

# Engenharia de Software – 2020/2 EARTE

Jordana S. Salamon

[jordana.salamon@ufes.br](mailto:jordana.salamon@ufes.br)

[jssalamon@inf.ufes.br](mailto:jssalamon@inf.ufes.br)

[jordanasalamon@gmail.com](mailto:jordanasalamon@gmail.com)

<http://inf.ufes.br/~jssalamon>

Departamento de Informática

Centro Tecnológico

Universidade Federal do Espírito Santo



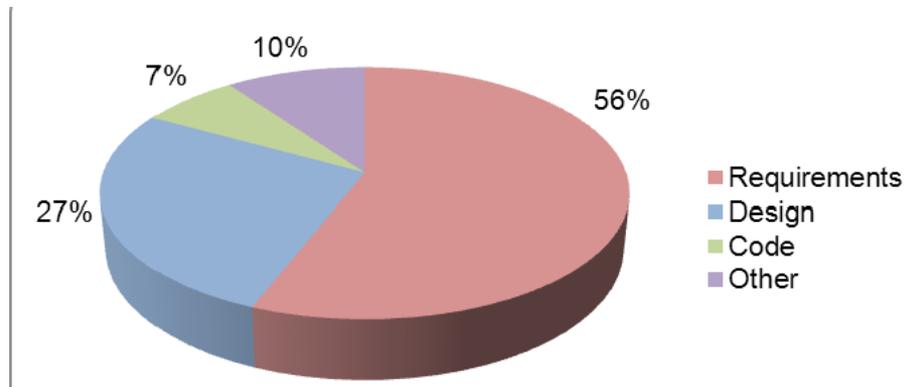
# Teste baseado em Requisitos

- Muitos estudos mostram que a maioria dos projetos de software não atinge os objetivos de cronograma e orçamento.
- A má qualidade do software é uma das principais razões por trás de muitas falhas. Isso geralmente resulta em um grande retrabalho dos requisitos, design e código da aplicação.
- A experiência e vários estudos mostram que: por trás da baixa qualidade do software estão os defeitos nas especificações dos requisitos e a cobertura de testes problemática do sistema. Simplificando, a baixa qualidade de entrada causa baixa qualidade de saída, não importa a experiência da equipe do projeto, qual metodologia é usada e quais restrições de orçamento e cronograma são estabelecidas.

Fonte: [1]

# Teste baseado em Requisitos

- De acordo com James Martin, a causa raiz de 56% de todos os defeitos identificados em projetos de software são introduzidos na fase de requisitos.
- Cerca de 50 por cento dos defeitos de requisitos são resultado de requisitos mal escritos, pouco claros, ambíguos e incorretos. Os outros 50 por cento dos defeitos de requisitos são devido à incompletude da especificação (ou seja, requisitos omitidos). [2]



Fonte: [2]

# Teste baseado em Requisitos

- Outras estatísticas apontam para problemas semelhantes [3]:
  - 82 por cento do retrabalho em aplicações está relacionado a erros nos requisitos.
  - Problemas nos requisitos representam 44% dos motivos por trás dos cancelamentos de projetos.
  - Apenas 54% dos requisitos iniciais do projeto são realmente realizados.
- Não é incomum que um sistema testado cuidadosamente e com sucesso deixe seus usuários insatisfeitos. A razão por trás desse fato é que a equipe de desenvolvimento entendeu os requisitos errados.

Fonte: [3]

# Teste baseado em Requisitos

- O foco do RBT (*Requirements-based Testing*) é descobrir e corrigir requisitos de baixa qualidade, tornando assim dados válidos que contribuem muito na definição do escopo claro do projeto.
- Ao combinar métodos de engenharia de requisitos e teste de software, a metodologia de teste baseada em requisitos fornece um conjunto de atividades de garantia de qualidade e ferramentas de gerenciamento que permitem obter os requisitos desde o início.
- Usando o RBT, é possível descobrir erros de requisitos antes que se tornem extremamente caros para corrigir e gerenciar mudanças inevitáveis durante o ciclo de vida do software.

# Teste baseado em Requisitos

- O processo de teste baseado em requisitos aborda duas questões principais: primeiro, avaliar se os requisitos estão corretos, completos, não ambíguos e logicamente consistentes; e, segundo, projetar um conjunto necessário e suficiente (de uma perspectiva de caixa preta) de casos de teste a partir desses requisitos, para garantir que o design e o código atendam totalmente aos requisitos.
- Ao projetar os testes, dois problemas precisam ser superados: reduzir o enorme número de testes potenciais para um conjunto de tamanho razoável e garantir que os testes tenham a resposta certa pelo motivo certo.
- O processo RBT não assume que teremos boas especificações de requisitos.

Fonte: [1]

# Teste baseado em Requisitos

- A estratégia geral do RBT é integrar os testes ao longo do ciclo de vida de desenvolvimento e focar na qualidade da especificação dos requisitos.
- Isso leva à detecção precoce de defeitos, que se mostrou muito menos dispendiosa do que encontrar defeitos durante o teste de integração ou posteriormente.
- O processo RBT também tem foco na prevenção de defeitos, não apenas na detecção de defeitos [4].

Fonte: [4]

# Teste baseado em Requisitos

- A metodologia RBT é um processo de 12 passos [5], [6]:
  1. *Valide os requisitos em relação aos objetivos.*
  2. *Aplique os casos de uso aos requisitos.*
  3. *Faça uma revisão inicial de ambiguidade.*
  4. *Realize análises com especialistas de domínio.*
  5. *Estruturar e formalizar requisitos.*

# Teste baseado em Requisitos

- *6. Verificar consistência lógica e projetar casos de teste.*
- *7. Revisão de casos de teste pelos autores de requisitos.*
- *8. Validar os casos de teste com os usuários / especialistas do domínio.*
- *9. Revisão dos casos de teste pelos desenvolvedores.*
- *10. Usar casos de teste na revisão de design.*
- *11. Usar casos de teste na revisão de código.*
- *12. Verificar o código em relação aos casos de teste derivados dos requisitos.*

# Teste baseado em Requisitos

- A rastreabilidade também desempenha um papel crítico se estiver usando RBT, uma vez que manter as informações de rastreabilidade entre requisitos e casos de teste lógicos e testes é crucial.
- Essas informações são necessárias para monitorar o progresso e a cobertura, bem como gerenciar adequadamente o impacto das mudanças nos requisitos.
- Sem ele, é mais difícil determinar quais casos de teste ou testes devem ser alterados se um requisito específico for alterado.

Fonte: [1]

# Documentando Casos de Teste

- Informações importantes a serem registradas sobre casos de teste:
- *Código Identificador;*
- *Nome do teste;*
- *Descrição do teste;*
- *Prioridade do teste;*
- *Nome do projetista do teste;*
- *Data do projeto do teste;*
- *Pré-condições;*
- *Dependências;*
- *Passos do teste;*
- *Dados usados como entrada para o teste;*
- *Resultados esperados;*
- *Pós-condições;*
- *Notas adicionais;*

Fonte: <https://www.guru99.com/download-sample-test-case-template-with-explanation-of-important-fields.html>

# Referências

- [1] SKOKOVIĆ, Predrag; RAKIĆ-SKOKOVIĆ, Marija. Requirements-based testing process in practice. International Journal of Industrial Engineering and Management (IJIE), v. 1, n. 4, p. 155-161, 2010.
- [2] Martin, J.,(1984), "An Information Systems Manifesto".
- [3] Borland, (2006), "Eliminate the Testing Bottleneck".
- [4] Bender RBT Inc., (2003), "Requirements Based Testing, Process Overview".
- [5] Mogyorodi, G., (2003), "What Is Requirements-Based Testing?".
- [6] Aharonovitz, M., (2008), "Three Tips to Improve Your Requirements-Based Testing (RBT)".
- SOMMERVILLE, I. Engenharia de Software. 10. ed. São Paulo: Pearson Prentice Hall, 2019

# Engenharia de Software – 2020/2 EARTE

Jordana S. Salamon

[jordana.salamon@ufes.br](mailto:jordana.salamon@ufes.br)

[jssalamon@inf.ufes.br](mailto:jssalamon@inf.ufes.br)

[jordanasalamon@gmail.com](mailto:jordanasalamon@gmail.com)

<http://inf.ufes.br/~jssalamon>

Departamento de Informática

Centro Tecnológico

Universidade Federal do Espírito Santo

