2

TEXTO BASE

QUALIDADE DE SOFTWARE

Faculdade IMPACTA

Texto base

1.2

Estratégias de Teste de Software

Marco Túlio Jeunon

Resumo

O melhoramento do processo que busca a garantia do produto de software se dá através de um processo robusto e que lhes dê domínio sobre as técnicas a serem utilizadas. Nesta aula o objetivo é apresentar alguns conceitos importantes, verificação e validação, para que em conjunto com as estratégias de teste caixa branca e caixa preta, permita ao aluno entender quais são os testes que deverão ser realizados nas diversas fases do ciclo de vida de desenvolvimento e manutenção de um software.

1.1. Entendendo o conceito de Verificação

É definido como um processo de avaliação dos artefatos gerados nas fases iniciais de um projeto de software. A verificação deve ser utilizada para validar todos os documentos, gráficos, manuais, códigos fonte, gerados em cada fase do ciclo de vida de um software, para que não gere nenhuma dúvida ou questão mau resolvida propague para outra fase. A verificação não envolve nenhum recurso de execução do software no computador, portanto tenta garantir que algum erro de definição ou criação seja propagado para outra fase. Exemplo de fases a serem verificadas:

- Requisito;
- Análise;
- Arquitetura;
- Desenvolvimento.

Dentre as validações a serem realizadas, podemos destacar:

- Reuniões de inspeções, também chamadas de revisões técnicas, baseando em reuniões de revisão do documento de análise, procurando identificar possíveis erros. É uma reunião formal e os participantes devem se preparar previamente para realização da mesma;
- Reuniões de acompanhamento, não é tanto formal quanto a reunião de inspeção, pois não precisam de preparação prévia dos participantes. O objetivo principal é a disseminação do conteúdo entre os participantes, para que todos tenham entendimento comum, a identificação de algum erro é um objetivo secundário;
- Revisão por outro profissional, ou seja trata se de uma revisão individual do profissional diferente do autor, com o objetivo da identificação de algum problema.
- *Checklist*, é um poderoso instrumento de verificação, através do qual são checados vários pontos devidamente estruturados, para que seja identificada a sua completude ou não.

1.2. Entendendo o conceito de Validação

É definido como um processo de avaliação de uma aplicação (sistema) ou seus componentes de software, a fim de validar se seu desenvolvimento está de acordo com as especificações de requisitos funcionais e não funcionais, validado nas fases iniciais do projeto. Na validação será necessária a execução do software desenvolvido. Nestas fases é que são identificados os principais erros sistêmicos. Exemplo dos testes que devem ser efetuados para que se tenha a validação:

- Teste Unitário é o teste do desenvolvedor do software, requer conhecimento das estruturas internas, com esse teste ele tenta garantir que nenhum erro seja propagado para as próximas fases de teste;
- Teste Integrado é o teste que é realizado para se garantir que os componentes ou programas testados individualmente, sejam testados se integrando com as outras partes, requer conhecimento da arquitetura interna da aplicação.
- Teste de homologação, esses testes são realizados no sistema como um todo, requer uma ambiente similar ao de produção e são realizados por uma equipe independente da equipe de desenvolvimento.
- Teste de aceitação, teste similar ao teste de homologação, porém normalmente são executados pelos usuários finais ou eles indicam quais as situações de teste devam ser evidenciadas para que eles validem e são executadas pela equipe de testes.

1.3. Entendendo o Modelo V

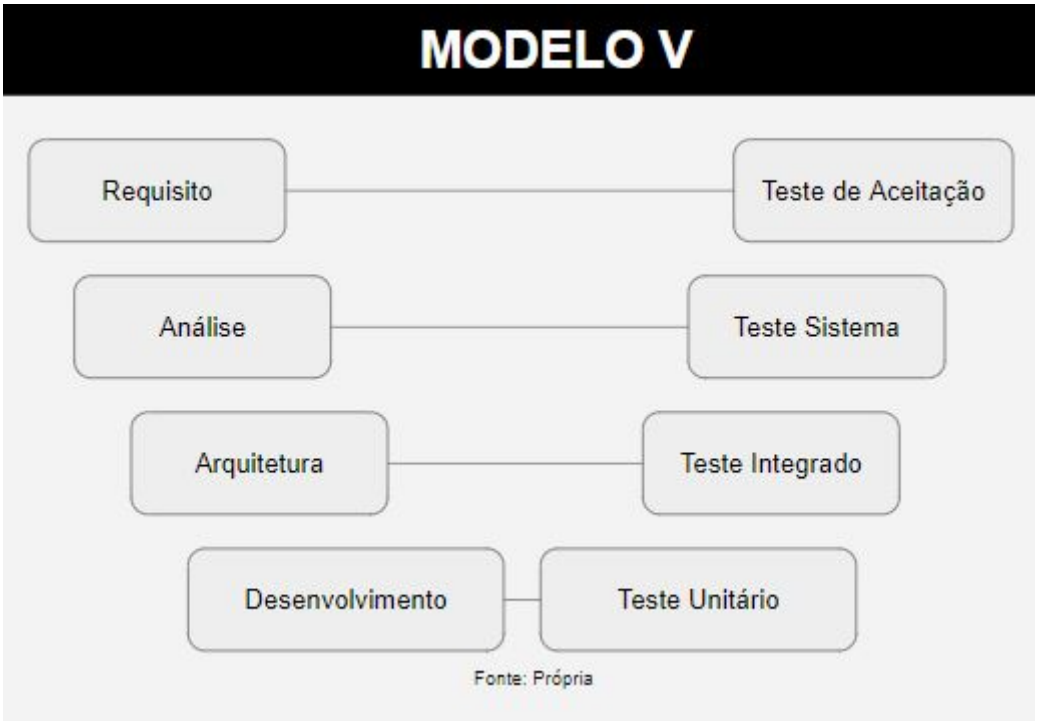

Figura 1.3.1. Modelo V

A figura 1.3.1 representa o modelo "V" que relaciona as fases do ciclo de vida com os testes a serem realizados. Como estratégia da verificação, devemos realizar a verificação de cada artefato do ciclo de vida, requisito, análise, arquitetura e desenvolvimento, bem como a execução da validação, que é a execução dos testes unitários, de integração, homologação e aceitação.

1.4. Estratégia Caixa Branca

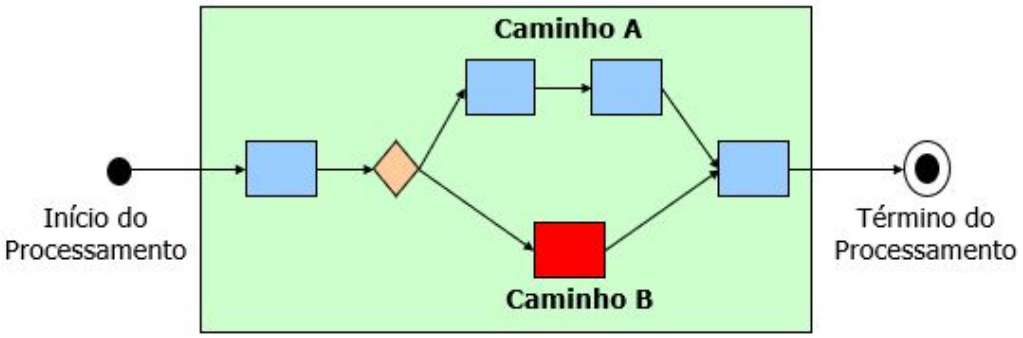

Figura 1.4.1. Estratégica Caixa Branca (Fonte: própria)

A figura 1.4.1, representa a estratégia caixa branca. Essa estratégia deve ser aplicada aos testes unitários e testes integrados e tem como objetivo identificar defeitos nas partes internas dos programas, procurando localizar algum código inalcançável, loop infinito ou erros no código.

1.5. Estratégia Caixa Preta

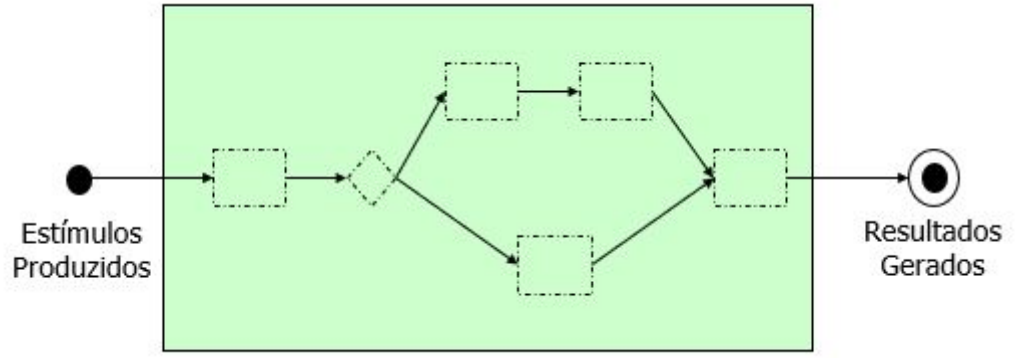

Figura 1.4.2. Estratégica Caixa Preta (Fonte própria)

A figura 1.4.2 representa a estratégia caixa preta, ou seja não importa a estrutura interna do software. Essa estratégia deve ser aplicada aos testes de homologação e aceitação do sistema e tem como objetivo principal garantir que os requisitos do sistema sejam plenamente atendidos. Para execução desses testes não há necessidade de conhecimento tecnológico e normalmente são executados pelas equipes de teste e usuários do sistema.

1.6. Teste progressivo

A medida que um software evolui, ou seja adquire ou necessita de manutenção nas funcionalidades existentes, um novo conjunto de testes devem ser criados ou revistos para que garanta que não tenham falhas ou defeitos, portanto somente as inovações são testadas.

1.7. Teste regressivo

A medida que um software evolui, mesmo que algumas funcionalidades do sistema não tenham sido impactadas pela manutenção, devemos assegurar que as novas implementações não produziram falhas ou defeitos. Para tanto para garantir que o software continua funcionando normalmente, devemos retestar essas funcionalidades.

1.8. Referências

PEZZE, M.& YOUNG, M. Teste e Análise de Software: processos, princípios e técnicas. Porto Alegre: Bookman, 2008. ISBN: 978-85-778-0262-3.

BARTIE, A. Garantia da Qualidade de Software. Rio de Janeiro: Elsevier, 2002. ISBN: 978-85-352-1124-5.

DELAMARO, M. E.; MALDONADO, J.C. & JINI, M. Introdução ao teste de software, Rio de Janeiro: Elsevier, 2007. ISBN: 978-85-352-2634-8.

BECK, K. TDD - Desenvolvimento Guiado por Testes. Porto Alegre: Bookman, 2010. ISBN: 978-85-778-0724-6.

HUMBLER, J. & FARLEY, D. **Entrega contínua**. Porto Alegre: Bookman, 2014. https://integrada.minhabiblioteca.com.br/#/books/9788582601044