

Testes de software na norma ISO/IEC 25051

Joana Vilas Boas

Instituto Politécnico do Cávado e Ave – Escola de
Tecnologia
Barcelos, Portugal

Nuno Lopes

Instituto Politécnico do Cávado e Ave – Escola de
Tecnologia
Barcelos, Portugal

Resumo — O teste de software é uma das atividades que faz parte do desenvolvimento de software, e tem como objetivo apresentar evidências de confiabilidade contribuindo para uma maior qualidade do software. Para auferir mais qualidade ao software, este pode ser certificado por uma norma de qualidade. As normas têm um papel importante na definição dos requisitos de teste, casos de teste e relatórios de teste que contemplam a atividade de testes permitindo elaborar um plano de testes mais rigoroso.

No desenvolvimento de software é fundamental seguir normas de qualidade para criar métodos coerentes, rigorosos e uniformes. As normas a adotar devem abranger três características, a norma deve ser necessária, deve ser aplicada e demonstrar que a mesma está a ser seguida.

Neste artigo vão ser analisados e identificados os testes de software necessários no âmbito da certificação do software Cypeterm aplicável à avaliação do desempenho energético de edifícios pela norma ISO/IEC 25051. Esta norma abrange os requisitos de qualidade para produtos de software e avaliação (SQuaRE), requisitos para a qualidade de produtos de software pronto a utilizar (RUSP) e instruções para testes.

Palavras-chave – Testes de Software; Certificação de software ISO/IEC 25051

1. Introdução

Testar Software engloba um conjunto de atividades com o objetivo de encontrar erros no software antes da entrega ao cliente. Estas atividades devem ser planeadas antecipadamente e conduzidas

sistematicamente, por esta razão é necessário um modelo com uma sequência de passos onde são definidos os desenhos de caso de testes e métodos para executar. Esta atividade consome uma parte significativa do esforço de um projeto de desenvolvimento de software, sempre com o objetivo de encontrar erros antes da fase de manutenção, pois o custo de correção nesta fase pode ser até 100 vezes maior (Pressman, 2010).

Os testes “têm por objetivo identificar o maior número possível de erros tanto nos componentes isolados quanto na solução tecnológica como um todo” (Bartié, 2002), e para isso existem basicamente dois tipos de abordagem diferentes, o “*black-box testing*” e o “*white-box testing*”. Normalmente o *white-box testing* tende a ser utilizado no início dos testes e o *black-box testing* mais no final dos testes, devido às suas especificações (Jovanovic, n.d.). Por não existir a possibilidade de acesso ao código, os testes a considerar no caso de estudo apresentado são do tipo *Black-box*. Esta abordagem é utilizada quando se conhecem as especificações do produto, de modo que os testes possam ser conduzidos para demonstrar que cada função está totalmente operacional ao mesmo tempo que se procuram os erros (Sirohi & Parashar, 2013).

Para este processo é fundamental seguir normas de qualidade para criar métodos coerentes, rigorosos e uniformes. As normas são desenvolvidas por instituições internacionais como a ISO ou associações profissionais IEEE. As normas a adotar devem abranger três características, a norma deve ser necessária, deve ser aplicada e demonstrar que a mesma está a ser seguida. O produtor do software ao

aplicar normas vai dar mais atenção à parte técnica da tarefa. A aplicação de normas deve estar bem definida e clara perante os intervenientes no desenvolvimento do software (António Miguel, 2006). A norma que será utilizada neste artigo será a ISO/IEC 25051, esta norma abrange os requisitos de qualidade para produtos de software e avaliação (SQuaRE), requisitos para a qualidade de produtos de software pronto a utilizar (RUSP) e instruções para testes.

O software Cypeterm aplicável à avaliação do desempenho energético de edifícios (TOP Informática, 2014) será apresentado no caso de estudo, este software foi submetido à certificação com a norma ISO/IEC 25051 do qual obteve o certificado de qualidade.

2. Requisitos de testes na norma ISO/IEC 25051

A norma ISO/IEC 25051 abrange os requisitos de qualidade para produtos de software e avaliação (SQuaRE), requisitos para a qualidade de produtos de software pronto a utilizar (RUSP) e instruções para testes.

Um produto de software pronto a utilizar (RUSP) deve conter: uma descrição do produto que deve apresentar toda a informação necessária para potenciais adquirentes e utilizadores do produto e documentação de utilizador que apresenta a informação necessária para a utilização do software.

Um software tem qualidade quando cumpre uma série de requisitos, nomeadamente: funcionalidade, fiabilidade, usabilidade, eficiência, manutenibilidade, portabilidade e qualidade na utilização.

Na norma ISO/IEC 25051 não são recomendados quaisquer métodos ou técnicas de teste específicos. A documentação de teste deve conter: o plano de teste, a descrição do teste e os resultados do teste. Antes de se proceder aos testes é necessário descrever todas as funções do programa, uma função é uma implementação de um algoritmo no software com o qual o utilizador final ou o software podem realizar parte ou a totalidade de uma tarefa de trabalho.

Cada função descrita deve ser objeto de pelo menos um caso de teste. Todos os procedimentos de instalação devem ser submetidos a casos de teste. Todos os limites operacionais indicados na descrição do produto e na documentação de utilizador devem ser submetidos a casos de teste. Violações identificadas de condições sintáticas para entrada devem ser submetidas a casos de teste. Se são

indicados exemplos na documentação de utilizador, estes devem ser utilizados como casos de teste mas a totalidade do teste não deve ser limitada a estes exemplos. Se qualquer requisito de qualidade para o software, não for aplicável, a razão deve ser declarada.

A descrição de cada caso de teste deve incluir: o seu objetivo, um identificador único, dados de entrada e limites, os passos detalhados para a realização, o comportamento esperado do sistema, a saída esperada do caso de teste, os critérios para a interpretação do resultado, os critérios utilizados para decidir sobre um resultado positivo ou negativo do caso de teste.

Depois de identificado o que se vai testar, é necessário identificar os critérios para decisão sobre o resultado, estes critérios devem estar em conformidade com a documentação do produto e a documentação do utilizador. No caso de testes não conformes, após a sua correção, deve existir um procedimento para repetir o ensaio das funções envolvidas e de quaisquer funções relacionadas.

O ambiente de teste (hardware e software) onde se vão realizar os testes, deve ser especificado no plano de testes. O software deve ser testado em todas as configurações de aplicação mencionadas na descrição do produto.

O procedimento de teste deve incluir: a preparação do teste, as ações necessárias para iniciar e executar o teste, as ações necessárias para registar os resultados do teste, as condições e ações para parar e eventualmente reiniciar os testes. Estes procedimentos devem ser suficientemente detalhados para proporcionar repetibilidade e reprodutibilidade dos testes.

O relatório de execução deve incluir um resumo global dos resultados dos casos de teste, deve demonstrar que todos os casos de teste foram executados de acordo com o plano de teste. Este relatório deve incluir: o identificador do caso de teste, a data da execução do teste, o nome e a função da pessoa que realizou o teste, a lista das anomalias encontradas e para cada anomalia a referência ao relatório de anomalia correspondente.

O relatório de anomalias deve incluir um resumo global das anomalias encontradas e, caso existam, as correções e as verificações por repetição do ensaio. Este relatório deve ainda incluir para cada anomalia: o identificador da anomalia, o identificador do software, a descrição da anomalia, o ponto no caso de teste

onde a anomalia ocorreu, a natureza da anomalia (exemplo: “impeditiva”, “maior”, “menor”).

A avaliação do relatório de execução e do relatório de anomalias deve demonstrar que foram obtidos todos os comportamentos esperados, dentro dos limites dos critérios utilizados para decidir se os resultados do teste mostram conformidade do software.

3. Caso de estudo

O software Cypeterm foi submetido à certificação pela norma ISO/IEC 25051 tendo obtido o certificado de qualidade (TOP Informática, 2015). Os testes realizados vão ao encontro dos requisitos apresentados anteriormente, como exemplo serão descritos uma série de testes realizados ao software para garantir a qualidade do mesmo.

Antes de começar o teste ao software, é necessário analisar todas as funções da aplicação, para criar um plano de testes. O plano de testes descreve todas as funções, neste caso foram descritas 367 funções (exemplo: Função: 1 - Abrir obra; Função: 2 - Criar obra; Função: 3 - Definir orientação da obra; etc.). O passo a seguir é definir os casos de teste para cada função, isto é, uma função pode ter mais que um caso de teste. Por exemplo, para a Função: 1 - Instalar a versão profissional é necessário analisar todos os cenários possíveis de instalação e funcionamento da versão profissional e traduzir isso para casos de teste. Nesta função foram criados seis casos de teste, um para cada versão do sistema operativo onde o software é compatível com os requisitos mínimos de hardware e software definidos.

Para cada caso de teste é obrigatório ter pelo menos um registo de teste, este registo deve ter referência a data de realização, o executor do teste, a função do executante, o ambiente de teste, a versão do software a testar e o resultado do teste, conforme ou não conforme. Em cada registo de teste é obrigatório anexar os comprovativos de teste, os comprovativos podem ser imagens, obras, alterações realizadas ou outro tipo de ficheiro. No caso do registo de teste ser não conforme é necessário registar e associar o erro ao registo de teste, este erro será enviado à programação para correção.

Este processo garante que existe uma série de registos e evidências de que todas as funções foram testadas e validadas antes do software ser comercializado e que se encontra em conformidade com a descrição do produto e a documentação de utilizador. Os erros não corrigidos e que não

impossibilitam a saída do software para comercialização ficam registados em lista de limitações para o utilizador ter conhecimento antes da aquisição do software.

4. Conclusão

A área de testes de software é fundamental no desenvolvimento de software com qualidade para dar cumprimento aos requisitos e às expectativas dos utilizadores. Os erros detetados quando o software está em comercialização chamam-se defeitos, e podem trazer custos avultados na sua correção e prejuízos para os utilizadores.

Como os produtos de software RUSP são utilizados numa variedade de áreas de aplicação cada vez mais vastas e a sua operação correta é muitas vezes vital para aplicações de negócios, segurança ou pessoais, a certificação atribui uma maior qualidade, confiabilidade e disponibilidade perante o utilizador com a certeza que o software foi submetido a testes rigorosos para garantir todas as especificações definidas.

Referências

António Miguel, P. (2006). *Gestão de Projectos de Software*. (FCA - Editora Informática, Ed.).

Bartié, A. (2002). *Garantia da qualidade de software*. CAMPUS - RJ, 2002.

ISO. (2014). ISO/IEC 25051. Retrieved from http://www.iso.org/iso/catalogue_detail.htm?csnumber=61579

Jovanovic, I. (n.d.). *Software Testing Methods and Techniques. The IPSI BgD Transactions on Internet Research* (pp. 30–41).

Pressman, R. S. (2010). *Software Engineering A Practitioner's Approach*.

Sirohi, N., & Parashar, A. (2013). Component Based System and Testing Techniques, 2(6), 2378–2383.

TOP Informática. (2014). Retrieved from TOP Informática:
http://servicos.topinformatica.pt/fich/manuaiscype/cype_term_manual_do_utilizador_exemplos_praticos_v2014.pdf

TOP Informática. (2015). Retrieved from TOP Informática:
<http://www.topinformatica.pt/index.php?cat=12&keyword=25051>