

Agile Software Developing and Testing Process

Valli.Rennig@Helmes.ee

Test Engineer

2009

- The Mission-Critical Systems developing IT firm in Estonia
- Number of developers over 160
- We have been and want to be the most efficient IT firm in Baltic States

Development team policy

- ❖ Team members should work closely within the team.
- ❖ Team members must share the information between each other.
- ❖ Developers must continuously pay attention to the technical excellence and good design.

Development process policy

- ❖ Development process is based on story cards.
User stories – captured user requirement, with following aspects:
 - focus on customer needs;
 - should be short;
 - user story management in testing management system.

Story cards – documented user requirement:

- card should comprise these parts:
 - Card** – name/description of the user story.
 - Conversation** – more detailed description of the user story.
 - Confirmation** – test-cases for the story.

Agile software development policy

- Unit tests are required.
- Collective code ownership.
- Automated build process (continuous integration) is required.
- “Simple is best” approach to software design.
- Requirements for the coding:
 - ❖ commented coding;
 - ❖ coding standards.

Testing process policy

- ❖ Acceptance tests should be created before the development.
- ❖ Testing should start as early as practical and will take place continuously throughout the lifecycle.
- ❖ Quality should be everyone's concern.
- ❖ Testing is the headlights of the project and provides information to the team.
- ❖ Multiple software iterations are required.
- ❖ Only working software will be delivered.

Skills for the testers in agile testing process.

- Domain knowledge about the system under test.
 - ❖ Testing Desktop applications
 - Focus on a specific environment;
 - Application testing can be done at once.
 - ❖ Testing Client-server applications
 - Two components to test – server and executable application on the client machine.
 - Application testing involves categories like: GUI (on both side), functionality, load, client – server interaction, back-end.
 - The knowledge of the number of the clients and servers.

- ❖ Testing Web applications (different and complex to test)
 - When you are testing the web application, you must consider following aspects:
 - numerous application usage paths are possible;
 - user knowledge about using the application;
 - security aspects;
 - network speed;
 - different types of OS platforms and browsers.
 - Web application testing involves also categories like: GUI, functionality, load, client - server interaction, back-end.

- The ability to treat development as the customer of testing.
- A level of technical competency to be able to interact effectively with the development team.

Therefore tester should have an ability to:

- ❖ understand the technology used;
- ❖ understand the context of the developed software;
- ❖ understand the written code.

Agile testing process in real life

- Developers viewpoint in matters of agile testing – their expectations.
- Best practices for testers in the agile testing process – focus on the functional testing area.

Developers expectations for testers:

- ❖ Practical and patient
 - Must understand the problems that arise.
 - Must understand the business logic.
- ❖ Technologically apprehensive
 - Understands technical logs.
 - Able to make white-box testing, when necessary.
- ❖ Shows the initiative for solving the bugs, issues.
 - Reminds developers of unsolved issues.

- ❖ Able to describe the issue precisely (reproducible issue)
- ❖ Able to analyze business logic
 - Able to associate problems, situations.
- ❖ Able to conduct testing in systematic manner
 - Tester should not do the testing as a monkey. Testing activity must be an organized activity.

Best practices or advices for the testers in the real life agile testing process.

- ❖ Learn the system and evolve test ideas through the iterations.
- ❖ Build test data between the iterations.
- ❖ Gather all possible questions and ask them all at once – in this way you disturb the developer less.
- ❖ Ask all questions raised in clear and compact manner.
- ❖ Think forward and use your domain and system knowledge to prevent further problems.
- ❖ Do not assume anything – if there's any doubt clear it out.
- ❖ Note all observations and changes in functionality.
- ❖ The written issue must motivate the developer to solve it.

Thank you for your attention!
Questions?

Valli.Rennig@Helmes.ee