



Erik van Veenendaal

# PRISMA: Product Risk Assessment for Agile projects

## Column

*Risk assessment and management is the backbone of sequential development models but how does it fit in agile environments? How can we be sure to identify new risks when they emerge and to ensure our understanding of all risks remains accurate? In agile much emphasis is on communication. Perfect for development issues where mistakes can be discussed and fixed. But product risk are not by nature iterative: they is absolute and exists all the time and making mistakes in dealing with them may not acceptable. Hence the discussion and consensus approach needs to be slightly formalized by the use of a systematic method and process. That's where PRISMA comes in.*

### PRISMA

PRISMA (Product RISK Management) is an approach for identifying the areas that are most important to test, i.e., identifying the areas that have the highest level of business and/or technical risk. The PRISMA method has been bottom-up developed by Improve Quality Services in practice over a large number of years. PRISMA has been proven to be successful in supporting (test) organizations as they apply risk-based testing. Today, it is taught at several universities to IT students. The PRISMA approach especially supports the test professional in performing product risk identification and product risk analysis as well as in working in close co-operation with stakeholders.

### Product Risk Matrix

The central theme in the PRISMA process is the creation of the so-called product risk matrix (see figure 1). For each product risk identified, the impact of possible defects and the likelihood of these defects occurring is determined. By assigning numeric values to both impact and likelihood, a product risk (test item) can be positioned in the product risk matrix. The standard risk matrix is divided in four areas each representing a different level and type of risk. A different level and/or type of risk should also imply a different test approach, to be documented in a (master) test plan. The product risk matrix can thus used as a basis for all testing performed in a project.



Figure 1. PRISMA product risk matrix

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 in a PRISMA product risk where impact is on the horizontal axis, likelihood is on the vertical axis, and the four quadrants each represent a level and type of risk – generally provides a much better basis for discussing and validating the product risks.

## Agile

Since risk mitigation is one of main objectives of Agile, an approach such as PRISMA can fit into an Agile development project perfectly. In practice PRISMA has proven to be a relatively light weight approach (unlike some), focused on producing tangible results, e.g., the product risk matrix and a differentiated risk-based test approach. Most often when organizations come from a more traditional environment using a structured testing approach such as TMap many testing practices are removed from day-to-day practice. One of the testing practices that is still necessary is a product risk assessment which determines where and how to focus the limited test resources to effectively meet the project deadlines. Where some methods use very detailed approaches for product risk assessment, PRISMA is generally considered relatively light weight and result-driven. In fact, from personal experience, most projects that convert to Agile software development keep PRISMA as one of their core testing practices. Note that in Agile the team is explicitly responsible for the quality of the product.

## The risk assessment process

How is PRISMA applied in Agile software development? Product risks are derived from documents (i.e., the list of backlog items assigned to the next sprint and user stories) and are typically identified in a brainstorm session(s). Of course the approach largely depends on the Agile approach that is being used and the cycle time. Based on personal experience, longer sprints of four weeks or one month are most common. The sprint team is often also the PRISMA team performing the product risk assessment. “External” stakeholders are contacted and asked for their input or actively participate in the process. It is usually carried out as a focused meeting, where the team runs through the PRISMA process as described below. At

the end of the meeting the team agrees on the product risk matrix and thus the focus of testing.

## Risk poker

Having the list of product risk, they are now scored (separately for likelihood and impact) using the essentials of the planning poker technique as often practiced in agile projects. Planning Poker is a consensus-based technique for estimating. It is a variation of the Wideband Delphi method. The PRISMA risk poker is uses the list of product risks (user stories) to be tested and several copies of a deck of cards. The decks have numbered cards and often use the sequence: 0, 1/2, 1, 2, 3, 5, 8, 13, 20, 40, 100, and optionally a “?” (unsure) and a coffee cup (I need a break). A common variation is not using a deck with numbers but colored cards, e.g., dark green, light green, yellow, orange and red, relating back to the “1 to 5” value set. This is practiced since the meaning of the numbers from the deck often lead to much discussion, and are ambiguous in the PRISMA context, when using them to estimate likelihood and impact.

Each team member receives a deck of cards with varying values (or colors). After a short explanation of the product risk item (user story), the moderator (e.g., a SCRUM Master) calls for an estimate for either likelihood or impact. After a few seconds of contemplation, each team member selects a card, without showing it to the other team members, and at a set time, all show their selected cards. It is important that all cards are shown at once, to prevent ‘peer pressure’ towards a lower or higher number (or color). If the numbers (or colors) are essentially the same, the moderator writes down the median value. If they differ wildly, the lowest estimator and highest estimator briefly explain their choice essentially going back to the PRISMA factors for likelihood and impact. Often then agreement is achieved for a number (or color) based on that discussion. If no agreement is reached, the moderator, business owner (for impact) or lead developer (for likelihood) act as a tie breaker and chooses a number (or color) from within the range. It is important to move quickly to the next product risk item. Optionally, an egg timer can be used to limit time spent in discussion of each item.

One common variation is providing each team member with a limited number of each value or color, and having them ‘use up’ each value card in the process. This prevents the tendency of some people to stick to very high or very low scores for all product risks.

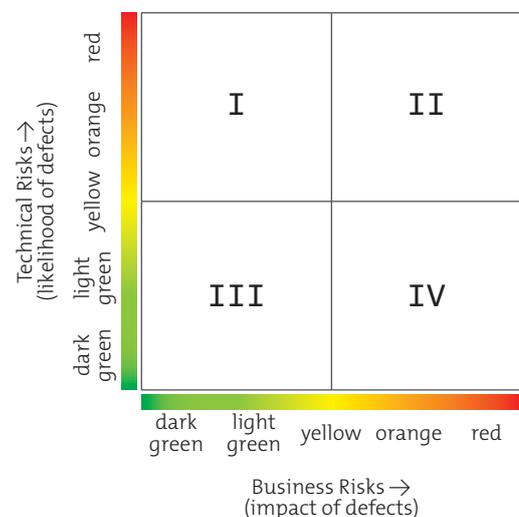


Figure 2. Planning poker Product risk matrix

Erik van Veenendaal  
Brian Wells



## Test Maturity Model integration TMMi®

Guidelines for Test Process Improvement



## NEW PUBLICATION

**Erik van Veenendaal and Brian Wells**

### Test Maturity Model integration TMMi – Guidelines for Test Process Improvement

TMMi is a not-for-profit independent test maturity model developed by the TMMi Foundation. The most important differences between TMMi and other test improvement models are independence, compliance with international testing standards, the business-driven (objective-driven) orientation and the complementary relationship with the CMMI framework.

**This book provides:**

- a comprehensive overview of the TMMi model
- the TMMi specific goals and specific practices
- many examples
- detailed insight into the relationship between TMMi and CMMI.

ISBN 978-94-90986-10-0  
pages: 352, price € 39.90  
Order at [www.utm.nl](http://www.utm.nl)



At the end of the session when all items have an assigned number (or color) for impact and likelihood, they are positioned in the product risk matrix. The axes of the product risk matrix then usually have a scale that reflects the numbers (or colors) on the deck (see figure 2). The end result of the session (the product risk matrix) is validated by the team to check whether there are no items that are positioned such that they need re-discussion.

### One page Test plan

In many Agile projects the product risk matrix including a defined differentiated approach are used as the test plan for the next sprint. By putting a picture (as shown in figures 1 and 2) on the wall, everyone can see the test actions to be performed. This picture is often enhanced by providing the Definition of Done criteria per quadrant. The two to three hour dedicated PRISMA product risk session delivers the sprint test plan in an easily readable format on one page. How much more efficient and effective can one become!?

### > about the author



**Erik van Veenendaal** ([www.erikvanveenendaal.nl](http://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 experience. He is the founder of Improve Quality Services BV ([www.improveqs.nl](http://www.improveqs.nl)). He holds the EuroSTAR record, winning the best tutorial award three times! 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 Testing Practitioner", "ISTQB Foundations of Software Testing" and "The Little TMMi". 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.