enhops
A proarch company

ImpactNOW
QA Assessment & PoC Discovery Sample Report

Your Partner in Software Testing

Companies have a lot to gain from Test Automation.
But before you blindly jump in, it’s important to understand gaps, risks analysis and have a tailor-made plan.

Enhops is here to help wherever you are in your Test Automation journey.

1. INTRODUCTION

At [Client Name], the goal is to enhance the QA process by incorporating industry best practices into the Software Development Life Cycle (SDLC), leveraging existing tools and infrastructure. The business objective is to prevent critical defects from reaching production, increase test automation coverage, and implement continuous testing in CI-CD pipelines.

To support this initiative, [Your Company] conducted a “Discovery Phase” aimed at analysing the current QA setup, providing process improvement recommendations, and defining a strategic roadmap for QA transformation.

This document highlights the key findings and recommendations from the Discovery Phase, with a focus on achieving a mature testing process, implementing industry best practices, enhancing testing coverage, providing relevant training, and evolving towards a robust and scalable QA model.

2. STAKEHOLDERS

The following key stakeholders have been involved in the discussions and review of the QA process to ensure alignment with business objectives:

- **Leadership Team**: [Leadership Team Name(s)]
- **[Department/Team Name]**: [Team Member Name(s)]

These stakeholders participated in meetings and discussions, contributing valuable insights that shaped the analysis and recommendations outlined in this report.

3. ASSESSMENT AREAS

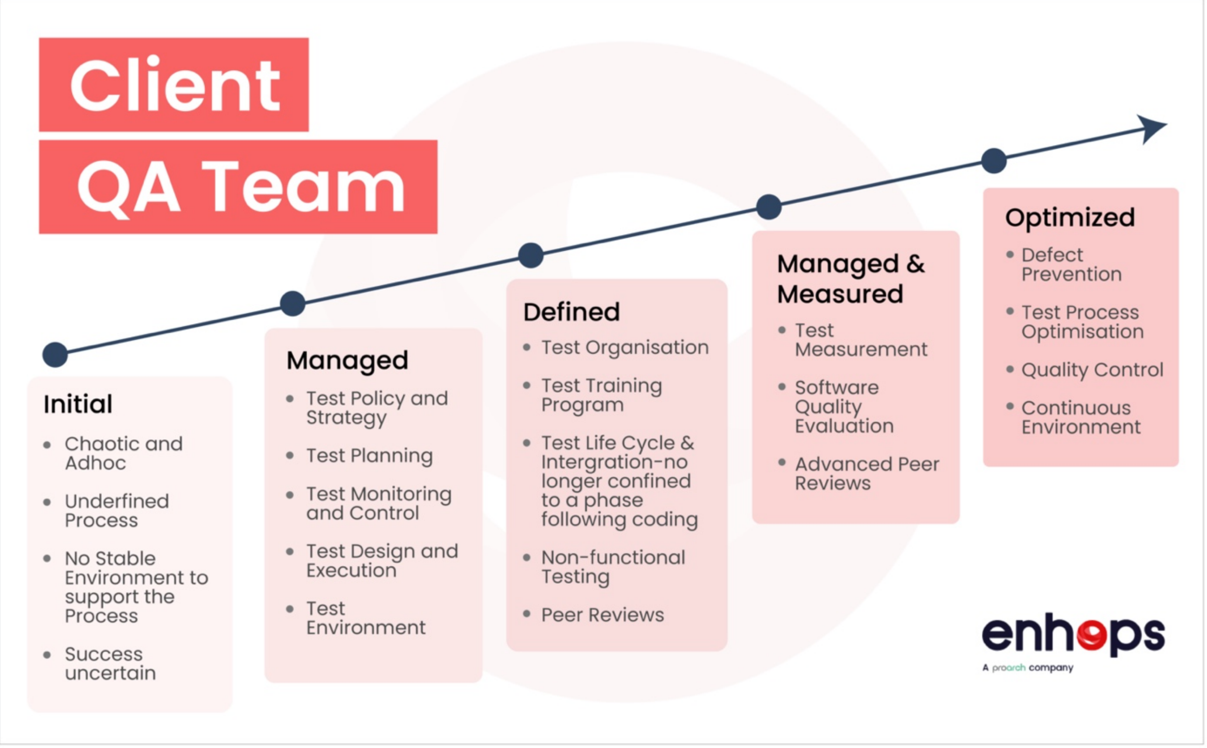
During the Discovery Phase, the following key areas were assessed to evaluate the current state of the QA process and identify opportunities for improvement:

» Knowledge Management
» Agile Process
» Effort Estimation
» Requirement Management
» Test Case Management
» Test Execution Management
» Regression Testing
» Bug Management
» Production Support/Service Testing
» Test Automation
» Performance Testing
» Security Testing

4. PROCESS MATURITY LEVEL

The TMMi (Test Maturity Model Integration) framework has been referenced in assessing client's current testing processes. TMMi is widely recognized as a global standard for test process improvement and maturity assessment. By leveraging this model, organizations can benchmark their testing maturity and implement strategic recommendations that drive key benefits. These include reduced risk, lower costs, faster delivery times, and enhanced software quality that aligns with business objectives.

.



Below charts list down the current maturity level of different groups in respective areas. The points are calculated out of 5. **5 represents best as per market standards and 1 represents that the practice is not yet followed.**

- 1 (Initial) - Indicates present practice does not exist, poorly controlled and is reactive
- 2 (Managed) - Indicates present practice exists, but not well defined/implemented (managed)
- 3 (Defined) - Indicates present practice exists, defined and is proactive
- 4 (Managed & Measured) - Indicates present practice is measured & controlled
- 5 (Optimized) - Indicates present practice focus on continuous improvement and optimized.

5. ASSESSMENT RESULTS

This section provides an insight into what is good, improvements and recommendations for existing QA process areas.

Test Center of Excellence

5.1.1 Knowledge Management

Knowledge Management is an essential aspect of large enterprise businesses with software products to ensure proper methods and principles are followed so they benefit from the following:

- Cost overruns
- Quality
- Scaling (up & down) with ease of continuity
- Information sharing with an objective to benefit from best practices amongst agile teams etc.

The major objectives include:

1) Retain the knowledge of the software/hardware
2) Help new team members scale rapidly
3) Reference documentation for software/hardware in maintenance

Current State	Current state of your applications and processes to store knowledge management artifacts like BRS (Business Requirement Specification) documentation which would help testers gain functional knowledge.
Business Criticality	High
Improvements	The Knowledge Management is basic and limited to sharing the business requirements of the software. It needs more maturity and focus on: • SME (Subject Matter Expert): Training videos are useful for a new person to get up-to-speed and become a subject matter expert. • Attrition: In case of attrition, because of lack of thorough/detailed information, there might be loss of knowledge from the eco-system. • Reference-ability: It is quite common for people to miss some test cases (especially corner test cases) when the tester revisits the software for an enhancement or modification after few months/years. • Videos vs Documents
Primary Recommendations	Training Videos Beyond Videos Reverse KT
Secondary Recommendations	**SDLC Process** KT's/ Functional Trainings should be an integral part of the SDLC in the agile process. Otherwise, it is easy to de-prioritize the KT, lose the momentum of knowledge sharing, and it would impact the overall productivity because of the inability to flex.
Contribution to Business Outcome	A detailed and focused KT/Training Videos will help teams to understand how users interact with applications and what is high vs. low impact areas to help reduce critical bugs.
Rating	2/5

Agile Process

Agile is an iterative approach to software development that helps teams deliver value to the customers faster, with fewer risks and high quality. Agile Practices includes - Workflow visualization. Using agile boards make it simple and easy for agile teams to monitor tasks progress and identify bottlenecks.

Current State	Following release cycle. Each sprint has 2-week duration and sprint ceremonies are in place like Sprint Planning, Scrum call, Spring Review & Sprint Retrospective.
Business Criticality	High
Improvements	Sprint Retrospective meeting: Scope for improvement i.e. capturing improvements and plan for action items. Sprint Signoff criteria/process. SLAs (response time and resolution time) for bug fixes are to be well defined and strictly followed
Recommendations	Sprint retrospective meeting should be conducted by all teams without fail and the improvements. SLAs (response time and resolution time) for bug fixes should be well defined based on severity/priority of the bugs.
Contribution to Business Outcome	Sprint ceremonies like Sprint Retrospective play a major role when working Agile Cost Effective People Focused/Collaborative Reduced Risk
Rating	2/5

Effort Estimation

Effort Estimation is a key activity in test planning. A well-thought effort estimate helps to rightly plan the testing phases, allocate the required resources, schedule the testing activities so that the testing phase gets completed on-time.

Estimated effort is a practice in software testing where teams estimate the relative size of a task or product backlog item based on how much effort it will take to complete it.

Current State	Effort estimations are taken from Testers, Developers for all sprints/releases Point Based method (3,5,8,13 points) is used for estimations during Sprint Planning.
Business Criticality	High
Improvements	Requirement/User story effort estimates could use some improvements. Testers are sometimes spending more time than expected in test design/test execution.
Primary Recommendations	Estimations should be tracked to ensure that development is completed as per the estimations given. Testing estimations should be efficiently calculated which would results in effective plan and efficient test results.

Secondary Recommendations	Tasks and user stories should be mapped with the agreed due dates and regular checks should be done. It would help tester find time to do thorough testing which reduces the bug leakage
Contribution to Business Outcome	Effective effort estimation would help to do thorough testing, mitigate schedule slippage, reduce bug leakage
Rating	2/5

Requirement Management

Requirements lifecycle is focused on clearly defining a project's (and sprint) scope, so that project/scrum team understand what needs to happen to meet the desired end goal better. It provides a high-level understanding of business goals, and outlines what is needed for the project to be a success.

Requirements Management objectives include:

- Enhance understanding of stakeholder needs, requirements, expectations and the problem or opportunity the product means to address.
- Gain clarity on scope, budget, and schedule.
- Minimize costly, time-consuming rework.
- Increase product quality.
- Mitigate risk

Current State	Requirements (/user stories) are being delivered by the business team. All requirements/user stories have detailed description, which is good
Business Criticality	High
Improvements	Detailed review (of requirements/stories) process requires improvement Requirement ambiguities could have been recorded/tracked separately.
Primary Recommendations	Requirements should be thoroughly reviewed by the project/scrum team (primarily testing team, business team and development team) to ensure requirements are complete and correct.
Secondary Recommendations	Monitor and control requirement bugs release by release
Contribution to Business Outcome	Bugs will be reduced early in development cycle leading to less testing rework and better speed to market.
Rating	2/5

Test Case Management

Test Case Management is the process of defining the scope of testing that needs to be performed on a system, writing the test cases that covers the scope. Test Case Management helps in following ways,

- It gives a clear idea of the testing activities to a testing team.
- Team will know what tests to execute and what to expect if the test succeeds or fails.

- It helps to keep track of test cases and group them into categories like designed, reviewed etc.,

Current State	Test cases are created and maintained Test Cases are attached to the appropriate stories/test tasks
Business Criticality	High
Improvements	Test case review - peer review, high level review with Development/Business team, is happening, but, not frequently/formally. Incorporating review comments needs to be done formally
Primary Recommendations	Test cases should be created and maintained Traceability should be build/maintained between test cases, stories Standard Test case template with traceability between requirement/user story, test case & bug) should be used across all projects.
Secondary Recommendations	Test Case review comments should be well documented and implemented Test Case cloning should be implemented for common functionality to save time and to perform effective integration testing
Contribution to Business Outcome	Guaranteed good test coverage. Reduced maintenance and software support costs. Reusable test cases.
Rating	2/5

5.1.2 Test Execution Management

Test Execution Management is the process of executing the test cases that cover the scope, managing the testers needed for testing and scheduling the testing activities to meet the release date.

Test Execution Management is important because,

- It minimizes the risks associated with the behavior of the system in a particular environment
- It is responsible for deciding the readiness of the software product, for its entry into the market.
- If the results of this execution are similar to the expected or desired results, the software product is considered ready for the release in the market.

Current State	Test Cases are executed, and the status is updated
Business Criticality	High
Primary Recommendations	Pair Testing is recommended to ensure requirements/stories are developed as per the acceptance criteria and to minimize the obvious bugs early.
Secondary Recommendations	Shuffling of requirements/stories among testers during test execution helps to minimize the bug leakage.

Contribution to Business Outcome	Improve Product Quality
Rating	2/5

5.1.3 Bug Management

Bug management is the process of reporting the bugs and tracking the progress of bugs from discovery to resolution. The Bug Management practice focuses on collecting, recording, and analyzing software bugs and enriching them with information to drive metrics-based decisions.

The practice's first stream deals with the process of handling and managing bugs to ensure released software has a given assurance level.

Current State	Bugs are identified during test execution and logged/linked to user story in JIRA
Business Criticality	High
Improvements	Bug life cycle/Bug workflow is to be streamlined as per Industry standards Severity, Priority for bugs & guidelines could have been better documented
Primary Recommendations	Bug life cycle/Bug Workflow should be streamlined & strictly followed across all teams
Secondary Recommendations	Sign off Criteria should be strictly followed across all teams Bug Summary report (customized report template or JIRA report) is recommended to be published at end of every sprint/release
Contribution to Business Outcome	Reduces development team and testing team efforts for the bug fixing/verification as well as analyzing the root cause of bugs found during pre-release and post-release phases.
Rating	2/5

5.1.4 Production Support/Service Testing

Production Support/Service Testing is testing production/customer tickets that are identified in Production environment/at Field by real users.

Current State	Service/Support tickets are created and assigned to Dev team
Business Criticality	High
Improvements	Analysis of service/support tickets and identification of impacted areas is limited Tester could have been involved in analyzing customer tickets, sharing reproducible steps to developers

Primary Recommendations	Tester should analyze the customer tickets, derive impacted areas along with sharing reproducible steps to Developers Tester should test impacted areas before the ticket fix is pushed to Production
Contribution to Business Outcome	Improve customer satisfaction
Rating	2/5

5.1.5 Test Automation

Test automation is the process of using automation tools to develop automation scripts, maintain test data, execute tests, and analyze test results to improve software quality. When executed well, it relieves much of the manual testing efforts.

Current State	Automation is to be done
Business Criticality	High
Improvements	Resources with required automation skill set is to be allocated Separate test suite should be available for test automation
Primary Recommendations	Test automation team with required skills should be deployed Automation test cases should be identified and prioritized Sanity test cases should be automated first
Secondary Recommendations	Separate automation suites should be created for Sanity testing, Regression testing End-end business workflows should be automated for user acceptance testing
Contribution to Business Outcome	Reduce the overall manual testing effort Reduce time to execute required set of test cases during regression testing Help testing team to run the regression suite for all releases
Rating	1/5

5.1.6 Performance Testing

Performance testing is a non-functional software testing technique that determines how the stability, speed, scalability, and responsiveness of an application holds up under a given workload.

Current State	There is no formal performance testing in place
Business Criticality	High
Primary Recommendations	Automate the performance testing for long-running tasks that has a possibility of degrading overall application performance
Secondary Recommendations	The benchmark values should be revised periodically to improve overall user experience
Contribution to Business Outcome	Performance bottlenecks will be identified & fixed before they surface in production
Rating	1/5