ATI Approach Description

V1.0, 2024/October

Sónia Cunha (MSc Student)
Alberto Rodrigues da Silva, Miguel Mira da Silva (Supervisors)

Instituto Superior Técnico, Universidade Aberta, Portugal

The ATI (Acceptance Test Improvement) approach was developed in to involve and facilitate the application of RSL and Generative AI improving the acceptance testing process. It includes a process diagram (Figure 1 and Attachment A with zoom) and a best practices guideline.

A - Process Diagram

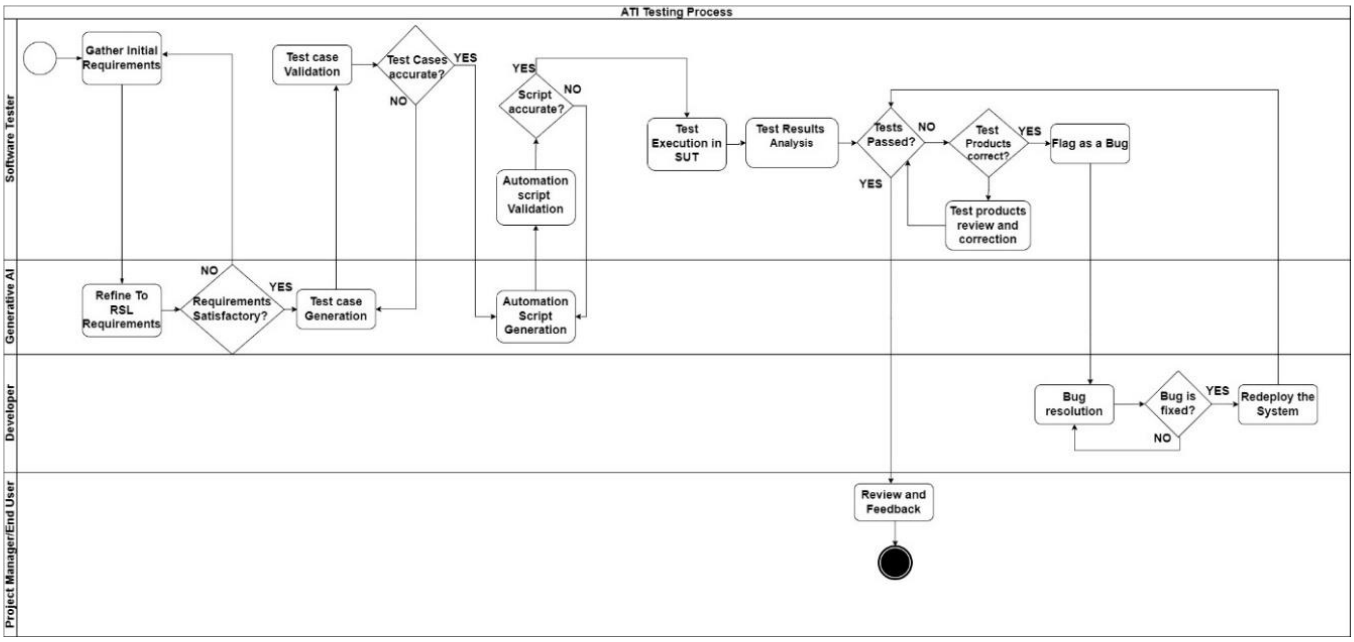


Figure 1 – ATI Approach process diagram

B – Best practices guidelines

To guide Testers in applying the Acceptance Testing Improvement (ATI) approach, which includes the use of both Generative AI and Requirements Specification Language (RSL) to improve the User Acceptance Testing (UAT) process, the following guidelines provide detailed steps to ensure a smooth and effective testing process.

1. Introduction

Objective: The ATI approach aims to enhance the UAT process by integrating Generative AI and RSL to improve efficiency, clarity, and automation in refining requirements, creating test cases and automated tests for User Acceptance Testing process.

Scope: These guidelines help Testers applying the ATI approach during the UAT phase of the SDLC, focusing on the support of AI and RSL in facilitating and refining Testers tasks.

2. Preparation and Setup

2.1 Identify Stakeholders:

- Project Manager: Manages overall UAT progress.
- Software Tester: Creates and automates test cases.
- End User: Reviews and provides feedback.
- Generative AI ("Software Testing Guide"): Assists in refining requirements, generating test cases, automation script and traceability matrix.
- RSL: Provides a structured way to document requirements in a formal language that reduces ambiguity.

2.2 Tools and Resources:

- Generative AI Software (e.g., "Software Testing Guide"): Ensure Testers have access to the AI tool.
- RSL Documentation: Use the RSL language to write clear, concise, and testable requirements.
- Automation Framework: Set up tools such as Robot Framework to automate UAT tests.
- Documentation Tools: Excel, Word, or test management software to organize requirements and test work products.

2.3 Data Collection:

- Collect relevant documents, including the SRS (Software Requirements Specification), user stories, functional requirements, and UI designs.
- It´s recommended that these documents are expressed in the RSL format for consistency and clarity.

3. Process Guidelines for ATI Approach

Step 1: Requirement Refinement using RSL and AI

Involvement: Software Tester, Project Manager, Generative AI.

Actions:

- AI-Assisted Refinement: Enter the requirements of the SUT into the Generative AI tool to further translate into RSL and refine them, to ensure they are clear, concise, testable and aligned with the UAT objectives. RSL reduces ambiguity by using a structured syntax for defining triggers, actions, and expected outcomes.

- Validate the refined requirements, making sure they are understandable for both Testers and automation tools.

Best Practices:
- Continuously involve the Project Manager to ensure business needs are aligned with refined requirements.
- Ensure that requirements documented in RSL are consistently updated.

Step 2: Test Case Development with RSL and AI

Involvement: Software Tester, Generative AI.

Actions:
- AI-Assisted Test Case Generation: Use Generative AI to create initial test cases based on the RSL requirements. AI should generate detailed test steps and expected results.
- Link to RSL: Ensure that every test case is linked directly to a specific RSL requirement.
- Manually review AI-generated test cases, ensuring that each step correctly validates the requirement.

Best Practices:
- Ensure test cases are clear and cover both positive and negative test scenarios.
- Document each test case properly, linking it back to the specific user stories and RSL-based requirements.

Step 3: Test Case Automation with Generative AI

Involvement: Software Tester, Generative AI.

Actions:
- Automate Test Cases: Leverage AI to generate automation scripts (e.g., using Robot Framework) from the developed test cases.
- Link Automation to RSL: Maintain a clear mapping of automated test scripts to the RSL requirements, ensuring full coverage of the documented functionalities.
- Review and refine the automated scripts to enhance efficiency and reliability.

Best Practices:
- Modularize the automation scripts to make them reusable and easier to maintain.
- Ensure automation scripts are optimized for performance and execution speed.

Step 4: Test Execution and Monitoring

Involvement: Software Tester, SUT.

Actions:

- Execute Automated Tests: Run the automated tests generated with AI against the SUT.
- Monitor Test Execution: Keep track of any deviations or failures during the test execution phase.
- Capture Results: AI tools can assist in capturing and organizing test results for further analysis.

Best Practices:

- Ensure the testing environment is a close replica of the production environment for accurate results.
- Test under multiple conditions to ensure broad coverage of real-world usage scenarios.

Step 5: Reporting and Feedback

Involvement: Software Tester, Project Manager, End User, Generative AI.

Actions:

- Generate AI-Assisted Reports: Use Generative AI to create detailed test reports, summarizing the results of test execution, including passed/failed tests, coverage reports, and defect logs.
- Provide Feedback Loop: Share test reports with the Project Manager and End Users for review and feedback.
- Use the feedback to make iterative improvements in the requirements, test cases, and scripts.

Best Practices:

- Include clear action points and insights in the report to ensure that feedback is actionable.
- Keep stakeholders informed of progress and key testing milestones.

4. Best Practices for ATI Approach

a. Collaboration:

- Foster close collaboration between Software Professional Testers, Project Managers, and AI to maximize the effectiveness of AI and RSL in refining requirements and generating test work products.

b. Sequencing:

- For more consistent results, traceability and efficiency, perform all the test work products generation sequentially by a reference like an identifier (ID). For example, first generate all the test products for requirement ID 1, then generate all the test products for requirement ID 2, and so on.

c. Continuous Learning:

- Update the AI tool with the latest project information and feedback from stakeholders to ensure continuous improvement in its performance and output.

d. Data Security:

- Ensure there is a formal contracted agreement between all stakeholders giving permission to share SUT (System Under Test) information in generative AI prompts.
- Adhere to strict data security protocols and contracts when using AI and RSL to handle sensitive project data.
- Ensure test environments and production environments maintain data integrity.

e. RSL Customization:

- Customize RSL rules based on the project-specific needs of the SUT, ensuring that the syntax accurately reflects the functional and technical requirements.

5. Continuous Improvement

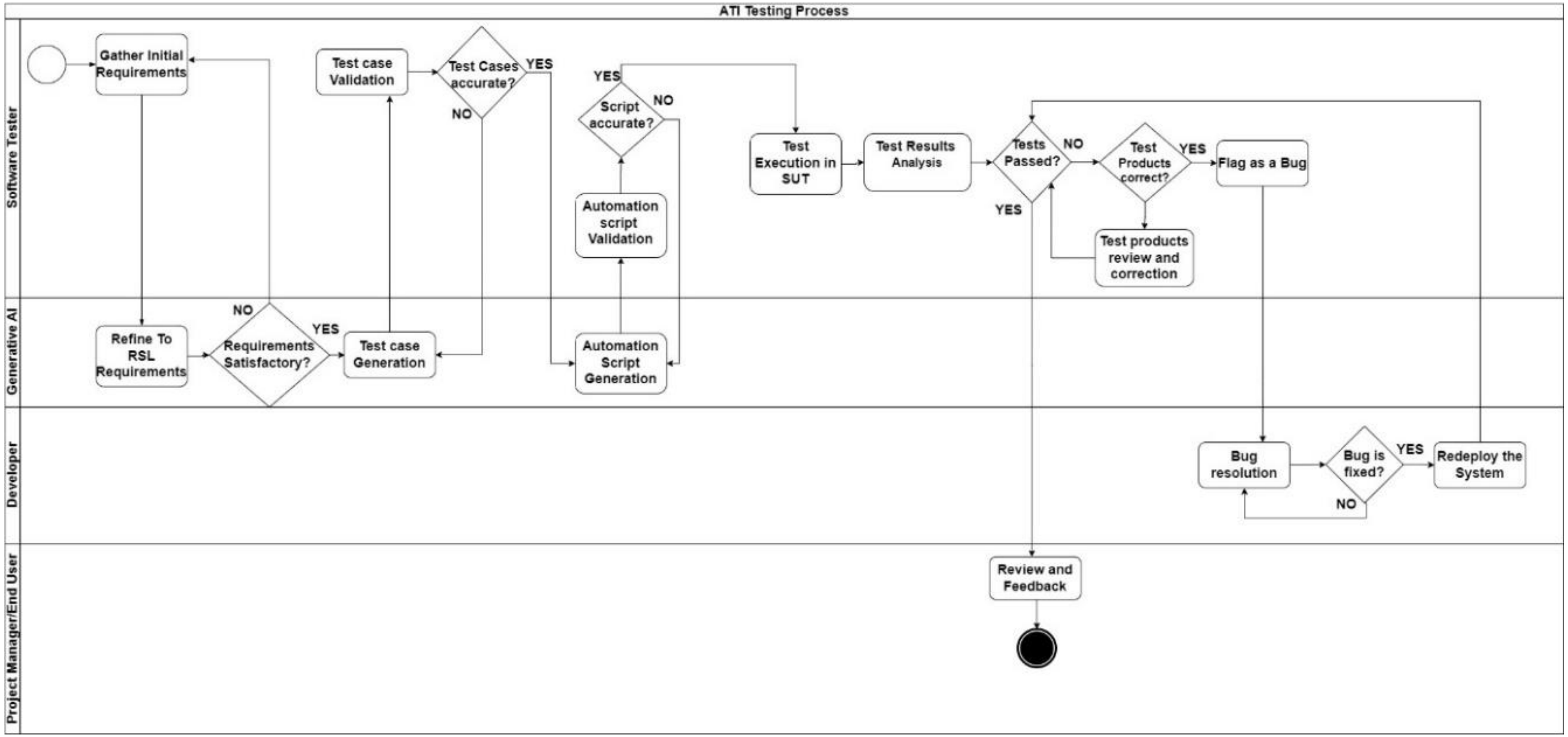
- Treat the UAT process as iterative. Feedback from test results and reports should be used to continually refine requirements (in RSL) and improve the test cases and automation scripts (with AI).
- Regularly review the AI tool’s performance and RSL clarity to ensure that they keep up with evolving project requirements.

6. Conclusion

The combination of generative AI and RSL in the ATI approach enhances the UAT process by improving clarity, testability, promoting the generation of test work products and traceability between them. By following these guidelines, Software Testing Professionals can ensure that UAT activities are more efficient, accurate, and aligned with stakeholder expectations. This approach provides a structured method to level AI for streamlined test case development and automated execution while improving requirements that are defined clearly and objectively using RSL.

These guidelines should help Software Testing Professionals apply the ATI Approach effectively, ensuring successful UAT for the SUT.

Attachment A


ATI Testing Process
Software Tester
Generative AI
Developer
Project Manager/End User
Gather Initial Requirements
Test case Validation
Test Cases accurate?
YES
NO
YES
Script accurate?
NO
Automation script Validation
Test Execution in SUT
Test Results Analysis
Tests Passed?
NO
YES
Test Products correct?
YES
Flag as a Bug
Test products review and correction
NO
Refine To RSL Requirements
Requirements Satisfactory?
YES
Test case Generation
Automation Script Generation
Bug resolution
Bug is fixed?
YES
NO
Redeploy the System
Review and Feedback