



# Serious Games for Testers

## Introduction

The use of ATDD and tools like FitNesse, Cucumber, and Robot Framework makes it necessary to create automated acceptance tests. These acceptance tests are a natural extension of the acceptance criteria used in user stories. You use acceptance tests to understand what needs to be developed so that you can develop the correct functionality. In order to develop the correct functionality, you need to create a common understanding of stakeholder value among all team members. You use story workshops (product backlog refinement meetings in Scrum) so that everyone can contribute to discovering the whys, hows, and whats of the stories. You also break down bigger stories (a.k.a. epics) into smaller stories (a.k.a. user stories), and create new user stories together with your stakeholders in these workshops.

Creating shared understanding as a team with your stakeholders gives you the following benefits:

1. You are more likely to create a solution that really delivers business value. The close collaboration makes you understand the expected outcomes. You understand why this functionality needs to be created and what business value it addresses.
2. You are better equipped to develop the correct solution. You focus on the solution only after you have understood the needs and problems from the customer's point of view. Without this understanding, agile just helps you create the wrong stuff faster.
3. Your mindset and approach shift from finding defects to preventing them at the level of business value and development. Remember that finding defects is a waste from a lean point of view.
4. You are able to write down in unambiguous acceptance tests what is meant by successfully developing a user story. You develop only what is needed – no more, no less.

Story workshops not only create a shared understanding and acceptance test cases, they also open up the opportunity for the whole team to test. The test cases can be automated, so the developers have work to do together with the testers. The test cases need to be fleshed out and extended with new ones, and everybody can do that now.

The common understanding also sets you up for defining exploratory test charters together with your team. And, as you know, everyone can execute a manual test as long as an experienced tester is coaching.

That is all very interesting, but how can you actually run a successful story workshop? What steps are needed, what games can you play, and how can you facilitate the meeting?

In this article we will tell you about how we do our story workshops using serious games.

## What is a Serious Game?

If you are like us, you have participated in meetings that were boring, non productive, dominated by a few people, and just a pain to be part of. You just showed up and said a few things because that was what management expected you to do. Well, it does not have to be that way. You can use game mechanics in all your meetings to make them not only more fun, but much more productive with better results. A way to successful meetings is through serious games.

A serious game [1] is a game designed to solve a business problem. In a “normal” game the purpose is to entertain. In a serious game you use the game mechanics that engage millions of people playing “normal” games all over the world. A serious game puts people into a creative environment where they are really engaged and can collaborate with others, moving things around and discussing points of view.

Places where you can use serious games are when understanding stakeholder needs, clarifying user stories, and distilling test cases. We do that in a story workshop.

## What is a Story Workshop?

An intended outcome of a story workshop is to create a shared understanding of user stories for the whole team and to break them down to the right size. You create a shared understanding by discussing why they are needed, what problems they solve for the customer, and by distilling test cases to create that shared understanding.

Outputs of the story workshop for us are:

1. We want to have two or three examples for each user story to be refined.
2. We want to have an exploratory test charter defined of each user story to be refined.
3. We want a size estimate in order to make tradeoffs.
4. We want an estimate of the impact the story is going to make on the intended business value.

A good time slot for a refinement meeting is between one and two hours. You can always stop the meeting if you finish early, although this seldom happens because of Parkinson's law [2].

The first problem you address in a story workshop is to better understand the user stories. A user story is a narrative of a need of a particular persona. You need to understand what problem and needs the persona actually has. This is a discovery problem. The question to be answered is: “What problem does the customer have and why does he/she want it to be solved?”

Another problem you address in a story workshop is a design problem. A user story is also an experiment for which a solution must be designed by development. Therefore, the question to answer is: "What solution best fits the customer's needs?" The exact details of the design are not provided by a story workshop but it does start off the thought process about a solution.

Finally you have a testing problem. The questions to be answered are:

1. How do we know that the solution solves the customer's problem? Does our solution add more value than his/her current solution?
2. How do we know that we are solving the right problem? Is the problem we are solving the problem the customer wants to be solved?
3. How do we know we have solved the problem correctly? How do we quantify the success and failure of the solution?

In a story workshop we try to address all of these questions.

## How to Facilitate a Story Workshop

There are a few things you need to take into account if you want a successful workshop.

First you need to state the goal of the workshop. It is very important to state the goal at the beginning of the meeting in order for people to become engaged.

Next you need to discuss the steps of your workshop clearly. So, what is the agenda and what are we going to do?

After that you will have to define the rules of the workshop. How about ringing phones? Can we read emails during the workshop? What about interrupting one another while talking? If you have already done some story workshops with the team, you just quickly remind them of their own rules and ask if they are still happy with them.

To boost creativity everything is time boxed, so you want to communicate the time limit.

During the workshop you also want to inform the team of time progress. You can, for example, let them know every 10 to 15 minutes how much time is left and discuss whether you are still working on the most important things.

Finally, a parking lot is very useful to have. If you have any questions or issues that take up too much time or are not relevant, you can put them on the parking lot and cover them at the end of the meeting.

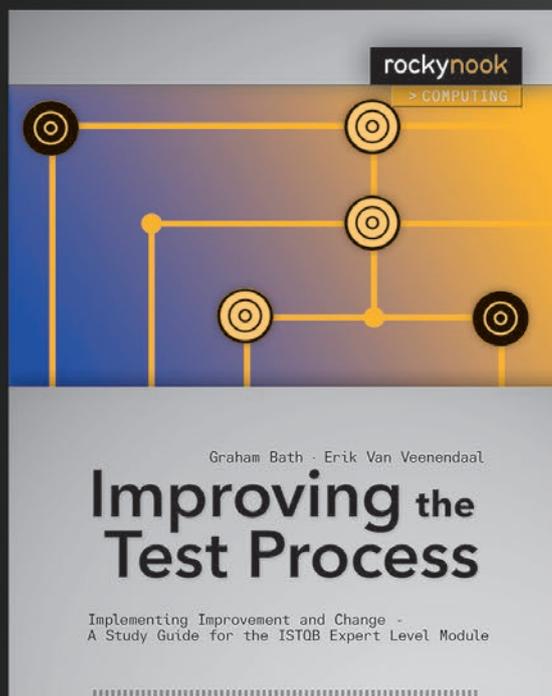
## A Game Sequence for a Successful Story Workshop

In a story workshop for distilling test cases to create shared understanding, you want to perform the following steps:

1. **Check in:** Explain the goal and agenda of the meeting.
2. **Understand the business value:** The product owner tells a coherent set of stories with a goal (Sprint Goal if you are using Scrum) and links them back to the business objective. The team discusses "why we are doing this." We do this using an impact map [3] and the 5 Whys [4].

# Ace the ISTQB Expert Level Module

## A Study Guide for the Experienced and Aspiring



### New Must-Have Book for Testing Experts

Authors Erik van Veenendaal and Graham Bath, who are among the creators of the ISTQB Expert Level syllabus, bring to you **Improving the Test Process**. This book covers the syllabus for the Improving the Test Process module of the ISTQB Expert Level.

This invaluable study guide is for those planning to take the ISTQB Expert Level exam, as well as experienced testing professionals who want to develop their skills in improving test processes.

#### Topics covered include:

- ▶ Model-based improvement
- ▶ Analytical-based improvement
- ▶ Selecting improvement approaches
- ▶ Processes for improvement
- ▶ Organization, roles, and skills
- ▶ Managing change

ISBN: 978-1-933952-82-6  
Pages: 432  
List price: £38.50  
Available at Amazon.co.uk

rockynook

www.rockynook.com

3. **Understand the customer value:** The team breaks up into two sub-teams, and each sub-team gets half of the user stories. The sub-teams then create scenarios of the current situation of the persona so that the team understands the persona's challenges as they are currently. The sub-teams also create scenarios of the persona with the user stories' solution implemented, so that the team understands the benefits the persona has from the new solution. The teams then get together and discuss the scenarios they have created with the other subteam. The team discusses "Why does the persona want this?" We do this using a storyboard [5] and a Pain Gain map [6]. This is also the step where you quantify your goal so you know how much value you have created at the end of the iteration.
4. **Distill acceptance tests:** The team then creates acceptance tests for the user stories. Depending on your tools, you create the use story narratives with Gherkin specifications [7], flow tables [8] and decisions tables [9]. The teams break up into sub-teams, and write tables and Gherkin scenarios in collaboration with the product owner. The sub-teams then get together and discuss the results with each other. We do this at the whiteboard using tables and scenario writing.
5. **Define exploratory test charters:** Identify risks to target your exploratory tests. Once you have identified which stories need manual testing, you set up an exploratory test charter for each. We do this using a risk impact matrix [10] and we use an exploratory testing tour to drive our test [11].
6. **Closing:** Quick summary of results and final remarks.

The above game sequence is just one way of running a story workshop. It assumed that you have ready user stories to begin with. The games explained are the games we use most of the time. There are a lot more games you can use. We encourage you to try some out in your workshops to discover which work best in your particular context. ■

## References

- [1] Serious game/innovation games: <http://www.innovationgames.com>
- [2] Parkinson's Law: [http://en.wikipedia.org/wiki/Parkinson's\\_law](http://en.wikipedia.org/wiki/Parkinson's_law)
- [3] Impact Mapping: <http://www.impactmapping.org>
- [4] The 5 whys: <http://www.gamestorming.com/games-for-problem-solving/the-5-whys/>
- [5] Storyboard: <http://www.romanpichler.com/blog/agile-scenarios-and-storyboards/>
- [6] Pain Gain map: <http://www.gamestorming.com/games-for-design/pain-gain-map/>
- [7] Gherkin Language: <http://cukes.info/gherkin.html>
- [8] Flow tables: <http://fitnesse.org/FitNesse.UserGuide.FixtureGallery.ImportantConcepts.FlowMode>

- [9] Decision tables: <http://fitnesse.org/FitNesse.FullReferenceGuide.UserGuide.WritingAcceptanceTests.SliM.DecisionTable>
- [10] Risk quadrants: <http://pascaldufour.wordpress.com/2013/11/19/user-story-risk-quadrants/>
- [11] Testing tours: <http://michaeldkelly.com/blog/2005/9/20/touring-heuristic.html> and: [http://msdn.microsoft.com/en-us/library/jj620911.aspx#bkmk\\_tours](http://msdn.microsoft.com/en-us/library/jj620911.aspx#bkmk_tours)

## > about the authors



**Cesario Ramos** is the founder of AgiliX Agile Consulting where he works as a professional Agile Coach and Scrum trainer @ Scrum.org. He helps teams to deliver maximum value within the constraints of time and budget. Cesario also leads agile transformation programs at his customers through coaching, training and hands-on guidance of management and development teams. When Cesario is not coaching or enjoying family life he enjoys writing and speaking. Cesario is the author of the book EMERGENT – Lean Agile adoption for an innovation workplace and speaker at international agile conferences where he passionately discusses the latest agile topics. You can contact Cesario at [cesario@agilix.nl](mailto:cesario@agilix.nl).



**Pascal Dufour** is a passionate tester and Scrum coach at Validate-it. Pascal has a passion for Agile projects, where he tries to implement pragmatic test strategies combining Agile and context-driven testing. With over ten years of experience at large international companies, Pascal has experience with different types of testing from embedded software to system integration. Enthusiastic and creative, he tries to make testing more fun, making his work visual and as simple as possible. He helps team members improve their efforts in Scrum, emphasizing ethics, commitment, and transparency, and motivating people to use a dynamic approach to testing. He believes teams should learn, try, experiment, and work together to create solutions that solve problems. You can contact Pascal at [pascal@validate-it.nl](mailto:pascal@validate-it.nl).