Risk-Based Testing
(Track Introduction)

Michael Felderer[1], Marc-Florian Wendland[2], and Ina Schieferdecker[2]

[1] University of Innsbruck, Innsbruck, Austria
michael.felderer@uibk.ac.at
[2] Fraunhofer Institute FOKUS, Berlin, Germany
{marc-florian.wendland,ina.schieferdecker}@fokus.fraunhofer.de

1 Motivation and Goals

In many development projects, testing has to be done under severe pressure due to limited resources, a challenging time schedule, and the demand to guarantee security and safety of the released software system. Risk-based testing, which utilizes identified risks of a software system for testing purposes, has a high potential to improve testing in this context. It optimizes the allocation of resources and time, is a means for mitigating risks, helps to early identify critical areas, and provides decision support for the management [1, 2]. Risk-based testing is a type of software testing that explicitly considers risks of the software system as the guiding factor to solve decision problems in all phases of the test process, i.e., test planning, design, implementation, execution and evaluation [3–5]. It is based on the intuitive idea to focus testing activities on those areas that trigger the most critical situations for a software system [6]. The precise understanding of risks as well as their focused treatment by risk-based testing has become one of the cornerstones for critical decisions within complex software development projects and recently gained much attention [7]. Lately, the international standard ISO/IEC/IEEE 29119 Software Testing [8] on testing techniques, processes and documentation even explicitly considers risks as an integral part of the test planning process. As a result, several risk-based testing approaches (e.g., [9] or [10]) and empirical studies (e.g., [11] or [12]) have recently been provided to address increased practical need in this area, but further research is still inevitable.

This special track on risk-based testing serves as a platform for researchers and practitioners to present approaches, results, experiences and advances in risk-based testing. Its goal was to bring together researchers and practitioners working in the area of risk-based testing to discuss actual challenges and solutions to them. For this purpose, we invited leading researchers and practitioners to present their solutions to tackle actual challenges of risk-based testing. The invited format ensured broad coverage of this important topic. All contributed papers represent systematic rather than ad-hoc proposals which makes them interesting for a wide audience. Together, the papers in this track, which are summarized in the next section, provide a comprehensive and up-to-date overview of the communitys response to challenges of risk-based testing.

T. Margaria and B. Steffen (Eds.): ISoLA 2014, Part II, LNCS 8803, pp. 274–276, 2014.

2 Contributions

The special track comprises six contributed papers summarized in the following paragraphs.

Seehusen [13] presents a technique for risk-based test procedure identification, prioritization, and selection. The technique takes a risk model in the form of a risk graph as input, and produces a list of prioritized selected test procedures as output. The technique is generic as it can be used with many existing risk documentation languages and many kinds of likelihood and risk types. In the paper, the technique is demonstrated on the CORAS threat diagram language [14].

Felderer et al. [15] present a framework for integrating risk assessment, i.e., risk identification, analysis and evaluation, into an established test process. Their framework contains a risk assessment model which configures the test process. This model and its artifacts therefore determine the overall risk-based test process and are the main component of their risk assessment framework for testing purposes. The risk assessment model defines the test scope, the risk identification method, a risk model, and the tooling for risk assessment. It is derived on the basis of best practices extracted from published risk-based testing approaches and applied to an industrial test process.

Yahav et al. [16] address the quality risk of open source software components. For this purpose, Yahav et al. predict occurrence of bugs in these components using communication and community data, i.e., data on email communication traffic and social network dynamics on the basis of regression models. The information on predicted bugs is then intended to be used to allocate test efforts. The approach is illustrated with data from four open source projects.

Grossmann et al. [17] present an approach called Risk-Based Security Testing that combines risk analysis and risk-based test design activities based on formalized security test patterns. The involved security test patterns are formalized by using a minimal test design strategies language framework which is represented as a UML profile. Such a (semi-)formal security test pattern is then used as the input for a test generator accompanied by the test design model out of which the test cases are generated. The approach is based on the CORAS method [14] for risk analysis activities. Finally, a tool prototype is presented which shows how to combine the CORAS-based risk analysis with pattern-based test generation.

Botella et al. [18] describe an approach to security testing called Risk-Based Vulnarability Testing, which is guided by risk assessment and coverage to perform and automate vulnerability testing for web applications. Risk-Based Vulnerability testing adapts model-based testing techniques using a pattern-based approach for the generation of test cases according to previously identified risks and criticalities. For risk identification and analysis, the CORAS method [14] is utilized. The integration of information from risk analysis activities with the model-based test generation approach is realized by a test purpose language. It is used to formalize security test patterns in order to make them usable for test generators. Risk-Based Vulnerability Testing is applied to security testing of a web application.

References

1. Felderer, M., Haisjackl, C., Breu, R., Motz, J.: Integrating manual and automatic risk assessment for risk-based testing. In: Biffl, S., Winkler, D., Bergsmann, J. (eds.) SWQD 2012. LNBIP, vol. 94, pp. 159–180. Springer, Heidelberg (2012)
2. Felderer, M., Ramler, R.: Experiences and challenges of introducing risk-based testing in an industrial project. In: Winkler, D., Biffl, S., Bergsmann, J. (eds.) SWQD 2013. LNBIP, vol. 133, pp. 10–29. Springer, Heidelberg (2013)
3. Gerrard, P., Thompson, N.: Risk-based e-business testing. Artech House Publishers (2002)
4. Schieferdecker, I., Grossmann, J., Schneider, M.: Model-based security testing. In: Proceedings 7th Workshop on Model-Based Testing (2012)
5. Felderer, M., Ramler, R.: Integrating risk-based testing in industrial test processes. Software Quality Journal 22(3), 543–575 (2014)
6. Wendland, M.F., Kranz, M., Schieferdecker, I.: A systematic approach to risk-based testing using risk-annotated requirements models. In: ICSEA 2012, The Seventh International Conference on Software Engineering Advances, pp. 636–642 (2012)
7. Felderer, M., Schieferdecker, I.: A taxonomy of risk-based testing. STTT (2014), doi:10.1007/s10009-014-0332-3
8. ISO: ISO/IEC/IEEE 29119 Software Testing (2013), `http://softwaretestingstandard.org/` (accessed: August 12, 2014)
9. Neubauer, J., Windmüller, S., Steffen, B.: Risk-based testing via active continuous quality control. STTT (2014), doi:10.1007/s10009-014-0321-6
10. Carrozza, G., Pietrantuono, R., Russo, S.: Dynamic test planning: a study into an industrial context. STTT (2014), doi:10.1007/s10009-014-0319-0
11. Felderer, M., Ramler, R.: A multiple case study on risk-based testing in industry. STTT (2014), doi:10.1007/s10009-014-0328-z
12. Erdogan, G., Li, Y., Runde, R.K., Seehusen, F., Stølen, K.: Approaches for the combined use of risk analysis and testing: A systematic literature review. STTT (2014), doi:10.1007/s10009-014-0330-5
13. Seehusen, F.: A technique for risk-based test procedure identification, prioritization and selection. In: Margaria, T., Steffen, B. (eds.) ISoLA 2014, Part II. LNCS, vol. 8803, pp. 277–291. Springer, Heidelberg (2014)
14. Lund, M.S., Solhaug, B., Stolen, K.: Model-driven Risk Analysis. Springer (2011)
15. Felderer, M., Haisjackl, C., Pekar, V., Breu, R.: A risk assessment framework for software testing. In: Margaria, T., Steffen, B. (eds.) ISoLA 2014, Part II. LNCS, vol. 8803, pp. 292–308. Springer, Heidelberg (2014)
16. Yahav, I., Kenett, R.S., Bai, X.: Data driven testing of open source software. In: Margaria, T., Steffen, B. (eds.) ISoLA 2014, Part II. LNCS, vol. 8803, pp. 309–321. Springer, Heidelberg (2014)
17. Großmann, J., Schneider, M., Viehmann, J., Wendland, M.-F.: Combining risk analysis and security testing. In: Margaria, T., Steffen, B. (eds.) ISoLA 2014, Part II. LNCS, vol. 8803, pp. 322–336. Springer, Heidelberg (2014)
18. Botella, J., Legeard, B., Peureux, F., Vernotte, A.: Risk-based vulnerability testing using security test patterns. In: Margaria, T., Steffen, B. (eds.) ISoLA 2014, Part II. LNCS, vol. 8803, pp. 337–352. Springer, Heidelberg (2014)