

# Exploring Expectations About Risk-Based Testing: Towards Increasing Effectiveness and Efficiency

Michael Felderer<sup>1</sup> and Rudolf Ramler<sup>2</sup>

<sup>1</sup> Institute of Computer Science, University of Innsbruck, Technikerstrasse 21a,  
6020 Innsbruck, Austria

michael.felderer@uibk.ac.at

<sup>2</sup> Software Competence Center Hagenberg GmbH, Softwarepark 21, 4232 Hagenberg, Austria  
rudolf.ramler@scch.at

**Abstract.** Risk-based testing is sometimes reduced to an approach that focuses on cutting costs and time in testing. While the high effort involved in testing makes efficiency an important issue, for many companies the main concern is still to find the critical defects in their software products. Such defects can cause costly remedial upgrades and fixes, and they threaten the company's long-term business success. In this paper we explore how the two goals "effectiveness" and "efficiency" motivate a risk-based testing approach in different organizations. Furthermore, we identify a third goal, summarized as "management support". In a survey conducted as part of a tutorial on risk-based testing we investigated common expectations and potential benefits associated with these three goals. The results indicate that the main motivation for a risk-based approach is making testing more efficient. Nevertheless, efficiency and effectiveness are not conflicting goals and the main challenge is therefore finding strategies that increase the overall benefit of including risk information in testing.

**Keywords:** Test management · Software risk management · Software testing · Risk-based testing · Test process improvement · Effectiveness · Efficiency

## 1 Introduction

Software testing is an essential and widely practiced measure for assuring software quality. By accounting for up to 50 % of the overall project effort, testing is also a highly costly and time-intensive activity in software development. Hence, an adequate test strategy plays a key role in balancing product quality with cost and time-to-market. Ideally this balance is achieved by taking the risks into account, which are associated with the consequences of poor quality caused by software defects. A systematic consideration of the involved risks is suggested by risk-based testing [1].

However, risk-based testing is sometimes (mis-)understood as an approach that focuses primarily on minimizing costs and time – according to the pun "cut testing and take the risk". The high effort involved in software testing makes efficiency an important management goal. Although important, for many companies the main concern is still to miss critical bugs. Critical bugs can significantly delay time-to-market, result in costly

hotfixes, and threaten the acceptance of software products and services by customers on the long run. In such a case, effectiveness is often the primary goal in software testing. It has to be balanced with short-term economic constraints such as limited time and or resources available for software testing.

The need to balance effectiveness and efficiency requires adequate management support. Providing management supports is therefore a further, commonly observed goal of risk-based testing. It is also found in contexts where testing is determined by the need to fulfill industry standards and organizational regulations.

In this paper we explore these main goals associated with risk-based software testing. Section 2 provides an overview of the relevant concepts and links them to findings from related work. Section 3 shows the results from a survey conducted as part of a previous tutorial held at the Software Quality Days 2016<sup>1</sup>. The paper is concluded by a summary and discussion in Sect. 4.

## 2 Background

*Risk-based testing* is a testing approach which considers risks of the software product as the guiding factor to support decisions in all phases of the test process [2]. In previous studies we investigated the potential of risk-based testing in large enterprises [3] as well as in small and medium enterprises [4]. We also studied the introduction of risk-based testing in an organization [5], where an essential first step is to establish a risk-based test strategy with clear goals and expectation for all testing activities. Among other research questions we explored “What is the benefit and improvement potential that defect prediction can provide for software testing?” in these different contexts. Our findings can be summarized as follows.

The general motivation is that information about fault-prone modules, i.e., those that have a high risk of causing critical failures, allows focusing the testing effort on selected parts of the software system instead of testing the entire system with the same rigor (Fig. 1). The associated improvement potential is based on the observation that the majority of the faults (usually approximated as “80 % of the defects”) comes from a relatively small amount of the code (usually approximated as “20 % of the modules”) and that “about half the modules are defect free” [6, 7].

### 2.1 Effectiveness

One of the resulting benefits of risk-based testing is its ability to *increase the effectiveness of testing*, which can be defined as the degree to which testing is able to detect all defects in the system under test (e.g., defects actually detected per total defects). Testing is an investment in the quality of a software product [8, 9]. Even though resources are limited in general, it is important to achieve quality requirements such as functional correctness, reliability, performance etc.

---

<sup>1</sup> <https://2016.software-quality-days.com/en/>.

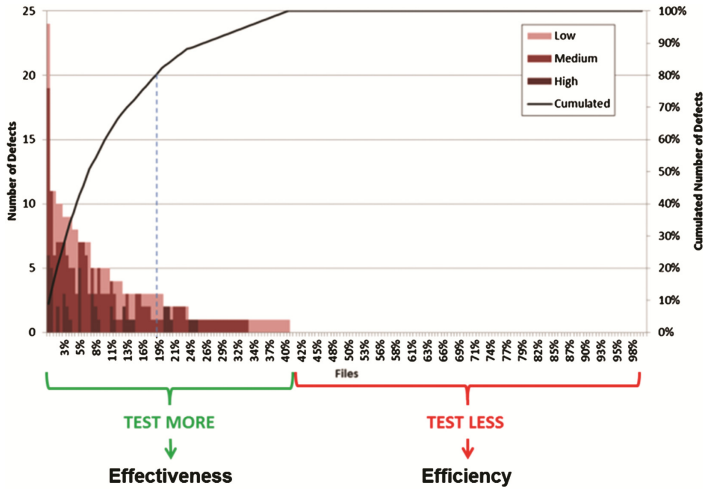


Fig. 1. Increasing effectiveness and efficiency with risk-based software testing.

A risk-based approach can help to make testing more effective by including information about high-risk components in test planning from the very beginning. Directing the main testing effort to high-risk components to pursue a thorough, systematic testing approach (1) *increases the likelihood that the existing defects are detected*. Besides an improvement of the product's quality, (2) setting priorities based on risk considerations fosters the *detection of critical defects* first. Risk-based prioritization also ensures that (3) defects are found *in the early iterations of testing*.

In our study on risk-based testing in industry [3] we found that the investigated companies use risk information to further increase the range of testing with additional risk-based test cases. These test cases are added to the existing set of tests that were systematically derived from the requirements in order to boost the chance of detecting additional defects. It is expected that fewer defects will slip through to the field.

## 2.2 Efficiency

Another potential benefit of risk-based testing is its ability to *increase the efficiency of testing*, which can be defined as ratio between output and input (e.g., detected defects per time spent). Time and manpower are constrained resources in the development of software systems. This also affects the resource allocation in software testing, which has to cope with the fundamental challenge that exhaustive testing of a software system is not possible in a realistic timeframe [10]. Therefore, it is sensible to allocate (prioritize) the available resources in a way so that most output can be generated.

Most organizations use testing as a means to reduce the risk of delivering software with critical bugs. However, a harsh competitive environment can imply severe time and resource constraints that outweigh quality risks. In the struggle to meet strict release deadlines or to maintain short-term business opportunities, companies may be forced to (1) *reduce testing time* and (2) *cutback resources* allocated for testing. In this context,

the understanding of risks is typically used to make testing more efficient, i.e., to adjust the amount of testing to optimally use the available time and resources for covering at least the most critical parts.

Small and medium enterprises (SME) seem to be most affected by time and resource constraints. In our study on risk orientation in software testing processes of SME [4] we observed cases where risk information has been used to increase test efficiency, i.e., to adjust the amount of testing to reduce cost and time. Larger organizations seem to tackle time and resource constraints by using risk information to increase the chance of finding critical defects in the early iterations of testing and, in consequence, to reduce the overall costs and time required for stabilization [3].

### 2.3 Management Support

Finally, we also found that companies benefit from the risk information used in risk-based testing for supporting management in decision making and in pursuing process improvement initiatives. A risk-based approach helps to make the balance between cost and quality transparent and easier to communicate. These benefits were observable in small, agile organizations [4] as well as in large enterprises [3] that have to cope with the fulfillment of industry standards and organizational regulations.

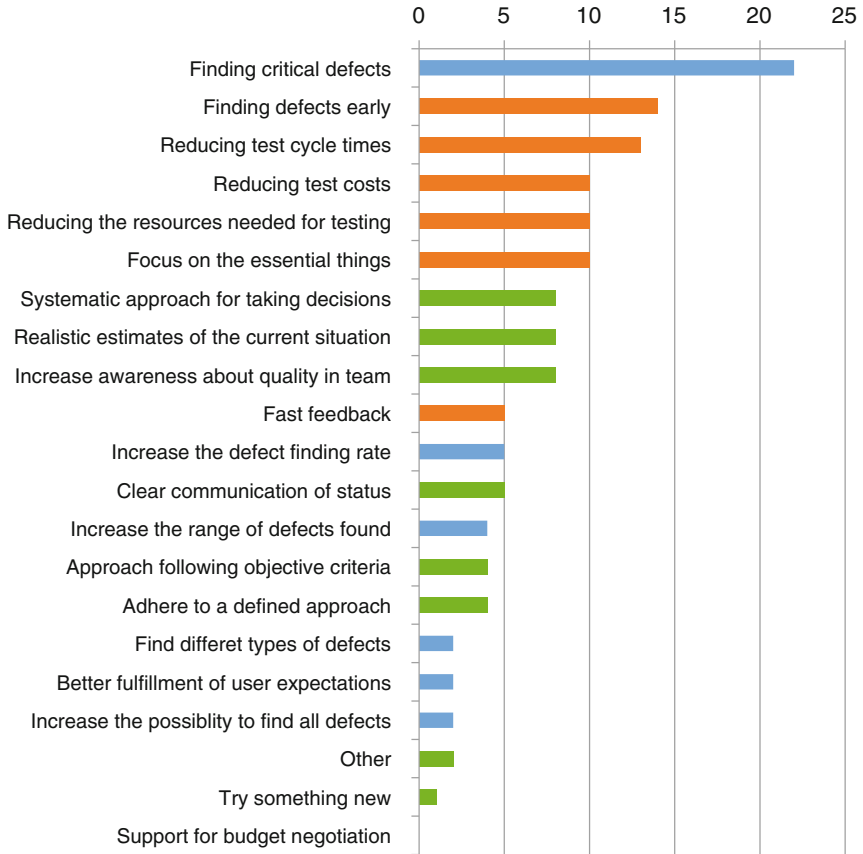
One example is the use of risk-burndown charts [5]. Traceability between the executed test cases, the test results and the risk items enables reliable release quality statements as well as the estimation of residual risks. In risk-based evaluation and reporting, risk burn-down charts, which illustrate the development of risk exposure for a system or specific artifacts over time, are a suitable measure for release quality and residual risk estimation. The risk that is estimated in the beginning is reduced with every test cycle until a risk level is reached where it is acceptable to release the software application. This method based on risk burn-down charts proposed in the studied project has been found intuitively accessible and suitable for supporting release decisions by project management.

## 3 Preliminary Survey Results

A list of commonly observed expectations and potential benefits that motivate the adoption of a risk-based testing approach has been collected from previous projects (e.g. [5]) as well as related literature (e.g., [11, 12]). In a survey conducted as part of a tutorial held at the Software Quality Days 2016, one of the biggest industry-academia conventions on software quality in Europe, we asked the participants about their motivation for risk-based testing in their projects or products.

The survey showed 20 different motivations for risk-based testing. Each participant was asked to select all motivations considered relevant in context of his or her work. The participants' main roles were software tester, test manager, project manager, quality manager, team lead or head of QA, software architect and developer with practical experience in software testing ranging from 1 to more than 10 years; majority in the range of 3 to 5 years. We received answers from 23 participants ( $N = 23$ ).

Figure 2 shows the results sorted by the number of times a motivation has been selected by a survey participant. Motivations related to *effectiveness* are shown in blue, motivations related to *efficiency* are shown in orange, and motivations related to organizational and *management support* are shown in green color.



**Fig. 2.** Number of times a motivation has been selected by a survey participants ( $N = 23$ ). Motivations are related to effectiveness (blue), efficiency (orange) or management support (green). (Color figure online)

## 4 Conclusions and Future Work

Despite its preliminary character, the survey indicates that the main motivation for introducing or conducting risk-based testing is making testing more efficient. Weighted over all possible motivations selected by the participants, 48 % can be related to *efficiency* aspects, 29 % to *effectivity*, and 23 % to *management support*. The emphasis on efficiency is also visible from the ranking of the motivating factors. It is, however, worth noting that the top most motivation selected by 22 out of 23 participants is “finding critical

defects”, which is remarkable since we consider this motivation to be associated more with effectiveness than with efficiency.

Efficiency and effectiveness are not conflicting objectives in risk-based testing. From a research perspective, thus, the main question for future work is to explore which strategies that maximize the overall benefit of testing with risk information.

**Acknowledgements.** This work has been supported by the COMET Competence Center program of the Austrian Research Promotion Agency (FFG), and the project QE LaB – Living Models for Open Systems funded by the Austrian Federal Ministry of Science, Research and Economy.

## References

1. Ramler, R., Felderer, M.: A process for risk-based test strategy development and its industrial evaluation. In: Abrahamsson, P., Corral, L., Oivo, M., Russo, B. (eds.) PROFES 2015. LNCS, vol. 9459, pp. 355–371. Springer, Heidelberg (2015). doi:[10.1007/978-3-319-26844-6\\_26](https://doi.org/10.1007/978-3-319-26844-6_26)
2. Felderer, M., Schieferdecker, I.: A taxonomy of risk-based testing. *Int. J. Softw. Tools Technol. Transf.* **16**(5), 559–568 (2014). Springer
3. Felderer, M., Ramler, R.: A multiple case study on risk-based testing in industry. *Int. J. Softw. Tools Technol. Transf.* **16**(5), 609–625 (2014). Springer
4. Felderer, M., Ramler, R.: Risk orientation in software testing processes of small and medium enterprises: an exploratory and comparative study. *Softw. Q. J.* **24**(3), 519–548 (2016). Springer
5. Felderer, M., Ramler, R.: Integrating risk-based testing in industrial test processes. *Softw. Q. J.* **22**(3), 543–575 (2014). Springer
6. Boehm, B., Basili, V.R.: Software defect reduction top 10 list. *IEEE Comput.* **34**(1), 135–137 (2001). IEEE
7. Shull, F., Basili, V.R., Boehm, B., Brown, A.W., Costa, P., Lindvall, M., Port, D., Rus, I., Tesoriero, R., Zelkowitz, M.: What we have learned about fighting defects. In: 8th Symposium on Software Metrics, METRICS 2002, p. 249. IEEE (2002)
8. Huang, L.G., Boehm, B.: How much software quality investment is enough: a value-based approach. *IEEE Softw.* **23**(5), 88–95 (2006)
9. Ramler, R., Biffi, S., Grünbacher, P.: Value-based management of software testing. In: Value-Based Software Engineering, pp. 225–244. Springer, Heidelberg (2006)
10. Ramler, R., Wolfmaier, K.: Economic perspectives in test automation: balancing automated and manual testing with opportunity cost. In: International Workshop on Automation of Software Test, AST 2006, pp. 85–91. ACM (2006)
11. Amland, S.: Risk-based testing: risk analysis fundamentals and metrics for software testing including a financial application case study. *J. Syst. Softw.* **53**(3), 287–295 (2000)
12. Redmill, F.: Theory and practice of risk-based testing. *Softw. Test. Verification Reliab.* **15**(1), 3–20 (2005). Wiley