

Erik Van Veenendaal - Founder - Improve Quality Services Ltd.

June 2012

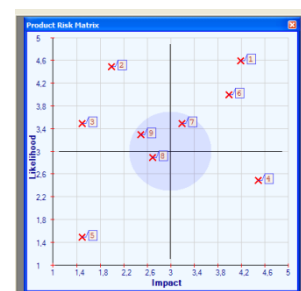


10 things to remember about risk-based testing

Most projects apply some kind of (implicit) risk-based testing. We all have to balance between product quality and tight deadlines. Risk-based testing is the basis of almost every testing activity. Of course risk-based testing should be driven by business objectives. Testing is not the risk owner, but the products' stakeholders are. It is our job to inform the stakeholders about risk-based decisions and provide visibility on product risk status. Risk-based testing starts by doing risk identification and analysis in close co-operation with stakeholders. It also addressed the mitigation approach regarding the identified product risks.

From many practical experiences in various domains, Erik, author of the latest book on risk based testing "Practical Risk-Based Testing: The PRISMA Approach, shares 10 essential lessons learned regarding risk-based testing; 10 things to remember.

1. **Start risk-analysis by doing a proper stakeholder analysis.** Since stakeholders provide the essential information for the identification and analysis of risks, having the right set of stakeholders is essential. In Utopia a thorough stakeholder analysis has already taken place during requirements phase. Both stakeholders from a business perspective and from a technical perspective (e.g. architect, lead engineer) are required. Remember, a forgotten stakeholder implies forgotten risks.
2. **State the product risks in a business language.** Communication is vital to a successful project. Product risks should be stated in such a way that they are understood by the business. It should be clear to them what it means if a risk becomes apparent. Only product risks where all understand what the impact is, in case of a failure, will get focus in communication
3. **Recognize that impact and likelihood are different.** Some product risks analysis techniques calculate the level of risk by multiplying impact by likelihood and from then on just the resulting calculated risk level is used. An extremely high impact risk (e.g. safety) with a low likelihood may then not get any or too little attention. 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 also have different mitigation approach.
4. **Visualize the results of the product risk analysis.** A picture is worth more than a thousand words. Presenting the results in a diagram is usually much more clear to all stakeholders than a table (excel sheet) with many numbers. The latter becomes unreadable very easily and



PRISMA Risk Matrix

some loose themselves in a number based discussion.

5. **Consider both functional and non-functional risks.** Like with requirements specification some “forget” the non-functional product risks. However, in practice the non-functional quality attributes, such as performance, reliability and usability, often make the difference. Beware not to go overboard and lose yourself in long and detailed non-functional list of attributes that nobody really understands. Only discuss a limited set of non-functional attributes that could be off importance, and that you are capable of testing using test design techniques.
6. **Define a differentiated risk-based test approach.** Product risks that are more critical than others should be tested differently, with more coverage, stricter exit criteria etc. A tester testing an item related to critical product risk should act differently than testing a less critical item. This differentiated risk-based test approach should be clearly defined upfront to allow for effective usage of test resources.
7. **Report against the identified product risks.** Many do a product risk analysis but test reports are again defect based. Stakeholders tell us which product risks are important and should be mitigated before being to release the system. A test report should provide this information and support the release decision. In practice, defect based reports are often not the most usable for business stakeholders. It is recommended to define product risk coverage and product risks mitigated as completion criteria.
8. **Choose the product risk analysis method that meets your needs.** Many methods on product risk analyses are not light weight and extremely thorough. This may fit when doing testing in a V-model environment for a safety critical system. When doing testing in an agile context it is still important to make choices. However, the product-risks analysis should be light weight and very focused. A simple brainstorm may suffice at the beginning of an increment. In general, don't make it more difficult than necessary.
9. **Revisit the product risks analysis on a regular basis.** The product risk identification and analysis are based on stakeholders' perceptions and expectations. These will change over time. Early testing will reveal some new risks while mitigating others. Changing requirements usually means changing product risks. It pays to revisit the product risk analysis on a periodic basis, at least at every project milestone.
10. **Establish clear risk ownership and responsibilities.** In many organizations testers' identify and analyze the risks. This is wrong; testers are not the risk owners!! Our responsibility is 'only' to facilitate the risk analysis process and inform our stakeholders on the status of the product risks. When stakeholders are asked to identify product risks and thereby indicate what to test and what no to test, they suddenly become aware that they are the deciding factor. If they do it wrong (e.g. miss a product risk), they are to blame and not the tester. This often leads to stakeholders' resistance: „It was so easy when the tester took the decision for us.”

References

E. van Veenendaal (2012), Practical Risk-Based Testing: The PRISMA Approach, UTN Publishers, ISBN 9-789490-986070 (www.utn.nl)

For more information:

Erik van Veenendaal (www.erikvanveenendaal.nl) is a leading international consultant and trainer, and a widely recognized expert in the area of software testing and quality management with over 20 years of practical testing experiences. He is the founder of Improve Quality Services Ltd. (www.improveqs.nl). In 2007 he received the European Testing Excellence Award for his contribution to the testing profession over the years. He has been working as a test manager and consultant in various domains for more than 20 years. He has written numerous papers and a number of books, including “The Little TMMi”, “ISTQB Foundations of Software Testing” and “Testing according to TMap”. Erik is also a former part-time senior lecturer at the Eindhoven University of Technology, vice-president of the International Software Testing Qualifications Board (2005–2009) and currently vice chair of the TMMi Foundation.

*Erik Van Veenendaal – Founder – Improve Quality Services Ltd.
eve@improveqs.nl
<http://www.erikvanveenedaal.nl>*