

## Case Study #1

### Solution:

Per my knowledge and experience we need to adapt to different testing processes based on various testing methodologies.

#### a) Agile Methodology:

According to Agile methodology, larger goals are divided into small/short goals, i.e. there are shorter time frames called as sprints (about 14 days). Within each sprint there is continuous development, continuous testing and frequent releases. Basically, each sprint includes various scrum ceremonies like Daily scrum, Sprint Refinement, sprint planning, Sprint retrospective and Sprint Review. Hence the same regular SIT process needs to be adapted to fit in shorter cycles/sprints.

- **Test Planning:** This is actually done at the beginning of every sprint during sprint planning. Here the user stories are defined where the acceptance criteria is discussed and understood and high-level test planning is defined for the planned user stories for that particular sprint.
- **Test Case Design:** As the user stories are clearly understood by both the development and testing team during the Sprint planning activity, development and test case design is done parallelly. Test cases are designed by keeping in mind the Acceptance criteria of the user stories. Automated test scripts are developed in this phase.
- **Test Execution:** Continuous Sanity testing is carried out here to check the stability of every new build. This phase involves execution of Automated tests on a daily basis, exploratory testing and continuous tracking of application stability.
- **Defect Management:** As and when the defects are identified by the testing team, the defects are quickly assigned to the respective developers bringing the severity of the defect to the scrum team's notice. Usually, a quick Defect Triage call is scheduled everyday discussing the open defects with the dev, QA and business analysts to accelerate faster resolution and re testing of the defects.
- **Results and Analysis:** Daily testing updates and issues are addressed in the Daily stand-up meetings. A mini test report is usually provided at the end of every sprint. Other than that, at the end of the sprint a Short demo of the application is provided based on what features were built

and tested. Also in the Sprint Retrospective, the entire team analyzes what went well, what went wrong and what could we do to get better in the next sprint. This helps the team to analyze and learn from the mistakes done in the previous sprints.

#### **b) Waterfall Methodology:**

It's a sequential method where each phase is completed before the next phase begins. The Application is entirely developed and testing is done in the last phase unlike Agile Method where development and testing go hand in hand.

- Test planning: Involves creating a detailed test plan covering all testing phases- unit, SIT, UAT,
- Test design: Test cases are derived from detailed Requirement documents. This phase involved creating comprehensive test cases, traceability matrices for all functionalities. Test cases are usually reviewed by Test leads and Business Analyst's in this phase.
- Test Execution: Test execution occurs in a dedicated phase after all the development is complete and the application is stable. Sanity testing is performed as an entry point for the formal test execution phase.
- Defect Management: Defects are logged, prioritized and resolved systematically. But the defect resolution process is comparatively slow as any major defects the entire application has to be redeployed and tested again.
- Results and Analysis: A formal Test Completion Report is generated at the end of the entire testing phase, summarizing all test activities, results, and remaining defects.

#### **c) Hybrid Methodology:**

A hybrid approach combines both Agile and Waterfall Methods. Waterfall method is followed for initial planning and architecture. Agile method is followed for development and testing.

#### **i) How to achieve a high degree of test automation (more than 80%) in functional testing?**

- Selection of the Right Tools: Select automation tools that align with the technology stack (e.g., Selenium for web, Appium for mobile, Postman/Rest Assured for API).

- Scalable and Maintainable Framework: Develop a framework with clear coding standards, modularity (e.g., Page Object Model), reusability, maintainability and easy reporting.
- Start writing automated tests from the beginning of the development cycle (even before code is fully stable). This includes unit tests, API tests, and early UI tests.
- Automate Stable & Critical Paths: Focus on automating core business flows, frequently used functionalities, and high-risk areas.
- Automate Regression Suites: Build a regression suite of automated tests that can be run quickly and frequently to ensure existing functionality remains unaltered after any new modifications.
- Avoid writing automating tests where UI is expected to change frequently.
- Continuous Integration/Continuous Deployment (CI/CD) Integration: Integrate automated tests into the CI/CD pipeline. Tests should run automatically with every code commit or build.
- Data-Driven and Keyword-Driven Testing: Design tests to be data-driven, allowing a single test script to run with multiple sets of data, increasing coverage efficiently. Use keyword-driven approaches for common actions, making scripts easier to write and maintain.
- Investing on skilled and Automaton testers and their training.
- Continuous Monitoring and Maintenance of Automated scripts: check for test failures and refinements needed if any.
- Continuous review and improvement are crucial for maintaining an effective testing process.

**ii) Describe the strategies used for continuously review and improvements of the process. Continuous review and improvement are crucial for maintaining an effective testing process.**

- Regular Retrospectives: The Sprint retrospectives conducted in Agile projects focus on "what went well," "what could be improved," and "actionable items if any." This meeting involves all relevant stakeholders (developers, testers, product owners, project managers).
- Monitor metrics such as Defect density, defect leakage, test pass/fail rates.
- Regular reporting: In one of my projects where we used Azure DevOps for project management, we used Dashboards to generate reports on a

daily basis and sent it out to the larger team. This helped in identifying areas needing attention and clear reporting.

- Regular Feedback: Actively seek feedback from product owners, business users, and end-users regarding application quality and testing effectiveness.
- When any defect is resolved by the development team, always ask for an RCA (Root cause Analysis). Once the RCA is clear only then re test and close the defect.
- Stay Updated and Continuously learn new testing tools, frameworks, and technologies that could enhance efficiency and effectiveness.
- Conduct workshops and training sessions to share best practices, new skills, and lessons learned within the team.
- Maintain up-to-date process documentation. This will help in smooth KT (Knowledge transfer) for any new resources joining the team.

**iii) What makes you think that the proposed team members are able to execute the process tailored efficiently?**

To confidently state that proposed team members can execute the tailored process efficiently, I would need to consider the following points:

- Skills and Expertise: Need to check if the resource has a deep knowledge of the domain and the application under test.
- Familiarity with the Test Methodology being used: Need to check if the team members have knowledge and experience working on Agile/Waterfall projects based on the requirements.
- Technical Skills: For manual testers they need to possess strong analytical, logical, and observational skills, should be great with communications, attention to details, write clear and detailed test cases, Ad hoc and exploratory testing and test reporting. For automation Testers they need to be proficient in programming languages , automation frameworks, CI/CD tools, and version control etc.
- The team members should be always willing to learn and adapt themselves to new tools/frameworks as per market demand. They need to be quick learners.
- The team members should be great communicators and should be willing to take ownership of the testing processes. Also need to check if they are accountable for all the test deliverables and meeting the project deadlines.

- The team members should be team players and willing to share their knowledge with the team and help the team as and when needed.

## Case Study #2

### **a) Refine the process so that the process can be used to perform early test automation to achieve the Sprint Testing through automation in the Agile Sprints.**

The original automation testing process can be refined for early test automation in Agile Sprints as follows:

- Sprint planning: During sprint planning, identify test cases suitable for Automation and define scope of automation for the upcoming sprint.
- Design Automation scripts: As soon as the functional requirements or the initial wireframes are ready, start designing the automation scripts side by side with development.
- Test data management: Prepare required test data in advance. This should be an ongoing activity throughout the sprint.
- Execute the automated test scripts on daily basis as and when the new build is released. This can also be scheduled in advance.
- Review the automation scripts regularly and refine if needed. Maintenance of the test scripts is necessary.
- Monitor automation results and analyze failures to provide quick feedback to the development team.

### **b) Describe how does the process allow the Test Manager and Senior Test Automation Engineer to be able to start designing and developing the automation based on Figma screen design and/or wireframes design (provided by the UI/UX QA designer) and develop the test automation concurrently while the developers are developing the application so that automation can be used since day 1 of Sprint Testing until test completion.**

Test Automation can be used since day 1 of the sprint cycle by following below mentioned practices:

- When the initial UI/UX Wireframes or functional requirements are ready, The Test Manager and Senior Test Automation Engineer can initiate designing the automation scripts for the same.
- While developers are actively building the application, the Senior Test Automation Engineer can concurrently develop the automation scripts. This can include Framework Setup, API Automation, Test Data Preparation.
- By designing and developing automation concurrently with application development, the automated tests are ready to be executed as soon as the first Build is deployed in the sprint. This means that instead of waiting for a fully developed feature to begin testing, automated tests can be run from day one of sprint testing, providing immediate feedback to developers and identifying defects earlier in the sprint.

**c) What makes you think that the proposed team members are able to achieve this early test automation based on the refined process?**

The ability of the proposed team members (Test Manager and Senior Test Automation Engineer) to achieve this early test automation based on the refined process depends on below factors:

- Expertise of Senior Test Automation Engineer.
- Pro activeness of the Test Manager and Senior Automation Engineer.
- A collaborative mindset between the Test Manager, Senior Test Automation Engineer, UI/UX QA designer, and developers ensures that the team is aligned on the benefits of early automation and committed to frequent communication. By clear communications they can overcome challenges and achieve the desired outcome.
- The refined process aligns with the Agile principles. By embracing early automation, the team can deliver higher quality software faster and with greater confidence.
- Having a dedicated like person like a Senior Test Automation Engineer, rather than just relying on manual testers, ensures that there is specialized expertise focused solely on automation development and maintenance. This allows for dedicated effort and continuous improvement in the automation suite.