Session based testing in Agile

Authors: Mahathi Ramya Vanapalli
         Shiva Shanker Pasupunuri

Reviewer:
Background

Agile is an iterative and incremental software methodology to deliver working software in time boxed sprints. Many organizations prefer agile to other life cycle methodologies due to the advantages like frequent delivery of working software, efficient collaboration between cross functional teams and flexible self organizing teams. Since software testing is a major area to ensure quality, it is imperative to use new testing approach to pickup with the pace and process of agile to ensure quality deliverable on time. So, there is a need to use a testing approach which can gel well with 'sprint' culture and give fast results.

Session based testing is an already proven way in software testing industry to measure and manage exploratory testing. Since exploratory testing is un-scripted, structured testing, it provides quick test results and testing approach is improved based on creative on the fly design. Agile methodology also prefers creative and collaborative way to improve quality with less documentation and more focus on ensuring quality with simple, but understandable user stories. Thus the combination of Session based testing in Agile methodology will give fruitful results for delivery of a good working software.

This paper talks about the session based testing approach and how it can be customized to fit into agile process.

Testing challenges in agile model

There are many challenges while testing in agile model. First one being getting accustomed to frequent changes in requirements and revising the test strategy on regular basis. Testing cycles are shorter making it difficult to analyze requirements author and execute test cases. This may lead to inadequate test coverage.

Since working software is delivered and tested in bits and pieces, there is high risk of missing integration points. In agile model, code that worked on earlier sprints may get modified by new feature changes, increasing the need of regression testing at system level.

Introduction to Session based testing

What is session based testing?

Unlike traditional script based testing, exploratory testing is an approach to find bugs fast by focusing on high risk areas and minimizing the time spent on documentation. Though this approach is affective, it is difficult to organize the process and summarize the test results. Session based testing is an approach to make exploratory testing organized without obstructing the flexibility and creativity of testers. Session based testing is nothing but managing testing based on sessions.

As per testing expert James Bach, A Session is an uninterrupted block of reviewable, chartered test effort. ‘Chartered’ mean that each session is associated with a ‘mission’. Usually, a session will range between 45 minutes to 2 hours. During this session, a tester may analyze a function, validate a feature or verify a set of bug fixes. A Debrief meeting happens after the session to discuss on the issues found and the action to be taken. The main objective is managing the testing activity with good coverage and accountability by making the process more adaptive.
### How different it is from traditional testing approach?

<table>
<thead>
<tr>
<th>Traditional testing Approach</th>
<th>Session based testing approach</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test execution takes place after test design</td>
<td>Test design and execution takes place at the same time with more focus on manageable unit</td>
</tr>
<tr>
<td>Predefined test procedures</td>
<td>Test design on fly which helps in improving test strategy as per the changes</td>
</tr>
<tr>
<td>Time spent on documentation will be more</td>
<td>Less time spent on documentation and more time on testing</td>
</tr>
<tr>
<td>Less freedom for the tester to explore more around the feature and restricted to the test case</td>
<td>More freedom to explore around the test features</td>
</tr>
<tr>
<td>Structured way of testing, more easy to monitor and report</td>
<td>Not as structured as traditional, possible to monitor and report</td>
</tr>
<tr>
<td>Experienced resources are required to create and document test scripts, but anyone can execute</td>
<td>Experienced resources are required to create and execute tests</td>
</tr>
<tr>
<td>Repeatable and reproducible</td>
<td>Not repeatable and reproducible until it is documented properly</td>
</tr>
<tr>
<td>Test case pass percentage might be more, eventhough defects are more</td>
<td>Test strategy would be updated as per the defects found</td>
</tr>
<tr>
<td>Lot of rework in cases of changes in requirements, since documentation is more</td>
<td>Comparitively easy to adapt the change as documentation is less</td>
</tr>
<tr>
<td>Quality assessment is easy</td>
<td>Quality assessment might be difficult – more focus on finding defect and these might not be user-centric scenarios</td>
</tr>
</tbody>
</table>

### Our Approach – Session based testing in Agile

Session based testing can be followed as a 3 steps approach.

1. Preparing for the session
2. Conducting the session
3. Debriefing about the session

#### Preparing for the session

In Agile model, software delivery is planned in sprints which are of 2 to 3 weeks duration. A number of user stories are scoped for one sprint, which will be anlayed, developed and tested within sprint cycle. To prepare for a session, the test lead along with team should understand, analyze and discuss each user story. It can be called as a ‘Kickoff’ session, where testing team brainstorms on splitting up the user story to different test charters. After deciding on scope of each user story, test lead can comeup with a ‘test charter’, which lists the objective of each session. The charter may focus on a new feature in a user story or defect retests and regression or looking into a particular problem.
Characteristics of a “Session”:

- Timebounded (can vary from 1 hr to 2 hrs)
- Charter for each session which defines the scope
- Uninterrupted
- Test report which can be reviewable

Characteristics of a “Charter”:

- Charter is an objective or mission for the session
- Provides information to the tester to focus during the session
- Can be includes the suggestions like what to test, how to test, where to explore
- Documentation of details related to time, references, test data etc.

A sample test charter:

<table>
<thead>
<tr>
<th>S.no</th>
<th>Scenarios/ Features to be tested/ Test ideas</th>
<th>Expected Results/ Oracles</th>
<th>Actual Results</th>
<th>Issues/ Defects</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Conducting the session

A test charter would define what to be tested and what aspects of the application to be considered like usability, performance etc., A typical session will include a pair of testers working on a charter for uninterrupted time (usually, 1 hr to 2 hrs excluding the time spent on setup, understanding the objective of session). Strong domain knowledge and experience would help for the testers in session based testing.

While conducting a session, we should ensure that it is uninterrupted (no emails, no phone, no discussions etc...). Maintaining the focus on objective alone is important instead of touching on other areas of application. If a feature cannot be tested due to time constraint or technical limitation or dependency on another defect, these should be mentioned in Comments section of test charter. Actual testing time alone should be counted in the session. Time spent on discussing and understanding the feature to be tested, collecting test data, setting up the environment etc., to be done before starting the session as part of ‘Preparing the session’. Testers should be encouraged to try creative ways to break the system and look for different ways on how it might fail to meet user’s expectations. All variations in the expected result to be noted in ‘Issues/ Defects’ section, which can be logged into defect tracking system after Debrief which follows the session.

Debriefing the session

Debriefing is the meeting between test leads and testers who completed the session. Main objectives are to understand and accept the session and to provide feedback and coaching to the tester. Depending on the experience of the testing team members, these debriefs will take less time.

This meeting plays an important role in planning and estimating session times. Team also discusses on what went well, what items need to be tested again, what can be improved etc., This helps in collecting metrics as well, like time taken for the session, number of defects found, test coverage etc., A test manager can also understand the challenges faced by testers during the session and decrease the scope or increase time of the session depending on the feedback.

5 important topics covered by a test manager in a debrief:

- What happened during the session?
- What was achieved?
- What were the challenges faced?
- What are pending?
- Feedback from tester to improve

Results from the case study

The project was building a client server application for a long running Financial need for one of the client. The application will be used by financial users across 18 countries. R2.0 was developed using waterfall, for R3.0 and R3.1 scrum methodology was adopted.

Cross functional teams signed up for 8 sprints for R3.1.
Good practices

There are no best practices in session based testing, below are some of the good practices observed during implementation.

- Optimal duration for a session can be between 1 to 2 hours and plan not more than 3 sessions in a day. Don’t impose the time on testers very strictly because they may more obsess with time than testing.

- Uninterrupted quality time during a session (60, 90 or 120 min.), can be achieved by conducting the session in a conference room with no meetings, chatting, emails or telephone calls.

- Charter should be prioritized and should be broken down in to smaller logical units which can be tested completely during a session. This will help in planning and completing the user stories as per the timelines.

- If tester performs testing outside the session and realizes should have been in the session, create a session sheet for the testing done outside session or add to the existing session sheet. Testers should always get the credit for testing they have done.

- If a session needs to be interrupted because of any mandatory meetings or tester is called by developer to help in reproducing a defect, then session should be suspended and resumed later.

- For a new tester in the team or to the session based testing, can be given areas that are already tested which required to be regressed again as part of bug fixes/changes. Tester can review the test notes from the previous...
session and refine/change wherever required as applicable to the current session. This will train the tester and absorb the impact of new team members.

- In case of two testers are paired in a session on a single charter, they can have one common session sheet and test notes such that they work together and cooperate with each other and save time in doing redundant work.

- Testers sometimes tend to deviate from the charter like while testing find a new opportunity or spends time on investigating a bug then test manager can change the charter to fit whatever the tester has done. A new session should be set up to test the original charter.

- In debriefing, came to know that a charter is not achieved because of any reason, then the charter should be cancelled and new session should be planned. If the tester made lot of progress but did not complete the charter as it is much bigger than expected, then the scope of the charter can be reduced and create more charters by dividing into small chunks.

Challenges

- Resistance from testers as this is different from scripted testing and may not be how they do it.

- Experience of testers on the application and the expertise of testers in exploratory testing as not everybody can explore things in right way at the right time.

- Inconsistent documentation of charters, session notes, logs and time consumed on different aspects in a session which makes difficult to repeat and reproducible.

- Difficult to consolidate and come up with the test suite whenever there is a need for regression during stabilization period.

- Difficult to produce various testing metrics and reports based on the session notes and charters which makes quality assessment difficult.

Conclusion

- Session based testing focuses more on actual testing in a smart and effective way than the scripted testing.

- Very focused and stronger connection between testing and test objectives.

- Scripted testing is more efficient in finding critical bugs quickly which gives sufficient time to the team to fix within the sprints before customer finds.

- Scripted testing gives tester more freedom to be innovative and explore out of the scripted test cases.

- Very flexible and easy to conduct and test effectively in case of partial or periodic delivery of code and helps in assessing the health of the code at initial stages.

- Since in agile requirements are incomplete or updated frequently, scripted tests becomes problem to maintain whereas session based testing is more suitable, meaningful and effective.

- Well structured, planned and documentation (sessions, charters, session notes, debriefs) can definitely provide complete coverage and tracking of the test results and in session based testing.

- Based on the results achieved from the case study and analysis, session based testing is very effective and more focused test approach that suits agile practices.
References

Internal

• https://kx.deloitteresources.com/G1000/Tags.aspx?Type=TagsView&TagId=3478&TagValue=Agile+Software+Development
• https://kx.deloitteresources.com/G1000/Lists/PublishedContent/Attachments/96210/Agile_Testing_Methods.pptx
• https://kx.deloitteresources.com/G1000/Lists/PublishedContent/Attachments/125209/Agile%20Eminence%20Contest–Software%20Testing%20Challenges%20in%20Agile%20Projects%20POV.pptx

External:

• Jonathan Bach, Session based test management, *Software Testing and Quality Engineering* magazine, 11/00)
• Anders Claesson, How to perform exploratory testing by using test charters
• Rony Wolfinzon and Ayal Zylberman, 10 Most Popular Misconceptions about Exploratory Testing.
• Carsten Feilberg, Session based test management in practice.
• Andrew Prentice, Testing’s in session
• Peter Tennekers, Session based testing.