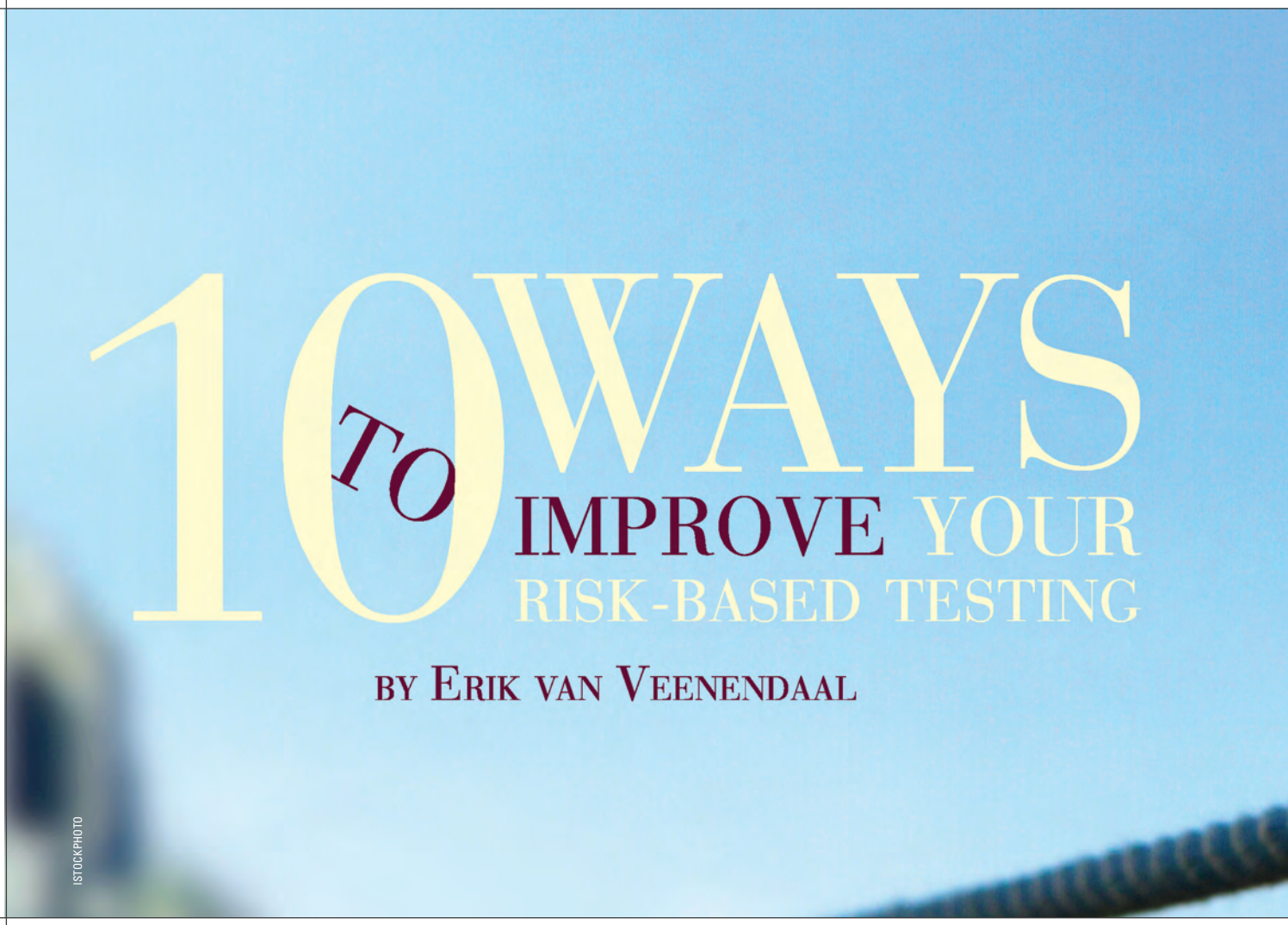

10 WAYS TO IMPROVE YOUR RISK-BASED TESTING

BY ERIK VAN VEENENDAAL

ISTOCKPHOTO

Risk-based testing is the foundation of almost every testing activity, as we strive to strike a balance between product quality and tight deadlines. Whether you are involved in a traditional waterfall or V-model environment or you're applying agile development methodologies, establishing testing priorities is always a challenge. Of course, risk-based testing should be driven by business objectives. Testers do not determine business risk—the products' stakeholders do. It is our responsibility to provide information on product and project risk status to enable stakeholders to make better decisions.

These ten essential ideas will improve your risk-based testing.

START RISK-ANALYSIS BY IDENTIFYING THE FULL SET OF STAKEHOLDERS

Since stakeholders provide the essential information for the identification and analysis of risks, having the right set of stakeholders is essential. Stakeholders include those who are responsible for the new system (e.g., product owner, project manager), those who are affected when the product doesn't function correctly (e.g., maintenance, help desk), and those who use the system in their work (e.g., end-users, management). In Utopia, thorough stakeholder identification has already taken place during the requirements phase. In our world, stakeholder identification may need to be done as part of the risk-analysis process. Remember, a forgotten stakeholder means forgotten product risks.

STATE THE PRODUCT RISKS IN THE LANGUAGE OF THE BUSINESS

Communication is vital to a successful project. Product risks should be stated in such a way that they are understood by the business stakeholders. It should be clear to them what it means if a risk becomes reality. Product risks will only receive the full attention of the stakeholders when they understand all the consequences. Often, when too few business people are involved in a product-risk analysis, the result is a list of risks that has no meaning to them. Specify product risks explicitly in terms of both the failure and the resulting negative impact.

RECOGNIZE THAT IMPACT AND LIKELIHOOD ARE DIFFERENT

Some risk analysis techniques calculate the level of risk by multiplying impact by likelihood to create a risk level. An extremely high impact risk (e.g., safety) with a low likelihood may then receive too little attention. Consider table 1, where impact and likelihood are both rated on a scale from 1 to 5.

	Impact	Likelihood	Risk level
Product Risk A	5	2	10
Product Risk B	2	5	10

Table 1

Although the risk level for A and B is the same, the nature of risk A and risk B is very different. Just looking at the risk level (10) and sorting the product risk list by risk level is very dangerous. Vital information is lost using this approach. Impact usually relates to business factors and business risks; likelihood relates to technical factors and technical risks. These types of risks are by nature very different and should have different mitigation approaches.

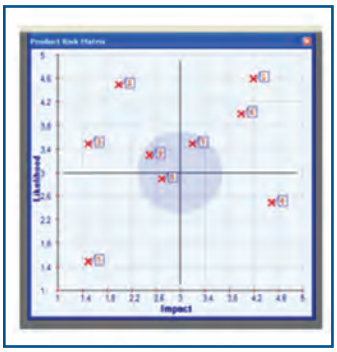

Figure 1

Visualize the Results of the Product-Risk Analysis

A picture is often worth more than a thousand words. Presenting risk-assessment results in a diagram is usually much more effective than in tabular form with many numbers. The table becomes indecipherable very quickly, and often stakeholders lose themselves in a number-based discussion. Presenting the results of a risk analysis in a matrix format, as shown in figure 1—where impact is on the horizontal axis, likelihood is on the vertical axis, and the four squares each represent a level and type of risk—generally provides a much better basis for discussing the product risks [1].

Consider Both Functional and Nonfunctional Risks

Some requirements documents omit nonfunctional requirements. When nonfunctional quality attributes such as performance, reliability, and usability are not well specified, the product built from those requirements may not satisfy stakeholder needs. When nonfunctional risks are forgotten, additional problems can arise. Standard checklists for nonfunctional risks are available [2, 3], but don't go overboard and lose yourself in a detailed list of nonfunctional quality attributes that no one understands. Focus on a small set of nonfunctional attributes that are important and that you are able to test.

Define a Differentiated, Risk-Based Test Approach

Product risks that are more critical than others should be tested differently, with greater coverage and stricter exit criteria. This could involve using different test-design techniques, performing reviews, doing additional regression testing, and having users perform tests that utilize their knowledge. This differentiated, risk-based test approach should be clearly defined early in the testing process to allow for effective allocation of test resources. If such an approach is not explicitly taken and it is up to the individual tester to

decide how much testing is enough to cover a product risk, the whole structure of risk-based testing falls apart.

Report against the identified product risks

Even with an effective product-risk-analysis process, many test reports remain defect based. Test reports should be organized around the product risks that have been identified. This allows stakeholders to make an informed decision regarding product release. In practice, detailed defect-based reports are often not meaningful to business stakeholders. A short, focused test report in a stakeholder's language provides real value. This report should define the product risks, the planned test coverage, the actual test coverage, and the risks mitigated.

Choose the product-risk-analysis method that meets your needs

Many methods of product-risk analysis are extremely thorough but take a substantial amount of time. This may be appropriate when testing a safety-critical system. In an agile context, risk-based testing is still important and may be more important due to strict delivery deadlines. However, the product-risk analysis should be lightweight and very focused. A simple brainstorm session at the beginning of an increment may suffice. In general, don't make the analysis more complex than necessary. When defining a process for product-risk analysis or choosing a method, keep in mind why you are doing it and what you are trying to achieve. The result that you are trying to achieve—not the process—should be leading.

Revisit product risks on a regular basis

Many testers perform one product-risk analysis at the beginning of a project and then act as though risks don't change throughout the project. Remember, the product-risk identification and analysis are based on stakeholders' perceptions and expectations. These will almost always change over time. Early testing will reveal some new risks while mitigating others. Evolving requirements usually means evolving product risks. It pays to revisit the risk analysis on a periodic basis, at least at every major milestone. This will make testing more effective by addressing the newly identified product risks and more efficient by not wasting precious time testing for risks that have become less important.

Establish clear risk ownership and responsibilities

In many organizations, testers identify and analyze risks. This is wrong. Testers are not the owners of risk. Our responsibility is to facilitate the risk-analysis process and inform our stakeholders of the status of product risks. When stakeholders are asked to identify product risks and indicate the level of testing to be performed, they become aware that they are the deciding factor. If the stakeholders miss a product risk, it is their—not the tester's—responsibility. It is important to keep this in mind when faced with a stakeholder's resistance to becoming the risk owner.

Doing a thorough product-risk analysis is easier said than done. Some text books make product-risk analysis much too theoretical and difficult, others do not really address the issue and only skim the surface. These ten practices should make you aware of some of the most critical issues that need to be dealt with in the real world when doing a product-risk analysis. Remember, without product-risk analysis, a tester may develop great test cases—e.g., using formal test design techniques—but the test cases may well be directed toward the less critical test items and features. **{end}**

eve@improveqs.nl

Sticky Notes

For more on the following topic go to www.StickyMinds.com/bettersoftware.

- References