

# Product Owner in Test

Srilu Balla (Sridevi Sabarinath Balla)

<http://www.linkedin.com/in/SriluBalla>

## Abstract

In the ever-evolving landscape of software development, traditional Quality Assurance (QA) practices have become increasingly insufficient, often relegated to narrow scopes within sprint-specific tickets. This constrained approach hampers comprehensive product evaluation and undermines the potential for holistic quality enhancement. Moreover, prevailing methodologies like Agile fail to adequately integrate the QA role, leading to misconceptions of QA professions such as:

1. **Agile Challenges:** Agile methodologies do not recognize the QA role in their frameworks.
2. **Limited Scope:** QA teams are frequently restricted to testing tasks defined in tickets, which do not encompass the entire product or its integration with other systems.
3. **Lack of Empowerment:** Organizations ignore the recommendations made by QA professionals.
4. **Scapegoat:** QA is typically only remembered when something goes wrong.
5. **Bug Finders:** QA is perceived merely as 'bug finders'; some QA teams function as such too.
6. **Final Step:** QA is considered the 'final step in the process'. Not understanding their needs.
7. **Automation Overreliance:** Automating is perceived to be superior to a human testing a product.
8. **Process Confusion:** There is often confusion between testing and acceptance criteria checking
9. **Lack of Standardization:** Bugs and test cases created by one person are hard for others to understand and utilize.

This paper introduces the role of the "Product Owner in Test" (POT), a transformative position designed to revitalize and expand the scope of quality practices in software development. By redefining titles—such as reimagining QA as "Quality Assistant"—and introducing new responsibilities and frameworks, the POT role seeks to address longstanding challenges. Key proposals include:

- **Terminology Evolution:** Reframing "Test Cases" as "Check Cases" to distinguish between exploratory testing and pass/fail validation, and viewing "Bugs" as opportunities for improvement rather than failures.
- **Expanded Roles:** Establishing positions like Product Owners in Test (POTs), Quality Assistants, and Software Developers in Test (SDETs) to ensure a comprehensive approach to product quality.
- **Broadened Testing Scope:** Advocating for an evaluation of the entire product ecosystem, beyond the confines of individual tickets.
- **Standardization and Certification:** Introducing certifications such as Certified Product Owner in Test (CPOT), Scaled CPOT (SCPOT), Certified Issue Reporting Professional (CIRP), and Certified Check-Case Design Professional (CCDP), accompanied by a robust code of conduct.

## Biography

*Sridevi "Srilu" Balla has over 20 years of experience as a Quality Assistant, working in various roles including Quality Professional, Agile Coach, and Product Manager. She developed the Interface Agnostic Automation Framework, which she presented at the PNSQC in 2011 and 2012. Additionally, she presented on the test case life cycle in 2015. Affectionately known as "the American in the Saree," Srilu combines her love for America with her passion for traditional sarees.*

# 1 Introduction

Quality Assurance (QA) has long been an essential part of the software development process. However, much like processes such as Scrum and Agile, QA practices are directly derived from manufacturing methodologies.

In manufacturing, the production process begins after identifying specific needs and requirements. Following research and development, prototypes are tested by beta customers. Once the product enters the manufacturing phase, the Quality Assurance (QA) team steps in with a set of requirements to evaluate the product based on a documented checklist.

In manufacturing, the primary focus is on the product. An entire role, Quality Control, was created specifically to check the product that has been produced, even if the formula has not changed in years. The expectations in manufacturing are very clear and well-defined.

Software development, on the other hand, is unique with every feature, element, integration, and schema. Unlike manufacturing, where products are mass-produced in batches with proven requirements, software is developed one time only, often without extensive R&D or prototyping. Consequently, traditional manufacturing processes have been inadequate for software development for decades.

## 2 New Role and Vocabulary

Repeating the manufacturing life cycle steps in the Software Development Life Cycle (SDLC) will not lead to improved outcomes. To address the gaps and shortcomings in applying manufacturing processes to software development, we should have a unique process to better suit the software development context.

### 2.1 Product Owner in Test

Similar to a product owner who is empowered to add value and make independent decisions, a **Product Owner in Test (POT)** will learn, explain, and reflect on the product. They will own the product's quality and integrity. Teams must collaborate with the POT to include, enhance, or remove any features at any stage of the product life cycle. (Qualifications, Responsibilities listed ahead). One POT per Product. POT works independent of an engineering team.

#### 2.1.1 Quality Assistant

Quality Assurance (QA) is often misconstrued as a guarantee of perfection. The role may be accurately described as Quality Assistance. As a **Quality Assistant (QA)**, our focus is on bettering the product rather than guaranteeing specific outcomes. QA works within an engineering team. QAs are skilled professionals who excel in writing standardized test cases and logging detailed bug reports.

While the POT focuses on the broader, strategic aspects of product quality, the Quality Assistants will continue to be invaluable in handling the day-to-day execution of test cases and maintaining the operational aspects of quality assurance. This redefined role allows for a more efficient division of responsibilities, ensuring that both strategic oversight and meticulous execution are covered within the development process.

#### 2.1.2 Scaled Product Owner in Test

The introduction of a Scaled Product Owner in Test (Scaled POT) becomes essential as organizations adopt frameworks like SAFe (Scaled Agile Framework) and Scaled Scrum Teams. In this role, the Scaled POT will coordinate with various Scrum teams to ensure that quality assurance practices are aligned across the board. They will be responsible for integrating testing strategies into the broader agile framework,

ensuring that all teams adhere to standardized processes while still addressing the unique needs of each project. This includes overseeing cross-team dependencies, facilitating communication between teams, and ensuring that all products meet the high standards required for successful integration and release.

## 2.2 Testing

Testing is about understanding HOW something works. The only true form of 'testing' is exploratory testing, which is performed by humans, with or without the aid of tools. Unlike checking, testing can extend beyond the requirement scope. It uncovers dependencies, issues, inconsistencies in features, logic, UI/UX, ADA functionality, vulnerabilities, and much more.

### 2.2.1 Checking

Any process focused on PASS/FAIL criteria is checking, not testing. Unit Testing, Integration Testing, System Testing, Acceptance Testing, Performance Testing, Security Testing, Usability Testing, Compatibility Testing, Regression Testing, Smoke Testing, Alpha Testing, and Beta Testing are primarily used to check the known state of a system.

### 2.2.2 Check Cases

Testing reveals how a feature works. Once this understanding is achieved, reliable and repeatable steps can be established. These steps will be the **Check Cases**, which will help verify the application's state.

### 2.2.3 Automated Checking

Automation is a tool that executes tasks as instructed by a human. It cannot perform testing on its own; it can only check for the specific state of the system the code was written for. Therefore, it should be referred to as Automated Checking.

## 3 The Role and Responsibilities of Product Owner in Test

The Product Owner in Test (POT) is a crucial role within an organization, focusing solely on the product's quality and functionality. Unlike traditional QA roles embedded within scrum teams, the POT operates independently, ensuring that the product meets the highest standards before release.

### 3.1 Organizational Level Responsibilities

The Product Owner in Test (POT) plays a critical role in advocating for and implementing high-quality standards across the organization. By focusing on quality at the top level, the POT helps reduce issues throughout the process and products, leading to optimized time management and resource utilization.

#### 3.1.1 Production Data Sanctity

- **Customer Data Protection:** Ensure that customer-sensitive data is never used in testing unless explicitly requested under special circumstances.
- **Data Scrubbing:** All customer and employee data must be thoroughly scrubbed if it is to be used in lower environments to maintain data privacy and security.

### 3.1.2 Standardizing Common Features

- **Feature Uniformity:** Standardize common functionalities such as email formats, password criteria, registration forms, and authentication processes across all applications.
- **Efficiency in Decision-Making:** By standardizing these features, the POT reduces the decision-making burden on teams, leading to fewer issues and decreased testing effort.

### 3.1.3 Standardizing Lower Environments

- **Clear Environment Designations:** Establish clear designations for environments, including Production, Staging/Demo, QA/Test, and DEV/Sandbox.
- **Organizational Uniformity:** Ensure that the usage and naming of these environments are consistent across the organization, allowing for easy adaptation by any team member

### 3.1.4 Standardizing Releases

- **Release Timelines:** Set up clear timelines for major releases and hotfixes to ensure consistency and predictability.
- **Naming Conventions:** Bring uniformity to naming conventions for fix versions, labels, and tagging processes for release tickets across all products.
- **Code Freeze Protocols:** Define code freeze timelines for all types of releases, ensuring that these practices are consistent across the organization.

### 3.1.5 Standardizing MVP 1 Requirements

- **Foundational Elements Setup:** Ensure that critical foundational elements, such as database setup, API setup, environment setup, and release processes, are established early in the development process.
- **Preventing Rushed Implementations:** Enforce these standards to avoid rushed or incomplete setups that can lead to complications later in the product lifecycle..

## 3.2 Product Level Responsibilities

The role of Quality Assurance (QA) has often been misunderstood and undervalued, with QA teams frequently being seen as mere "bug catchers" who are only brought into the process when something goes wrong. This approach limits the impact QA can have on delivering a high-quality product. To maximize the benefits of the testing process, organizations need to provide the right tools, resources, and responsibilities to the Product Owner in Test (POT). The following recommendations are designed to empower the POT to succeed in this role:

### 3.2.1 Testing the Product

The **Product Owner in Test (POT)** is responsible for overseeing a product in its entirety, ensuring that all aspects of its quality, functionality, and performance are thoroughly evaluated and optimized. This singular focus allows the POT to make informed, strategic decisions based on a deep understanding of the product's complexities and nuances. Key responsibilities include:

- **Comprehensive Testing:** Most of the POT's time should be dedicated to conducting thorough testing across all aspects of the product.
- **Product Overview:** The Product Owner should provide the POT with a thorough understanding of the product, going beyond just assigning tickets for testing.

- **Technical Walkthroughs:** Development leads, architects, and database administrators should provide the POT with detailed walkthroughs of unit tests, integration tests, component tests, and data rules.
- **Access to Environments:** The POT should have access to the product, database, API, and provisioned users or accounts in lower environments, or the knowledge to start testing independently.
- **Strategic Testing Initiation:** Testing should begin only after a significant portion of the product is developed.
- **Holistic Testing Approach:** Testing should be conducted with and independently of the requirements, ensuring a comprehensive understanding of the product.
- **Consistent Testing:** Testing should be a continuous, exploratory process rather than just checking bits and pieces from time to time.

### 3.2.2 Documenting the Product

- **Creating Documentation:** The POT should create detailed check cases that include data, API, and integration steps along with UI steps, document any issues that need resolution, and maintain wikis to help team members understand the product.
- **Clear Test Cases:** Test cases should be written in such a way that anyone who can speak English can execute them and understand the results. The steps should also help identify the root cause of issues.

### 3.2.3 Demonstrating the Product

- **Product Demo:** The POT should conduct a product demo with the engineering and product teams, with or without stakeholders, to showcase the product's features and functionality.
- **Execution of Test Cases:** During the demo, the POT should execute every test case to demonstrate the product's features inside and out.
- **Explaining Data Handling:** The POT should explain how data is stored, processed, and reused within the application.
- **Assessing Product Readiness:** This demo will help determine the product's readiness and overall health.

### 3.2.4 Recommendations and Delegating Tasks

- **Task Delegation:** For large applications or projects, the POT may delegate tasks to Quality Assistants but should always oversee the overall quality of the entire product.
- **Recommendations for Testing:** The POT may recommend additional testing types such as performance, security, or functional automation, and allow the team to make the decision.
- **Enhancing Testing Automation:** The POT should recommend specific unit, integration, component, and functional tests to be added to increase automation, thereby reducing the need for redundant manual checks.

## 3.3 Tools and Resources for POT Success

To ensure the **Product Owner in Test (POT)** is set up for success, the organization should provide the following essential tools and resources:

### 3.3.1 Access to the Product and Database

- **Full Product Access:** Ensure the POT has full access to the product in a clean, dedicated environment for thorough testing and quality assurance.

- **Database Access and Training:** Provide access to a functional database, along with the necessary training to allow the POT to create and manage data independently.
- **Codebase Access:** Grant direct access to the codebase, enabling the POT to gain a deep understanding of the product's inner workings and to verify aspects of the product as needed.

### 3.3.2 Comprehensive Information

- **List of Services and Applications:** Provide a detailed list of all services and applications involved in the product, giving the POT a complete picture of the product's ecosystem.
- **Provisioned Users or Accounts:** Ensure the POT has access to provisioned users or accounts specifically for testing purposes, enabling realistic and thorough testing scenarios.

### 3.3.3 Product Overview

**Product Walkthrough:** A comprehensive walkthrough of the product by the Product Owner or Project Manager is crucial. This should cover all features and functionalities, giving the POT a holistic understanding of the product.

## 3.4 Testing results from POT

### 3.4.1 Test Data Set-up

- **Standardized Test Data:** Establish standardized test data that is accessible and usable by the entire team across all environments, including development, QA, staging, and production.
- **Comprehensive Wikis:** In cases where test data cannot be pre-created, develop detailed wikis with step-by-step instructions for team members to generate their own test data.
- **Consistent Naming Convention:** Implement and enforce a uniform naming convention for all test data to ensure clarity and uniformity across all testing activities.

### 3.4.2 Check Cases

- **Designing Check Cases:** After thoroughly testing and understanding the product, the POT designs and documents product-based check cases.
- **Linking to Resources:** Check cases will include links to wikis where relevant data can be accessed or created.
- **Clear Execution Guidelines:** These check cases should be so well-defined that anyone reading them can easily execute them without needing additional guidance, ensuring consistency and accuracy in testing across the team.

### 3.4.3 Issues / Bugs

- **Nature of Bugs:** Bugs are an inevitable byproduct of testing. Some bugs require immediate attention, while others can be addressed later or may not need to be fixed at all.
- **Clear Issue Logging:** Issues should be logged clearly with detailed steps, properly tagged, and prioritized with input from the team.
- **Opportunities for Improvement:** Issues should be seen as opportunities for improvement, not as indicators of the application's failure. This mindset promotes continuous improvement and helps maintain a positive approach to quality assurance.

## 3.5 Product Release Responsibilities

Once testing is completed and check cases are written down and features have been signed off on, products may be promoted to environments – Stage, PROD. The POT will assist the team in a Go/No-go decision. POT will seldom make that decision.

### 3.5.1 Release Preparation

- **Advance Notification:** The POT ensures that all stakeholders are notified in advance of the release dates and the features targeted for release.
- **Comprehensive Product Check:** Before the release, the POT conducts a final check to ensure all aspects of the product are covered, verifying that the product is ready for deployment.
- **Documentation and Demonstration:** The POT documents any findings from the final checks and demonstrates the product to the team to assist in making a Go/No-Go decision.

### 3.5.2 Production Verification

- **Assisting in Verification:** The POT assists in verifying the product in the production environment to ensure a smooth release process.
- **Data Integrity:** It is crucial that no customer production data is used during testing without explicit consent, and the POT ensures compliance with this standard.

## 4 Qualifications and Skills Required

To successfully lead a product through to production requires a refined skill set. Here are some key competencies that are recommended for this role.

### 4.1 Five years of Experience

- **Minimum Experience:** A minimum of 5 years of experience in a Quality Assurance or Developer role.
- **Software Development Expertise:** Deep understanding of software development processes and methodologies.
- **Beneficial Experience:** Experience as a domain expert or product owner is beneficial, but not essential for this role.

### 4.2 Understanding Software

- **Software Architecture:** The POT must possess a comprehensive understanding of software architecture, design principles, and development practices.
- **Component interaction:** This includes in-depth knowledge of how various components of the software interact and function together.
- **Impact Awareness:** Additionally, they should have a clear understanding of the impact of code changes on the overall system, with a strong emphasis on the significance of data flow throughout the application.

### 4.3 Demonstrating Expertise

The POT should be capable of taking full ownership of the product and effectively demonstrating its functionality through various means such as demos, detailed check cases, documented issues, and comprehensive wikis.

#### 4.3.1 Product Demo - How it works

- **Execution and Demonstration:** The POT should be able to demo the product by executing it in front of the team as needed.
- **Comprehensive Presentation:** This includes demonstrating test cases, describing issues, presenting stage and production releases, and showcasing the product to stakeholders.
- **Stakeholder Understanding:** This hands-on approach ensures that everyone involved has a clear understanding of the product's current state and its readiness for release.

#### 4.3.2 Check Cases

- **Designing Check Cases:** The POT must excel at designing product-based check cases, created after thoroughly testing and understanding the product.
- **Clarity and Execution:** These check cases should be well-defined, ensuring that anyone reading them can easily execute them without additional guidance, thereby maintaining consistency and accuracy in testing across the team.

#### 4.3.3 Data Setup

- **Data Proficiency:** The POT must be proficient in setting up data for both current and future checking needs, ensuring all necessary scenarios are covered.
- **Comprehensive Wikis:** Additionally, they should maintain detailed wikis that guide the team on effectively using and creating data, providing a valuable resource for consistent and accurate testing practices across the organization.

#### 4.4 Programing Knowledge

- **Understanding Developers' Work:** By having a basic grasp of programming, the POT can better understand the challenges and approaches developers take.
- **Informed Testing Decisions:** A POT with programming knowledge can make meaningfully and effectively understand the application and its code.
- **Debugging Skills:** While the POT is not expected to be a full-fledged developer, having basic debugging skills is important. It enables the POT to collaborate more effectively with developers by offering insights into potential problems and assisting in resolving issues more efficiently.

## 5 Product Ownership in Test Across Multiple Products

A POT is focused on a single product, the role can be expanded to manage multiple products, especially in organizations that have a suite of related products or where the products are not highly complex.

### 5.1 Importance of Dedicated POT for Each Product

- **Clear Ownership:** Each product should have a dedicated POT to maintain clear ownership and responsibility. This ensures that there is no ambiguity in decision-making and that the product's quality assurance is guided by a single vision.
- **Avoiding Conflicts:** When multiple POTs are responsible for the same product, it can lead to conflicting decisions, inconsistent testing strategies, and potential delays. A single POT ensures that all decisions are aligned with the product's goals and quality standards.
- **Streamlined Communication:** A single POT per product simplifies communication channels with the development team, stakeholders, and other involved parties, leading to more efficient collaboration and quicker resolution of issues.



## 5.2 Challenges of Managing Multiple Products

- **Time Management:** A POT managing multiple products must be highly effective in time management to ensure that each product receives attention and that deadlines are met.
- **Balancing Priorities:** The POT must constantly balance the priorities of different products, which may have varying levels of complexity, risk, and importance to the organization.
- **Maintaining Quality:** The POT must ensure that the quality of each product does not suffer due to the divided focus. This requires a strategic approach to testing, automation, and continuous monitoring.

## 6 Not Responsibilities of a POT

Last but not least, it is important to delineate the boundaries of this role to avoid misunderstandings about what the POT is and is not responsible for. Certain aspects of the development process remain the collective responsibility of the entire team.

### 6.1 Not a Bug Finder

While the POT will inevitably identify issues during the testing process, it is crucial to understand that finding bugs is not the primary responsibility of this role. Bugs are a natural byproduct of the testing process. Their discovery is a shared responsibility across the entire team, including developers, testers, and other stakeholders. The POT's focus is on understanding and improving the product as a whole, not just on the identification of individual defects.

### 6.2 Not Responsible for Validation of Acceptance Criteria

Validation of acceptance criteria is a critical part of the Scrum process, but it is not the responsibility of the POT. This task falls squarely within the domain of the Scrum team, which includes developers, testers, and the Product Owner. The POT's role is more strategic, focusing on the broader picture of product quality rather than on the specifics of individual user stories or features.

### 6.3 Not responsible for Overall Quality of the Product

Ensuring the overall quality of the product is a team effort and cannot be the sole responsibility of the POT. Quality is a collective goal that requires input and collaboration from all team members, including developers, designers, testers, and the Product Owner. The POT's role is to provide insights and recommendations based on their testing, but the implementation of these recommendations is a shared responsibility.

### 6.4 Not Decision Maker of the Product

Decision-making regarding the product's direction, features, and priorities is the responsibility of the Product Owner and the Scrum team. The POT's role is advisory, providing feedback based on their testing experience and insights into the product's performance. However, the final decisions about what features to develop, modify, or remove rest with the Product Owner and the team.

### 6.5 Not Gatekeeping Releases

The POT is not a gatekeeper who has the authority to block or approve releases. Instead, their role is to provide valuable feedback and insights to help the team make informed go/no-go decisions. The POT may highlight potential risks or issues, but the decision to proceed with a release should be made collectively by the Product Owner, Scrum team, and other relevant stakeholders.

## 6.6 Not a Domain Expert

The POT does not need to be a domain expert; their primary focus is on software and how it functions with the given requirements, rather than deep subject matter expertise in the product's specific domain.

## 6.7 Not a QA manager

When the POT delegates tasks such as performance testing, security testing, and regression testing, it's important to understand that this does not equate to a QA managerial role. The POT's focus remains on ensuring the product's quality and integrity, and delegation is simply a means to efficiently allocate specialized tasks to those best equipped to handle them.

# 7 Conclusion

In today's complex and rapidly evolving software development landscape, the traditional roles and practices of Quality Assurance (QA) are no longer sufficient to meet the demands of ensuring product quality. The introduction of the Product Owner in Test (POT) role represents a significant shift towards a more integrated, holistic approach to product quality. By focusing on the overall functionality, integrity, and quality of the product, the POT ensures that testing goes beyond simply validating requirements to providing a deeper understanding of how the product truly works.

Throughout this paper, we explored the limitations of conventional QA practices and the need for new roles, such as the POT and Quality Assistant, to better align with the unique challenges of software development. We discussed how the POT's responsibilities, including comprehensive testing, documentation, product demonstration, and collaboration across teams, position them as a crucial figure in the software development process. Moreover, the introduction of the Scaled Product Owner in Test (Scaled POT) role addresses the complexities of larger organizations adopting frameworks like SAFe, ensuring consistent quality standards across multiple teams and projects.

The proposed changes in vocabulary, such as distinguishing between testing and checking, further reinforce the need for a paradigm shift in how we approach software quality. By redefining the responsibilities and expectations of QA roles, and embracing the POT's strategic focus, organizations can better navigate the intricacies of modern software development. This approach not only leads to the delivery of higher-quality, more reliable products but also fosters a culture of continuous improvement, collaboration, and excellence in the development process.

As we move forward, the **Product Owner in Test (POT)** will play a pivotal role in shaping the future of quality assurance in software development, ensuring that products are not only functional but also meet the highest standards of security, performance, and user experience.

