



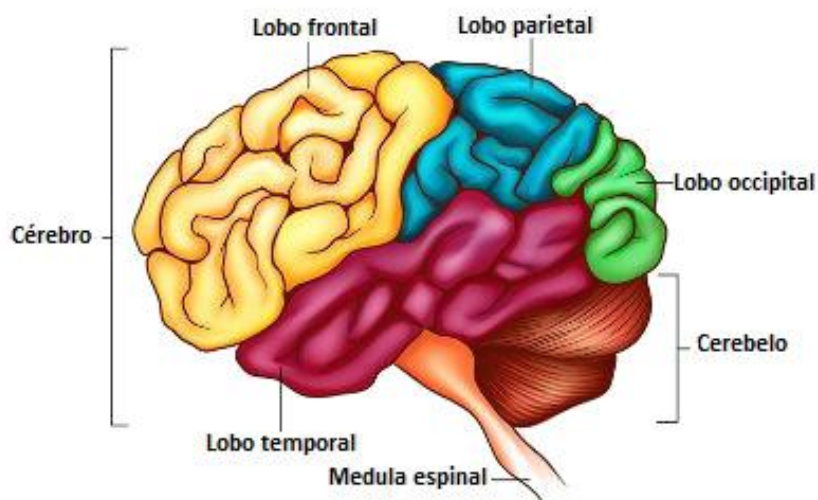
Teste de Software

Prof. MSc. Alan Souza

alan.souza@unama.br

2020

**Um pouco de como o cérebro funciona
(neurociência):**



Um pouco de como o cérebro funciona (neurociência):



Problemas	Propostas de solução
Conversa na aula; muitas faltas	Disciplina!
Estudar para prova apenas em cima da hora	Estudar diariamente! Matéria dada é matéria estudada (HOJE)!
Não praticar sozinho	A prática ajuda a armazenar o conhecimento
Não ler livros	Muito leitura!

Avisos



1. Você vai precisar **praticar** (sozinho)!
2. Não se iluda somente assistindo aulas;
3. Presença em sala: escada;
4. Primeiro prestar atenção na explicação e depois fazer;
5. Quem refizer todos os exemplos e exercícios em casa, terá grandes chances de se dar bem na disciplina.



Avaliações



- 1ª avaliação: 3,0 teste(s) prático(s) + 7,0 prova prática/teórica
- 2ª avaliação: 10,0 Prova Colegiada [questões de múltipla escolha]
- 2ª chamada*: 10,0 prova prática/teórica
- Avaliação Final*: 10,0 prova prática/teórica

*Envolve todo o conteúdo

OBS: Essa metodologia de avaliação pode mudar.

Frequência



- É necessário possuir 75% de presença às aulas no mínimo
- Curso de 60H: máximo de faltas: 15h (5 dias)
- A presença é fundamental para um bom rendimento do aluno!

Teste de Software

Pré-requisitos:

1. Engenharia de Software
2. Qualidade de Software
3. Programação de computadores (Java)
4. Programação Orientada a Objetos (Java)

Livro

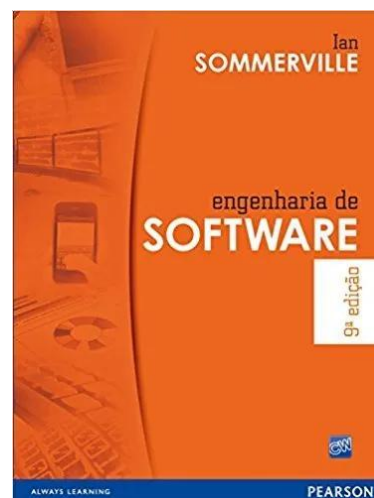
Título: Engenharia de Software

Autor: Ian Sommerville

Editora: Pearson

9ª edição

Capítulo: 8



Livro

Título: Testes Automatizados de Software

Autor: Mauricio Aniche

Editora: Casa do Código



Conteúdo

UNIDADE I:

- FUNDAMENTOS DE TESTES DE SOFTWARE
- TIPOS DE TESTES
- ABORDAGENS DE TESTES
- EXTRAÇÃO DE CASOS DE TESTES
- CRIAÇÃO DE CASOS DE TESTES

Conteúdo



UNIDADE II:

- AUTOMAÇÃO DE TESTES
- MÉTRICAS PARA TESTES DE SOFTWARE
- ESTIMATIVAS PARA TESTES
- PLANO DE TESTE



Teste de Software

Prof. MSc. Alan Souza

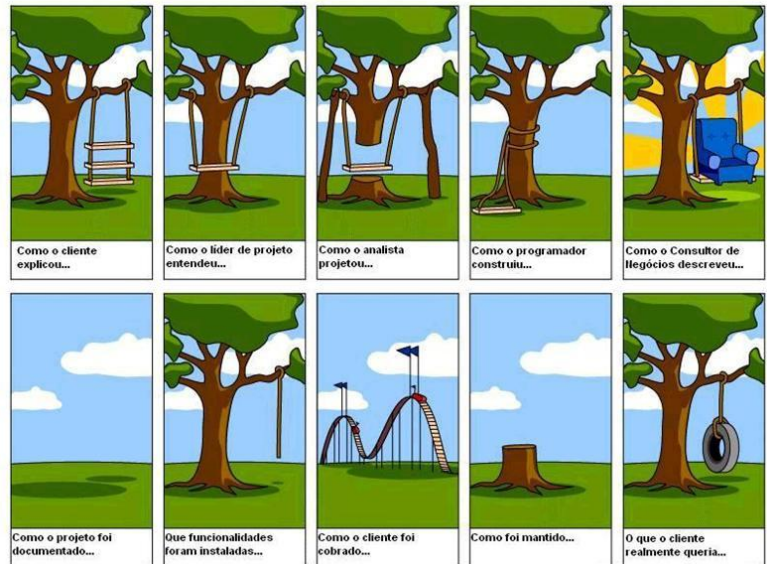
alan.souza@unama.br

2020

1. Fundamentos de Teste de Software

Por que testar software?

a. Mostrar que um programa faz o que é proposto a fazer (*gap* entre o planejamento e a prática) (Sommerville, 2011);



1. Fundamentos de Teste de Software

Por que testar software?

b. Descobrir os defeitos do programa antes do uso (Sommerville, 2011); Notícias interessantes:

. Alguns dos mais famosos erros de softwares da história (último acesso: janeiro/2020)

Link: <https://www.profissaisti.com.br/2012/01/alguns-dos-mais-famosos-erros-de-softwares-da-historia/>

. Por que empresas pagam por falhas em software (último acesso: janeiro/2020)

Link: <http://g1.globo.com/tecnologia/blog/seguranca-digital/post/por-que-empresas-pagam-por-falhas-em-software-g1-explica.html>

1. Fundamentos de Teste de Software



Por que testar software?

- c. Garante a qualidade do sistema;
- d. É uma forma de documentar o sistema (pequeno, médio porte);
- e. Torna o desenvolvimento do sistema mais eficiente;
- f. À medida que o sistema cresce, os testes são importantes para relembrar o programador de como algumas partes do sistema funcionam, caso ele tenha esquecido. Basta rodar a bateria de testes;
- g. Fica muito claro o que é certo e o que é bug para o cliente.

<https://hackerone.com/hacktivity> - Bug bounty!



The screenshot shows the HackerOne Hacktivity page with the following details:

- URL:** <https://hackerone.com/hacktivity?querystring=&filter=type:bounty>
- Navigation:** FOR BUSINESS, FOR HACKERS, HACKTIVITY, COMPANY, TRY HACKERONE
- Sort:** Popular (selected), New
- Type:** All, Bug Bounty (selected), Published, Disclosed
- Table of Bug Reports:**

Score	Icon	Title	By	Status	Severity	Amount	Disclosed
71	Vanilla	Abusing "Report as abuse" functionality to delete any user's post.	By h1_squirtle to Vanilla	Resolved	High	\$300.00	disclosed 2 days ago
576	PayPal	Token leak in security challenge flow allows retrieving victim's PayPal email and plain text password	By alexbirsan to PayPal	Resolved	High	\$15,300.00	disclosed 12 days ago
30	NordVPN	Open redirect	By nickelheck to NordVPN	Resolved	Medium	\$500.00	disclosed 2 days ago
92	Mail.ru	Information disclosure with sensitive data	By mickey01 to Mail.ru	Resolved		\$1,500.00	disclosed 6 days ago
65	NordVPN	Disclosure of User Information	By shardulb_23 to NordVPN	Resolved	Low	\$100.00	disclosed 4 days ago
203	Razer	Reflected XSS at https://pay.gold.razer.com escalated to account takeover	By corraldev to Razer	Resolved	High	\$750.00	disclosed 10 days ago

1. Fundamentos de Teste de Software



Requisitos (SWEBOK)

- Expressam as necessidades e restrições colocadas sobre o produto de software que contribuem para a solução de algum problema do mundo real. Essa 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].

1. Fundamentos de Teste de Software



Requisitos (SWEBOK) - Subáreas:

- **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: cálculos, processamentos em geral, cadastros de dados.
- **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).

1. Fundamentos de Teste de Software



Requisitos (SWEBOK) - 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, ambiente operacional e organizacional.
- **Técnicas de elicitação:** concentra-se em técnicas para obter as necessidades das partes interessadas.

1. Fundamentos de Teste de Software



Requisitos (SWEBOK) - Análise de Requisitos:

- “Se você não analisa, é altamente provável que construa uma solução de software muito elegante que resolve o problema errado”(Pressman, 2002).
- Uma vez que cada pessoa envolvida na construção do software contribui efetuando uma “pequena” modificação..
- Isso pode resultar em perda de tempo e dinheiro, pessoas frustradas e clientes insatisfeitos.

1. Fundamentos de Teste de Software



Requisitos (SWEBOK) - Análise de Requisitos - 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.

1. Fundamentos de Teste de Software



Requisitos (SWEBOK) - Análise de Requisitos - Subáreas:

- **Projeto arquitetural e distribuição de requisitos:** é o ponto no qual são descritos os requisitos do projeto de software, ilustrando as diferenças entre essas 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.

1. Fundamentos de Teste de Software



Requisitos (SWEBOK) - Especificação de Requisitos:

- A principal atividade dessa fase é a criação 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** listagem dos requisitos do 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 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.

1. Fundamentos de Teste de Software



Requisitos (SWEBOK) - Validação de Requisitos:

- 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 Subáreas:
- **Revisão dos requisitos:** um grupo de revisores efetua uma avaliação preliminar de erros, com o desenvolvimento de checklists de revisão;
- **Prototipação:** é normalmente um meio para validar a interpretação do desenvolvedor sobre os requisitos de software;

1. Fundamentos de Teste de Software



Requisitos (SWEBOK) - Validação de Requisitos:

- **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.

1. Fundamentos de Teste de Software



Requisitos (SWEBOK) - Exercício:

1.1 Considere um software de pedido e entrega de comida, por exemplo, o iFood, Uber Eats ou Rappi. Escreva as seguintes informações a respeito dos requisitos deles:

- a) Requisitos do produto
- b) 03 Requisitos do processo
- c) 03 Requisitos funcionais
- d) 03 Requisitos não funcionais

1. Fundamentos de Teste de Software

Requisitos (SWEBOK) - Exercício (continuação):

1.2 Use o modelo a seguir para detalhar um requisito funcional:

Requisito	
Ator	
Descrição	
Pré-condição	
Pós-condição	
Fluxo principal	
Validação	

1. Fundamentos de Teste de Software

Requisitos (SWEBOK) - Exercício (continuação):

1.3 Crie um prótipo da tela de um dos requisitos.



Fonte: Livro "Empreendedorismo Inovador" - Capítulo 15
 Exemplo de protótipo tipo "Mockup" - balsamiq.com