

# SWEBOK

---

## 1) O que é o SWEBOK? Para que ele foi escrito? Por quem ele foi escrito?

- O SWEBOK (Software Engineering Body of Knowledge) é um guia criado a partir de diversas parcerias da *IEEE Computer Society*.
- O mesmo foi escrito com o intuito de:
  - Promover uma visão consistente de engenheiros de software.
  - Diferenciar as peculiaridades da engenharia de software em relação a outros conhecimentos similares como a ciência da computação.
  - Especificar os conceitos da engenharia de software.
  - Providenciar conhecimento atual do que é a engenharia de software.
  - Providenciar um conhecimento base para o leitor.
- O SWEBOK foi escrito para engenheiros de software como público alvo.

Página: Introduction to the Guide

## 2) Como o Capítulo 2 "Design Software" define o Projeto de Interface do Usuário (User Interface Design).

- O SWEBOK define o projeto de interface do usuário como uma parte essencial do desenvolvimento do software. O design de interface tem como objetivo conectar o usuário à máquina da forma mais efetiva possível.

Página: 2-5/2-6

## 3) Quais são os Princípios Gerais Projeto de Interface do Usuário (General User Interface Design Principles)? Descreva cada um deles.

- Os principais princípios gerais projeto de interface do usuário são:
- **Fácil Aprendizado:** O software deve ser fácil de se aprender pelo usuário.
- **Familiaridade:** O software deve ter termos e conceitos de fácil entendimento pelo usuário.
- **Consistência:** O software deve possuir consistência em seu funcionamento.
- **Surpresas Mínimas:** O software não pode ter comportamentos repentinos que irão surpreender o usuário.
- **Recuperabilidade:** O software deve ser capaz de recuperar a interface do usuário após erros inesperados.
- **Ajuda/Feedback:** Após o acontecimento de erros, o software deve fornecer ao usuário uma página explicando o acontecimento do mesmo e como resolver o ocorrido.
- **Acessibilidade:** O software deve disponibilizar uma forma de interação para diversos usuários (Surdos, cegos, daltônicos, deficientes visuais e etc).

Página: 2-6

## 4) Quais são as duas questões-chaves que o projeto de interface deve resolver?

- As questões chaves são:
- Como o usuário deve interagir com o software.
- Como o software deve ser apresentado ao usuário.

Página: 2-6

## 5) Fale sobre o Processo de Design de Interface do Usuário? Quais são as três atividades principais desse processo?

- É um processo que desenvolve um protótipo e que determina as funções, a organização, o layout e entre outras características que o software terá quando concluído.
- O processo inclui:
- **Análise do usuário:** O desenvolvedor analisa os comportamentos do usuário em relação a softwares semelhantes.
- **Prototipagem de Software:** O desenvolvedor cria um protótipo para alguns usuários testarem.
- **Avaliação do Layout:** Os desenvolvedores observam a evolução da experiência do usuário em relação ao software.

Página: 2-7

## 6) O que é Teste de Software (Software testing)?

- Testes de Software são inúmeras verificações que são executadas diversas vezes para confirmar o funcionamento esperado do programa. O Teste de software é o que garante a qualidade do projeto, prevenindo com que o haja problemas graves no mesmo.

Página: 4-1/4-2

## 7) O que é teste de usabilidade e de interação Humana Computador (Usability and Human Computer Interaction Testing)? O que esse teste pode envolver?

- É um teste que é realizado com a finalidade de saber como o usuário final interage com o software. Com isso tem o desenvolvedor possuirá o feedback do usuário em relação ao projeto.
- O teste pode vir a envolver mostrar as tarefas que o software realiza, documentos guias para o usuário utilizar a plataforma, encontrar bugs que passaram despercebidos, mudar funcionalidade que não agradam o usuário final entre outros.

Página: 4-7

## 8) O que é Heurísticas de Observação de Usuário (User Observation Heuristics)?

- As Heurísticas de Observação de Usuário são orientações para que se possa identificar problemas na interface do projeto. As heurísticas são aplicadas a partir de observações feitas pelo desenvolvedor em relação ao usuário utilizando o software.
- Heurísticas incluem análises, observações, criatividade e até mesmo o feedback do usuário em relação ao software.

Página: 4-10

- 
- **Nome:** Danillo Gonçalves de Souza
  - **Matrícula:** 17/0139981