Quest Journals
Journal of Software Engineering and Simulation
Volume 9 ~ Issue 12 (2023) pp: 05-08
ISSN(Online) :2321-3795 ISSN (Print):2321-3809
www.questjournals.org

Research Paper

Reverse Build Testing

A White Paper

Ankit Tripathi
Organization: Persistent Systems

Received 04 Dec., 2023; Revised 14 Dec., 2023; Accepted 16 Dec., 2023 © The author(s) 2023. Published with open access at ***www.questjournals.org***

I. Introduction

In the rapidly evolving landscape of software development, traditional methodologies are increasingly being replaced by Agile models that focus on adaptability and customer-centricity. While this shift has allowed for more flexible and responsive software creation processes, it has also introduced new challenges in quality assurance and testing. As we transition from legacy development practices, there is an acute need for updated testing strategies that can adapt to the Agile framework without compromising on quality or efficiency. This white paper introduces a novel testing technique—Reverse Build Testing (RBT)—tailored to meet these unique challenges.

II. Objectives

1. To outline the limitations of traditional User Acceptance Testing (UAT) in Agile environments.
2. To introduce the concept of Reverse Build Testing (RBT) as a more flexible and effective alternative.
3. To explain the phases and steps involved in implementing RBT within an Agile team.
4. To emphasize the benefits of adopting RBT, such as improved alignment between requirements and test cases, and identification of potential product improvements.

Reverse Build Testing

In the modern era of software testing, development methodologies have evolved significantly over time. Consequently, testing techniques must also undergo substantial modifications to align effectively and efficiently with these advancements.

In the traditional testing landscape, User Acceptance Testing (UAT) is generally performed after the software has been extensively tested. However, in the Agile development environment, where products are delivered in sprints/iterations, a more agile mechanism for UAT is essential.

To bridge this gap, we introduce a new testing technique known as Reverse Build Testing (RBT). This lightweight method can be executed by any team member and encourages out-of-the-box thinking. RBT not only ensures robust User Acceptance Testing but also identifies any discrepancies between requirements and test cases, as well as highlights potential areas for product improvement.

Unfolding the Phases and Steps for RBT Implementation

Phase 1: Playground
Step 1: One or two members, either from the same team or a peer team.
Step 2: The Product Owner will conduct a brief call to outline the Business Logic or Business Requirements of a feature/story if needed.
Expected Outcome:
The primary objective of the Playground phase is for the RBT actor to grasp a high-level understanding of the feature. The idea is to position the RBT actor similarly to an end-user, who cannot be expected to read through every line of the requirements but still needs enough context to proceed.

Phase 2: Play and record.

In play and record phase: RBT actor will play around the application without referring the business logic but business context and record their score in a single liner or just maintain brief description about what he has done and what behavior or result he encountered as the outcome, mentioning the reference below-

What I Play: What I Score:

As an addition The RBT actor will be listing down all the integration modules/integration dependencies in the features as **Integration Traceability Matrix**

Expected Outcome:
RBT actor will have consolidated score and integration traceability matrix prepared

Phase 3: Referee call
In referee call phase RBT actor will setup a meeting with team and share their 'Play and record' lists, each 'what I play & what I score' will be tagged by a card either by Green, Yellow or Red as per below criteria:

Green Card Item: Aligns with business requirements.
Yellow Card Item: Does not align but is seen as a potential future improvement or bug.

Red Card Item: Does not align with business requirements and is not considered a potential feature but offers a new perspective that could benefit the team.

Integration Traceability Matrix : After discussing Green, Yellow, and Red card items, the team will be updating the matrix to reflect any newly identified module interactions or third-party dependencies.

Phase 4: Goal

For Green Card Items:

- Make sure the item is covered under existing test cases. If not, add those to fulfill identified gaps between requirements and test cases.

For Yellow Card Items:

- Create a bug if the item is a bug.
- Create an incident if the item requires further investigations.
- Create an item in backlog (with PO) if the item is an improvement feature.

For Red Card Items:

- These items neither fall under business requirements nor are seen as potential features by the team. However, they offer a new angle that could be beneficial to the team.

Integration Traceability Matrix:

- Incorporated findings into the final Integration Traceability Matrix to ensure comprehensive coverage of all identified aspects during subsequent integration testing or regression testing.

Phase 5: Conclusion and Feedback
The final phase involves gathering feedback from all team members involved in the RBT process. The aim is to identify areas of improvement for future RBT iterations and to make sure any adjustments to the existing test cases or stories as needed as well as will be passing the Integration Traceability Matrix to subsequent testing team as and when needed for regression and integration testing further cycle .

Advantages of Reverse Build Testing

1. User-Centric Testing: RBT mimics real-world usage, giving teams insights into how actual users interact with the product.
2. Gap Identification: Helps in finding the missing links between the requirements and test cases, ensuring that all necessary conditions are tested.

3. Continuous Improvement: Yellow card items can serve as a trigger for future improvements, allowing the product to evolve.
4. Resource Efficiency: Being a lightweight method, it does not require extensive resources, making it suitable for smaller teams or projects with tight timelines.
5. Enhanced Communication: The process encourages cross-functional collaboration and communication between development, testing, and product management teams.
6. Focused Testing: By categorizing components with the colored card system, teams can easily identify the modules and integrations that require immediate attention, thereby focusing their efforts during integration or regression testing.

7. Efficient Resource Allocation: Knowing which modules are integrated and how they interact allows for better resource allocation and finding dependencies.

8. Clear forward and backward Traceability: This matrix allows for seamless traceability between different testing phases and the modules or components involved as well to across iteration cycles.

Disadvantages of Reverse Build Testing

1. Limited Depth: While RBT is great for UAT with backwards compatibility, it might not catch intricate bugs that can be found using other focused testing methods.
2. Subjectivity: The outcome of the RBT is influenced by the expertise and perspective of the RBT actor, making it somewhat subjective.

III. Conclusion

Reverse Build Testing offers a flexible and efficient approach to User Acceptance Testing in Agile development environments. While it is not a complete replacement for traditional methods, its advantages make it a valuable addition to any Agile team's testing toolkit. As with any methodology, teams should weigh the advantages and disadvantages carefully to determine if RBT is suitable for their specific needs.

Use Case: Testing Telemedicine Video Consultation on a Healthcare Platform

Phase 1: Playground

Step 1: Amit, an experienced QA engineer specialized in healthcare applications, is designated as the RBT actor.
Step 2: The Product Owner gives Amit an overview of the telemedicine video consultation feature, emphasizing its key role in remote healthcare delivery.

Expected Outcome: Amit gains a understanding of the feature's objectives and requirements.

Phase 2: Play and Record

Amit navigates through the application, attempting various activities while documenting the outcomes.

Amit's List:
What I Play: Log in and schedule a video consultation.
What I Score: Consultation booked without issues. Confirmation email received.

What I Play: Join the consultation via smartphone.
What I Score: Video and audio work seamlessly.

What I Play: Attempt to share medical documents during the consultation.
What I Score: Feature not available.

What I Play: End consultation and view session summary.
What I Score: Session summary is generated but lacks downloadable format.

What I Play: Seek out a specialist for a second opinion through the platform.
What I Score: Option to search by specialty is not available.

What I Play: Access the consultation on a low-bandwidth connection.
What I Score: Frequent disconnections and video quality degradation.

Integration Traceability Matrix: Mark the 'User Authentication' and 'Email Notification Service' modules as interacting successfully**.**

Phase 3: Referee Call
Amit presents the findings to the team.

- **Green Card Items**: Easy consultation booking, video and audio quality.

- **Yellow Card Items:** Inability to share documents, no downloadable session summary, specialist search feature missing.'

- **Red Card Items:** Poor performance on low-bandwidth connections.

- **Integration Traceability Matrix:** After discussing Green, Yellow, and Red card items, the team updates the matrix to reflect any newly identified module interactions or third- party dependencies.

Phase 4: Goal

- **Green Card Item**: Validate test cases for basic functionalities and video/audio performance, add any new test case to fill the gap.

- **Yellow Card Item:** Create a bug for the absence of a document-sharing feature and the lack of a downloadable session summary. File a story to add a search by specialty feature.

- **Red Card Items:** Investigate the platform's poor performance on low-bandwidth connections as an interesting angle for performance optimization/Add as an enabler in backlog.

- **Integration Traceability Matrix**: The matrix is reviewed to ensure that all test cases cover the interactions and dependencies identified.

Phase 5: Conclusion and Feedback Loop
Green, Yellow Red Card: The team synthesizes the findings, refines test cases, and populates the backlog with new stories and bugs for further investigation.

Integration Traceability Matrix: The finalized matrix is reviewed one last time during the feedback loop. Any items identified for future sprints/iteration are marked, and the matrix itself becomes a point of discussion for continual improvement and keep tracks which module need to be regression tested going forward in case of any changes happens in a particular module.