem ser prontamente encontradas pelo engenheiro de software;
  - **Normas de construção** – envolvem o método de comunicação, linguagem de programação, plataforma, ferramentas, bem como normas internas da corporação;

CONSTRUÇÃO DE SOFTWARE

- Gerenciamento da Construção:
  - **Modelos de construção –** escolha de qual modelo adotar;
  - **Planejamento de construção –** define a ordem em que os componentes são criados e entregues de acordo com o modelo escolhido;
  - **Medição de construção**;

CONSTRUÇÃO DE SOFTWARE

- Considerações Práticas:
  - **Design de construção** – alguns detalhes do design ocorrerão em nível de construção;
  - **Linguagem de construção** – incluem todas as formas de comunicação em que um homem pode especificar uma solução de um problema executável para o computador;
  - **Codificando** – técnicas para criar o código fonte de forma compreensível, incluindo nomeações e layout;
  - **Testando a construção** – são divididos em teste de unidade e teste de integração;
  - **Reutilização**;
  - **Qualidade da construção**;
  - **Integração**.

TESTE DE SOFTWARE

- Compõem-se da verificação dinâmica de uma seleção de domínios de execuções normalmente infinito, contra o comportamento esperado.

- Essa área é subdividida em mais 5 subáreas:
  1. Fundamentos de Teste de Software;
  2. Níveis de Teste;
  3. Técnicas de Teste;
  4. Mensurações Relacionadas aos Testes;
  5. Processo de Testes.

TESTE DE SOFTWARE

- Fundamentos de Teste de Software:
  - Definir critérios de seleção de testes;
  - Objetivos do teste;
  - Teste para identificação de defeito;
  - Problema do Oráculo;
  - Limitações teóricas e práticas de teste;
  - Problema dos caminhos inviáveis;
  - Testabilidade;
  - Relação do teste com outras atividades.

TESTE DE SOFTWARE

- Níveis de Teste:
  - Alvo do teste (unidade, integração ou sistema);
  - Objetivos de teste (aceitação e qualificação, instalação, alfa e beta entre outros);

TESTE DE SOFTWARE

- Técnicas de Teste:
  - Baseada na intuição e na experiência do engenheiro de software;
  - Baseadas em especificação;
  - Baseadas em código;
  - Baseadas em falha;
  - Baseada em uso;
  - Baseadas na natureza da aplicação;
  - Seleção e combinação de técnicas.

TESTE DE SOFTWARE

- Mensurações Relacionadas aos Testes:
  - Avaliação do programa sob o teste;
  - Avaliação dos teste executados;
- Processo de Testes:
  - Considerações práticas;
  - Atividades de teste;

MANUTENÇÃO DE SOFTWARE

- Uma vez em operação, as anomalias são descobertas, modificações no ambiente operacional e novos requisitos dos usuários são expostos.

- Essa área é subdividida em mais 4 subáreas:
  1. Fundamentos de Manutenção de Software;
  2. Questões-Chave em Manutenção de Software;
  3. Processo de Manutenção;
  4. Técnicas para Manutenção.

MANUTENÇÃO DE SOFTWARE

- Fundamentos de Manutenção de Software:
  - Definição e terminologia;
  - Natureza da manutenção;
  - Necessidade da manutenção;
  - Custos da manutenção;
  - Evolução do Software;
  - Categorias de Manutenção.

MANUTENÇÃO DE SOFTWARE

- Questões-Chave em Manutenção de Software:
  - Problemas técnicos;
  - Gestão de problemas;
  - Estimativa de custo de manutenção;
  - Medição da manutenção do software;

MANUTENÇÃO DE SOFTWARE

- Processo de Manutenção:
  - Processo de manutenção;
  - Atividades de manutenção;

- Técnicas para Manutenção:
  - Compreensão;
  - Reengenharia;
  - Engenharia reversa;

GERÊNCIA DE CONFIGURAÇÃO DE SOFTWARE

- Responsável por identificar a configuração do software em pontos distintos no tempo com o propósito de sistematicamente controlar modificações à configuração e de manter a integridade e a autoridade da configuração em todas as partes do ciclo de vida de sistema.

- Essa área é subdividida em mais 6 subáreas:
  1. Gerência do Processo de GCS;
  2. Identificação de Configuração de Software;
  3. Controle de Configuração de Software;
  4. Registros de Estado de Configuração de Software;
  5. Auditoria de Configuração de Software;
  6. Gerência de Liberação e Entrega de Software.

GERÊNCIA DE CONFIGURAÇÃO DE SOFTWARE

- Gerência do Processo de GCS:
  - Contexto organizacional para o GCS;
  - Limitações e orientações para o processo de GCS;
  - Planejamento para GCS;
  - Plano GCS;
  - Monitoramento do GCS;

GERÊNCIA DE CONFIGURAÇÃO DE SOFTWARE

- Identificação de Configuração de Software:
  - Identificação dos itens a serem controlados;
  - Biblioteca de software;
- Controle de Configuração de Software:
  - Requerente, avaliação e aprovação de alterações de software;
  - Implementação de mudanças de software;
  - Desvios e dispensas;

GERÊNCIA DE CONFIGURAÇÃO DE SOFTWARE

- Registros de Estado de Configuração de Software:
  - Status de configuração de software;
  - Software para relatar o status de configuração;

- Auditoria de Configuração de Software:
  - Software de auditoria de configuração funcional;
  - Software de auditoria de configuração física;
  - Auditoria de processo de uma *baseline*;

GERÊNCIA DE CONFIGURAÇÃO DE SOFTWARE

- Gerência de Liberação e Entrega de Software:
  - Construindo software;
  - Software de gerenciamento de liberação;

GERÊNCIA DE ENGENHARIA DE SOFTWARE

- GERÊNCIA DA ENGENHARIA DE SOFTWARE
- Aponta o gerenciamento e mensuração da engenharia de software;
- Essa área é subdividida em mais 6 subáreas:
  1. Iniciação e Definição de Escopo;
  2. Planejamento de Projeto de Software;
  3. Aprovação do Projeto de Software;
  4. Revisão e Avaliação;
  5. Fechamento;
  6. Mensuração da Engenharia de Software.

INICIAÇÃO E DEFINIÇÃO DE ESCOPO

- Determinação e Negociação de Requisitos;
- Análise de Viabilidade;
- Processos para Análise e Revisão de Requisitos.

PLANEJAMENTO DE PROJETO DE SOFTWARE

- Planejamento do Processo;
- Determinação de Entregas;
- Estimativa de Esforço, Cronograma e Custo;
- Alocação de Recursos;
- Gerenciamento de Riscos e Qualidade;
- Gerenciamento do Planejamento.

APROVAÇÃO DO PROJETO DE SOFTWARE

- Implementação dos Planos;
- Gerenciamento de Contratação de Fornecedores;
- Implementação do Processo de Medição;
- Processo de Monitoramento;
- Controle do Processo;
- Relatórios.

REVISÃO E AVALIAÇÃO

- Determinação da Satisfação dos Requisitos;
- Análise e Avaliação do Desempenho.

FECHAMENTO

- Determinando o Término;
- Atividades de Fechamento.

PROCESSO DE ENGENHARIA DE SOFTWARE

- Trata da definição, implementação, avaliação, mensuração, gerenciamento, alterações e melhoria do próprio processo de engenharia de software

- Essa área é subdividida em mais 4 subáreas:

  1. Processo de Implementação e Mudanças;
  2. Definição do Processo;
  3. Avaliação de Processo;
  4. Mensuração de Produto e Processo.

PROCESSO DE IMPLEMENTAÇÃO E MUDANÇAS

- Processo de Infra-Estrutura;
- Ciclo de Gerenciamento do Processo de Software;
- Modelos de Processo de Implementação e Mudanças;
- Considerações Práticas.

DEFINIÇÃO DO PROCESSO

- Modelos de Ciclo de Vida de Software;
- Processos de Ciclo de Vida de Software;
- Notações para Definição de Processos;
- Adaptação de Processo;
- Automatização.

AVALIAÇÃO DE PROCESSO

- Modelos de Avaliação de Processos;
- Métodos para Avaliação de Processos.

FERRAMENTAS E MÉTODOS DA ENGENHARIA DE SOFTWARE

- Inclui ferramentas e métodos para serem aplicados na engenharia de software

- Essa área é subdividida em mais 2 subáreas:
  1. Ferramentas para Engenharia de Software;
  2. Métodos para Engenharia de Software.

FERRAMENTAS PARA ENGENHARIA DE SOFTWARE

- **Ferramentas de Requisitos** (modelagem e rastreabilidade);
- **Ferramentas de Design**;
- **Ferramentas de Construção** (Editores de Código, Compiladores, Interpretadores e Depuradores);
- **Ferramentas de Teste** (Geração, Execução, Avaliação, Gerenciamento, Análise e Desempenho);
- **Ferramentas de Manutenção** (Compreensão, Reengenharia e Engenharia Reversa);
- **Ferramentas de GCS** (Gerenciamento de Defeitos e Erros, Monitoramento, Versionamento e Correção);
- **Ferramentas de GES** (Planejamento e Acompanhamento, Riscos e Medição).

MÉTODOS PARA ENGENHARIA DE SOFTWARE

- **Métodos Heurísticos:** Estruturado, Orientado a Dados, Orientado a Objetos e Domínio Especifico;
- **Métodos Formais:** Especificação de linguagens e notações, Refinamento e Propriedades de Verificação/Confirmação;
- **Métodos de Prototipagem:** Estilo, Objetivo e Avaliação.

QUALIDADE DE SOFTWARE

- Aborda considerações relativas à qualidade de software que vão além dos processos de ciclo de vida do software

- Essa área é subdividida em mais 3 subáreas:
  - Fundamentos da Qualidade de Software;
  - Processos de Gerenciamento da Qualidade do Software;
  - Considerações Práticas.

FUNDAMENTOS DA QUALIDADE DE SOFTWARE

- Cultura e Ética da Engenharia de SW;
- Valor e Custo da Qualidade;
- Modelos e Características de Qualidade;
- Melhoria de Qualidade.

PROCESSOS DE GERENCIAMENTO DA QUALIDADE DO SOFTWARE

- Garantia da Qualidade;
- Verificação e Validação;
- Revisão e Auditoria.

CONSIDERAÇÕES PRÁTICAS

- Requisitos de Qualidade;
- Caracterização de Defeitos;
- Técnicas de Gerenciamento de Qualidade;
- Métrica de Qualidade.

Mais 5 áreas – v3.0

1. Práticas Profissionais em Engenharia de Software
2. Economia da Engenharia de Software
3. Fundamentos de Computação
4. Fundamentos de Matemática
5. Fundamentos de Engenharia