

Testes de software no âmbito da certificação de aplicações de e-commerce

Software testing under the e-commerce applications certification

Raquel Gaio, DIGARC, Instituto Politécnico do Cávado e Ave, Portugal, raquelgaio@gmail.com

Nuno Lopes, DIGARC, Instituto Politécnico do Cávado e Ave, Portugal, nlopes@ipca.pt

João Carlos Silva, DIGARC, Instituto Politécnico do Cávado e Ave, Portugal, jcsilva@ipca.pt

Resumo

O principal objetivo deste artigo consiste em propor uma abordagem para a certificação de aplicações de e-commerce, considerando várias normas de certificação. Pretende-se assim apresentar uma revisão de literatura bem como propor uma abordagem para o âmbito da certificação de aplicações de e-commerce fazendo uso da automatização de testes de software. Pretende-se apresentar as normas e os respetivos requisitos de qualidade e discutir a respetiva aplicação no âmbito de aplicações de e-commerce. Com a implementação da abordagem descrita neste artigo espera-se que a inclusão de processos automáticos nos testes permita certificar as aplicações de e-commerce. Como resultado, pretende-se um modelo automático de testes de aplicações de e-commerce, permitindo avaliar a qualidade das aplicações com base nas normas de certificação.

Palavras-chave: certificação; e-commerce; teste de software

Abstract

The main purpose of this paper is to propose an approach for the certification of e-commerce applications, considering various certification standards. The aim is to present a literature review and to describe an approach to the certification of e-commerce applications making use of automated software testing. We intend to present the standards and their respective quality requirements to discuss their application in the context of e-commerce applications. With the implementation approach described in this article it is expected that the inclusion of automatic testing processes will facilitate the certification of e-commerce applications. As a result, the aim is to specify an automatic model of e-commerce application testing, to assess the quality of applications based on certification standards.

Keywords: certification; e-commerce; software testing

1. INTRODUÇÃO

Estima-se que em 2018 serão realizadas em todo o mundo 125 mil milhões de transações para pagamento de compras *online*, segundo o estudo “*Mobile & Online Purchases: Cards, Carrier Billing and Third Party Payment Platforms 2015-2020*” da Juniper Research (Juniper Research, 2016). Estes valores traduzem o crescimento exponencial do comércio eletrónico. A designação de comércio eletrónico ou *e-commerce*, refere-se a todas as transações comerciais realizadas através de equipamento eletrónico. Levar a cabo estas transações eletronicamente proporciona amplas vantagens competitivas sobre os métodos tradicionais (Janice Reynolds, 2004). Quando devidamente implementado, o *e-commerce* é mais rápido, mais barato e mais conveniente do que os métodos tradicionais de transação de bens e serviços.

Hoje em dia a sociedade está ciente da importância dos sistemas de software, uma vez que os usamos no quotidiano, em casa, aplicações de negócio, finanças, saúde, estando também mais atentos aos possíveis

erros que estas poderão causar. Nem todos os sistemas de software têm o mesmo impacto quando ocorrem problemas, dependem do seu contexto, e do nível do risco (Dasso, A e Funes, A. 2007) (Klaus Lochmann e Andreas Goeb. 2011) (Zeeshan Ali Rana, Shafay Shamil, e Mian Muhammad Awais. 2008).

Com a ajuda de testes de software é possível medir a qualidade do software em termos de defeitos encontrados. Os testes permitem dar mais confiança relativa à qualidade do software, uma vez que um teste bem elaborado, ao passar com sucesso poderá reduzir o nível de risco geral do sistema (Tian, 2005).

A atividade de Teste de software consome uma parte significativa do esforço de um projeto de desenvolvimento de software, sempre com o objetivo de encontrar erros antes da fase de produção ou manutenção, pois o custo de correção nestas fases pode ser até 100 vezes maior (Pressman, 2010).

O principal objetivo deste artigo consiste em discutir o impacto da automatização de testes na qualidade do software, especificamente em aplicações de *e-commerce*, considerando as normas de certificação. Pretende-se assim apresentar uma revisão da literatura (secção 2) bem como caracterizar os testes de software para o âmbito da certificação de aplicações de *e-commerce* (secção 3). Na conclusão define-se o trabalho futuro com base na abordagem descrita (secção 4).

2. ESTADO DA ARTE

O estado da arte aborda os tópicos de testes de software; técnicas de teste; normas de qualidade em software e finalmente a certificação de software.

2.1. Testes de Software

O conceito de teste de software foi introduzido pela primeira vez, durante a conferência da NATO, que ocorreu em 1968, em Garmisch, Alemanha, pelos oradores, A.I. Llewelyn e R.F. Wicken, onde defendiam a ideia de que as consequências de um software inadequado poderiam ser desastrosas e que o software deveria ser submetido a uma disciplina de testes como qualquer outro produto (A.I. Llewelyn e R.F. Wickens, 1968).

Para Crespo et al. (2004), teste de software é o processo que permite executar o software de uma forma controlada com objetivo de avaliar se o mesmo se comporta conforme o esperado. Testar software engloba um conjunto de atividades com o objetivo de encontrar erros no software antes da entrega ao cliente. Os testes permitem encontrar comportamento inadequado considerando a especificação do sistema e/ou erros inseridos na conceção e/ou implementação dos requisitos. Os testes de software devem ser planeados antecipadamente e conduzidos 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 (Pressman, 2010).

Não se pode descrever a atividade de teste como uma única atividade, mas sim uma série de atividades desenvolvidas durante o ciclo de vida do software, o que pode ajudar a prevenir a inserção de defeitos no código. A atividade de testes é a que mais tempo consome num projeto de desenvolvimento de software, isto

é, em média 40% do esforço total. em contrapartida se os defeitos surgirem na fase de manutenção, o custo da correção pode ser de 60 a 100 vezes maior (Pressman, 2010).

As pessoas cometem erros, que se traduzem em defeitos no código do programa ou documento e posteriormente caso não sejam detetados resultam em falhas de software, o programa funciona de forma diferente do esperado (Dasso, A. e Funes, A. 2007).

2.2. Técnicas de teste de software

As técnicas puderam ser utilizadas na conceção dos casos de testes. Estas técnicas fornecem mecanismos que podem ajudar na elaboração dos casos de teste e fornecem uma probabilidade maior de descobrir erros (António Miguel, P. 2006). Essencialmente o produto poderá ser testado através de duas técnicas principais: técnica caixa preta e técnica caixa branca (Dasso, A., e Funes, A. 2007).

Os testes de software incorporam dois tipos distintos de testes, i.e. testes do tipo caixa preta ou caixa branca (Sirohi e Parashar, 2013). O primeiro tipo de técnica, caixa preta, tenta encontrar, por exemplo, erros de interface, erros de estrutura de dados ou de acesso a bases de dados, erros de comportamento e desempenho. Permite efetuar testes, funcionais ou não funcionais, sem referência à estrutura interna do componente ou sistema. (ISTQB, 2011). Pode ser usada durante todo o ciclo de vida de testes, no entanto tem maior ênfase nos últimos estágios de teste, a informação relativa ao código poderá ser nenhuma ou pouca. O segundo tipo de técnicas, testes do tipo caixa branca, ou testes baseados na estrutura, são testes baseados no acesso ao código e/ou especificações do sistema (ISTQB, 2011). Planear e conceber eficazmente testes do tipo caixa branca requer diversos conhecimentos, nomeadamente linguagem de programação utilizada, bases de dados, sistema operacional utilizado, conhecimento dos ciclos e condições do software e lógica interna do programa.

2.3. Certificação de software

O processo de desenvolvimento de software não é uma tarefa fácil, associando ainda o facto de que as empresas ou organizações estão cada vez mais dependentes da tecnologia e a procura de soluções eficazes de modo a reduzir custos e ampliar a forma de atuação. Neste contexto as variáveis envolvidas no processo de desenvolvimento têm um nível de complexidade crescente tornando difícil produzir software com um nível de qualidade desejado. As normas que podem ser consideradas neste contexto são as seguintes:

- ISO 8402 Noção de qualidade de um produto, processo ou serviço;
- ISO/IEC 9126 Engenharia de software - Qualidade de produto, edição de 2000;
- ISO/IEC 9126-1 Modelo de qualidade; edição de 2000;
- IEEE 1008 IEEE *Standard for software unit testing*, edição de 2008;
- IEEE 829-2008 Plano de testes, edição de 2008.

Por exemplo a norma ISO 8402, define a noção de qualidade, a totalidade dos aspetos e características de um produto, processo ou serviço que se relacionam com a sua capacidade para satisfazer necessidades estabelecidas ou implícitas. Sendo que o produto é um software, a qualidade exigida é simplesmente o somatório dos aspetos e características que o cliente, deseja que possua.

Qualidade de software é um processo sistemático que focaliza todas as etapas e artefactos produzidos com o objetivo de garantir a conformidade de processos e produtos, prevenindo e eliminando defeitos (Bartié, 2002).

Por sua vez, a norma ISO/IEC 9126, define a avaliação de produtos de software a fim de satisfazer as necessidades de qualidade. Trata-se de um processo inerente ao ciclo de vida do software. Assim a norma define a qualidade interna e qualidade externa. A qualidade interna refere-se aos atributos que se aplicam aos componentes do software, detalhes do código a ser melhorado, revisão e teste. A qualidade externa, diz respeito aos requisitos não funcionais, referenciando os atributos como a fiabilidade, eficiência, usabilidade, manutenibilidade e portabilidade.

A qualidade do software poderá ser garantida através de sistemáticas aplicações de testes em cada fase de seu desenvolvimento. A execução de testes bem elaborados permite reduzir o nível de risco do sistema. Quando se detetam defeitos, a qualidade do software aumenta com a sua correção.

A qualidade é uma característica subjetiva, que está diretamente relacionada com as percepções de cada indivíduo (Mariani et al, 2011). No sentido de padronizar determinadas atividades, processos, dispositivos, etc, existem normas publicadas por organizações, onde existem descrições de processos e atividades, de modo a reunir as melhores práticas acerca do tema. Foram surgindo muitas organizações que pretendem criar padrões, normas, sendo as principais a ISO, (Organização Internacional para Padronização), IEC (International Electrotechnical Commission) é a Comissão Eletrotécnica Internacional, com a colaboração estreita com a comissão ISO e a IEEE, (Instituto de Engenheiros Eletricistas e Eletrônicos).

Relativamente ao desenvolvimento de software é importante seguir as normas de qualidade publicadas pelas organizações anteriormente referidas de modo a criar métodos rigorosos e uniformes. Não basta que exista qualidade de software, mas esta tem de ser reconhecida, por uma determinada entidade. Para que um software seja reconhecido tem de existir um processo de avaliação de acordo com uma determinada norma emitida pelas entidades.

3. QUALIDADE DE SOFTWARE EM APLICAÇÕES DE E-COMMERCE

3.1. Testes de Software em aplicações de e-commerce

As aplicações de *e-commerce* são suscetíveis a falhas tal como qualquer outro software. Por exemplo, as falhas podem surgir caso os protocolos subjacentes não sejam devidamente projetados e analisados.

O teste de software é um meio essencial e eficaz de criação de qualidade em sistemas de *e-commerce*. No entanto as aplicações de *e-commerce* quando comparadas com outros sistemas de software, têm uma arquitetura que as torna mais complexas de planear e testar. Tratam-se de aplicações baseadas num modelo cliente/servidor, com componentes distribuídos e executados em ambientes heterogéneos de hardware e software, e existindo também uma diversidade de browsers e respetivas versões. Os tipos de teste mais importantes neste contexto, são os testes funcionais, testes de desempenho, testes de segurança e testes de carga. Testar aplicações de *e-commerce* de forma a garantir qualidade ao utilizador/cliente é um desafio que requer habilidade. O mais importante ao testar, é certificar-se de que cada recurso está implementado corretamente mediante as exigências do cliente, de acordo com as normas de qualidade.

3.2. Qualidade via automatização de testes

A abordagem proposta neste artigo para a certificação de aplicações de *e-commerce* consiste na verificação da satisfação de normas de qualidade através da automatização de testes (Suresh Thummalapenta et al. 2012). Pretende-se executar os testes através da utilização de programas ou scripts que permitam testar a qualidade de sistema de forma automática. A necessidade de automatizar testes, deve-se essencialmente à necessidade de repetir testes, permitindo um aumento da qualidade. De acordo com a ISTQB a utilização da automatização de testes induz vários benefícios, nomeadamente permitir reduzir o trabalho repetitivo e potenciar a consistência e repetibilidade dos testes.

A implementação da presente abordagem para a automatização de teste deverá permitir a criação e execução de testes. Espera-se que a inclusão de processos automáticos nos testes permita detetar problemas e garantir uma maior qualidade das aplicações de *e-commerce*, permitindo a redução dos custos. Espera-se que a análise das normas e os respetivos requisitos de qualidade possam ser direcionados para aplicações de *e-commerce*. Como resultado, pretende-se definir um modelo automático de testes de aplicações de *e-commerce*, permitindo avaliar a qualidade das aplicações com base em normas de certificação.

4. CONCLUSÃO

O presente artigo apresenta um trabalho de pesquisa sobre testes de software, certificação de software, normas de qualidade, aplicações de *e-commerce* e respetivos critérios de qualidade. Foram apresentadas as principais normas e respetivos requisitos de qualidade, sendo objetivo do presente estudo direcioná-los para aplicações de *e-commerce*. O artigo apresenta uma proposta de abordagem para a certificação de aplicações de *e-commerce*. Como trabalho futuro, pretende-se definir um modelo automático de testes de aplicações de *e-commerce*, permitindo avaliar a qualidade das aplicações com base em normas de certificação. Espera-se que a inclusão de processos automáticos nos testes permita certificar as aplicações de *e-commerce* através da aplicação de várias normas de qualidade.

5. REFERÊNCIAS

- António Miguel, P. (2006). *Gestão de Projetos de Software*. FCA - Editora Informática, Ed.
- Bartié, A. (2002). *Garantia da qualidade de software*. CAMPUS - RJ, 2002.
- Crespo, A.N.; Silva, O. J.; Borges, C. A.; Salviano, C. F., Argolo, M. T. Jino, M. (2004). “Uma metodologia para Teste de Software no contexto da Melhoria de Processo”, III Simpósio Brasileiro de Qualidade de Software (SBQS), Brasília.
- Dasso, A., e Funes, A. (2007). *Verification, validation and testing in software engineering*. Idea Group.
- ISTQB (2011). *Programa de Certificação de Testador (Tester) de Nível Foundation*.
- Janice Reynolds (2004). *The Complete E-Commerce Book: Design, Build & Maintain a Successful Web-based Business Paperback*.
- Juniper Research (2016). “Mobile & Online Purchases: Cards, Carrier Billing and Third Party Payment Platforms 2015-2020”. <http://www.acepi.pt/artigoDetalhe.php?idArtigo=91721>. Consultado em 09/01/2016.
- Klaus Lochmann e Andreas Goeb. (2011). A unifying model for software quality. In *Proceedings of the 8th international workshop on Software quality (WoSQ '11)*. ACM, New York, NY, USA, 3-10. DOI=<http://dx.doi.org/10.1145/2024587.2024591>.
- Llewelyn, A.I. e Wickens, R.F., (1968). “The testing of computer software”, in P. Naur and B. Randell, eds., *Software Engineering, Report on a conference sponsored by the NATO SCIENCE COMMITTEE*, Garmisch, Germany, 7-11 October 1968. Scientific Affairs Division NATO, Brussels, Belgium, pp. 189-200.
- Mariani, L., Pezzè, M., Riganelli, O., e Santoro, M. (2011). AutoBlackTest: a tool for automatic black-box testing. 2011 33rd International Conference on Software Engineering (ICSE), 1013–1015. doi:10.1145/1985793.1985979
- Pressman, R. S. (2010). *Software Engineering A Practitioner’s Approach*. McGraw-Hill Education.
- Sirohi, N., Parashar, A. (2013). Component Based System and Testing Techniques, 2(6), 2378–2383.
- Suresh Thummalapenta, Saurabh Sinha, Nimit Singhania, e Satish Chandra (2012). Automating test automation. In *Proceedings of the 34th International Conference on Software Engineering (ICSE '12)*. IEEE Press, Piscataway, NJ, USA, 881-891.
- Tian, J. (2005). *Software Quality Engineering Testing, Quality Assurance and Quantifiable Improvement*.
- Zeeshan Ali Rana, Shafay Shamil, e Mian Muhammad Awais. (2008). Towards a generic model for software quality prediction. In *Proceedings of the 6th international workshop on Software quality (WoSQ '08)*. ACM, New York, NY, USA, 35-40.