

# QA in an Agile Development Team

**Michelle Wu**

mwu@paraport.com

## **Abstract**

Our company has been using the Agile Development process for almost 10 years. Our QA team plays an important role in making the development process successful. During these 10 years, we have been experimenting with various processes and approaches to make our development process more efficient; some work well for us, some do not work at all, and some need to be modified to fit our needs. In this paper, I would like to discuss what role our QA team plays in different stages of the software development lifecycle. What are the challenges we have encountered and how we have resolved them?

## **Biography**

*Michelle Wu is Parametric's Master Software Quality Assurance (QA) Engineer. She provides oversight and leadership of the company QA department. She has over 25 years of working experience in software testing. Prior to joining Parametric in 2006, Michelle had worked in software testing on firmware, drivers, printing software, access point management software. She earned B.S. and M.S. in Computer Science from California State University, Chico.*

Copyright Michelle Wu Aug.24, 2018

# 1 Introduction

Parametric is a financial company that do wealth management. We build proprietary software for our business to manage daily business workflows. Our development team supports over 36 applications running with 67 services on 14 databases. We started using Agile Development process over 10 years ago. Our QA team has been playing an important role in making the development process successful. During these 10 years, we have been experimenting with various processes and approaches to make our development process more efficient; some work well for us, some do not work at all, and some need to be modified to fit our needs. In this paper, I would like to discuss what role our QA team plays in different stages of the software development lifecycle. What are the challenges we have encountered and how we have resolved them?

## 2 Development Team Overview

First I would like to give an overview of our development team structure. Our development team consists of four subdivisions. The responsibilities of the development team are split into four and distributed amongst these subdivided teams; Project Teams, Performance Team, Maintenance Team, and Tools Team. There are 4 Project Teams and they are responsible for the development of new features and applications to support company operations and workflows. The Performance Team is responsible for the improvement and optimization of our application performance to ensure smooth operation of all the features. The Maintenance Team is responsible for resolving production escalations and supporting minor feature backlog items. Finally, the Tools Team is responsible for development and deployment tools. These four subdivisions of the Development Team work in tandem to produce the most optimized and effective software applications and tools. With the exception of the tool team and one of the project teams, all of our development teams have one to two developers and only one QA. With the Agile development approach, we release new features and feature updates very frequently. Our development team has deployed over 250 feature updates to production in the year 2017. This makes the job of QA very challenging.

## 3 QA Lifecycle Overview

Before getting into the challenges, I will give a brief overview of the current QA workflow in our development team. Our QA workflow starts from the beginning of the product lifecycle. We begin with consulting the Business Analysis and Business Users to draw out acceptance criteria for the project. Based on the acceptance criteria, we develop test plans and test cases, followed by test cases execution, and user acceptance tests. Finally, we coordinate with the release team to release the project to production. QAs face different challenges in different phases of the QA lifecycle. In order to overcome and reduce the bad side effects of the challenges, we continuously adjust and modify our QA workflow to make the QA team more efficient. We will discuss each phase of the QA lifecycle in detail in the later section of this paper. For now, let us go over the challenges and explain how we adjusted the QA process and lifecycle to reduce the bad side effects of these challenges.

## **4 Challenges**

### **4.1 Frequent Releases**

As an Agile development shop, we release updates and new features frequently to production. Often times, we release updates for an application as frequent as a couple times each week. Because of that, the risk of breaking existing features becomes very high. In order to ensure that we do not break any existing features in a release, we have to run very extensive application regression tests. This places a massive overhead in the time and effort required to finish regression tests each time before the release. To address these challenges, we adjusted our project planning to get releases into smaller chunks. This will reduce the risk of breaking existing features and the impact of the release in production. Smaller releases means more frequent releases. However, this puts a large burden on the QA to regress the application features before every release. To resolve this, our QA team maintains a set of comprehensive regression test documents. We automate the regression tests as much as we can, specifically in the areas that would take a long time and effort to test manually. For example, data validation logic, mathematic calculation logic, etc. would be prime areas to automate. We also check developers' code commits to the source control to better understand what areas are being updated. This allows the QA to better plan regression tests to ensure that the updated areas are being tested sufficiently.

### **4.2 Developers are always ahead of QA in completing tasks**

Another challenge that QAs normally face is that developers are always ahead of the QA in completing tasks. QAs always get pushed at the end of the development cycle to finish testing new and existing features.

When we first employed the Agile Development methodology, we used the fixed iteration method, which was 2 weeks cycle. This was because we wanted to find our team velocity. In addition, by using fixed iteration development cycle, developers will not acquire any non-planned tasks before the end of the iteration, and subsequently, developers will not be too far ahead of QAs. However, we found out later that fixed iteration would only work well if we could accurately estimate how much time we needed to finish the tasks for the iteration. We often either overestimated or underestimated the time and effort for the iteration. The team either ran of tasks to do or could not finish all the tasks we planned for the iteration.

We adjusted our process to plan our tasks based on project features. We do not use fixed iteration. Instead, we release when features are ready to be released. But by having a dynamic release cycle, the problem of developers finishing tasks ahead of QAs came back to us. In order to narrow this gap, we integrated more automated tests in our code to test features. These automated integration tests are run after the project has been built. We have also adjusted our project task breakdown process to make every feature task testable. By having testable tasks, QAs can work in parallel with the developers to test the task as soon as the developers have finished implementing it. In addition, developers help our QAs demonstrate features to the business users and run regression tests. Since we have developed a set of comprehensive regression documents that the developers can just follow, the ability to run regression tests has been made vastly easier.

### **4.3 We are not building the right thing**

Building a feature that does not meet user expectations is a mistake commonly made in product development. In order to ensure that we develop the right feature that meets business user needs, the QA is involved very early in the product development lifecycle. We meet with the developers, business analysts, and business users to review test cases. Test case review meetings provide the opportunity for business users to communicate their expectations and business workflow to developers and QAs.

Through test case review, developers and QAs also communicate with each other how the features should work for the business users. And ensure that the test cases provide good feature coverage and the features are being tested correctly.

In addition to test case review sessions, we also demonstrate the features to business users as soon as the features are completed and tested. This allows development teams to get feedback from the business users and make necessary changes as early as possible. The further along the development cycle, the higher the cost is to make changes to the code. To further ensure the right features are built, QA conducts "Paired UAT" with the business users, which is a process in which the QA sits down with the business users to execute user's acceptance tests.

We also narrow the communication gap between us and the business users by co-locating with them. The development team, including QAs, sit right next to the business users. This allows QAs to demo new features to the business users and get feedback during the development process.

#### **4.4 Agile development emphasizes Test Driven Development (TDD) no QA is needed**

There is a myth that test automation is all that is required in software testing. While test automation is important in software testing, automation cannot completely replace manual testing. Manual testing is still needed for tests that are difficult and costly to automate. For example, end-to-end functional test.

End-to-end functional tests involve testing across applications and services. It is difficult to automate testing across applications as we need to pass data and commands between application projects.

## **5 QA Lifecycle at Parametric Portfolio Associates**

### **5.1 Review requested feature**

Our QAs work very closely with business analysts and business users (stake holders) to define acceptance criteria for feature requirements. We ensure that the acceptance criteria is clear and correct and that feature requirements are correctly translated to acceptance criteria.

### **5.2 Test cases development**

Based on the acceptance criteria, the development team goes through a task breakdown process to break down the projects into testable tasks. While developers are implementing the features, the QA develops test cases and reviews them with the developers. We also often involve business users in our test case review meetings to ensure that we are testing the features correctly and ensure that QAs and developers have a good understanding of feature workflow.

### **5.3 Feature testing and regression**

As developers finish a feature task, the QA will pick up the task and test the feature by running the test cases. The QA will also regress the feature with existing features of the application to ensure that the new feature does not break any existing features. This will offload the burden of running regression at the end of the project development cycle. Once the development team finishes implementing and testing a feature set, the QA will demo the features to the business users.

## **5.4 Demo**

### **5.4.1 Why do we demo?**

Demo is “Show and Tell”. Regardless of how well the requirement and acceptance criteria are written, it is difficult for business users to picture how exactly a feature works. Demo is always the best way to communicate how a feature will function to the business user. By using the feature during demo, the business user can visualize how the feature functions and verify if the feature fits their workflow and if the user interface is intuitive. Demo also allows development teams to gauge user feedback from the business user earlier. This will minimize the cost of changes, as changes in the later stage of the development cycle cost significantly more. Finally, by demoing features as soon as they are ready to be used to the business users, the business users will be more involved with development cycle, leading to better communication and more effective product development.

## **5.5 User acceptance testing**

User Acceptance Testing (UAT) plays an important role in ensuring that the development team built the correct product. The user acceptance testing phase takes place at the end of the development cycle after features have been demoed to the users. Developers will create a release candidate of the application and the QA will push the application to the test environment for business users to run tests on the application.

During feature demo to the users, the QA reviews acceptance criteria and test cases with the users as preparation for UAT. The QA also provides guidelines to the user in how to run tests and what they need to look for when running tests. UAT feedback, which can be bugs, feature updates, or new feature requests, are reported to the development team. These issues will be triaged and fixed accordingly. After the bugs or features are updated, updates will be pushed to the test environment for another round of testing and verification.

This cycle will continue until the user has verified that all the features are working as expected. At this point, users will give their approval to release the updates to production.

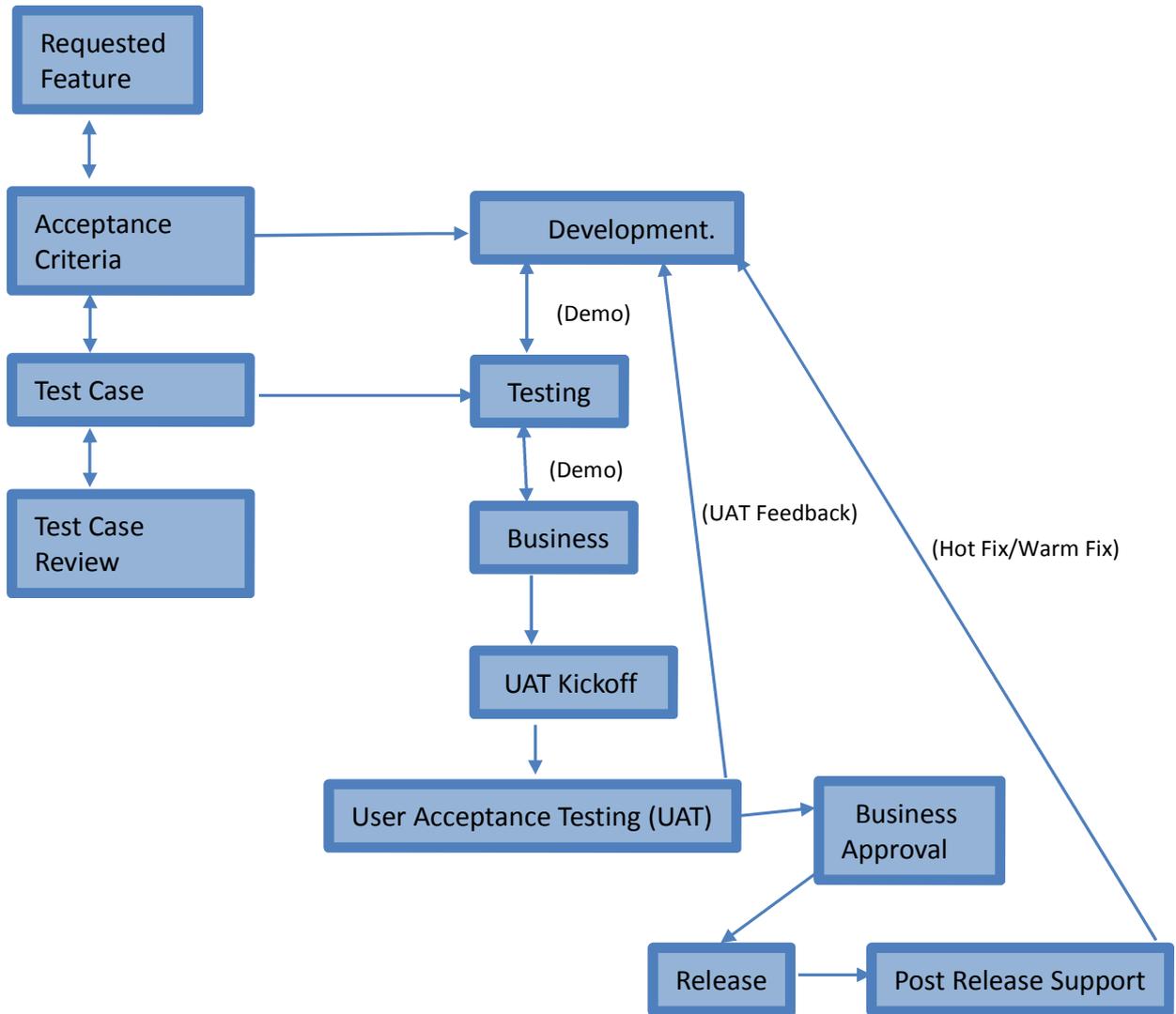
## **5.6 Prepare for release to production**

After getting business approval, the QA will run a final round of regression tests of the released application. After the regression tests have passed, the QA will submit release tickets to our application support ticket system. Then, the QA coordinates with the DevOps release team to prepare the release of the application to production.

## **5.7 Post release support**

The QA's responsibilities do not end with the release of the application. After the application release, we run demos for the Maintenance team and Application team so that they understand how the new features work and are able to support production requests and escalation of the release features. QAs will also continue to support the features if a warm fix (bug introduced with workaround) or hot fix (work stopping issue) is needed after features are released. Finally, QAs also update our application regression test document with the new features so that these feature will be tested in the next release of the application.

**Diagram 1 QA Lifecycle at Parametric**



## 6 Conclusion

As mentioned in the beginning of the paper, we started using Agile Development process over 10 years ago. Ten years ago, QA lifecycle at Parametric was limited to test planning, application testing and regression testing. In these 10 years, QA lifecycle has expanded to start at the beginning of the product development cycle, which is feature analysis with business users and business analysts; and end at providing post-release support.

QA works very closely with the business users and business analysts to derive acceptance criteria for the project. We conduct test case review sessions with the business users and developers to ensure features are being built correctly to meet business users' expectations. In order to narrow the gap that developers finish all the tasks before QA, we adjusted our project task breakdown process to make every feature task testable. By having testable tasks, QAs can work in parallel with the developers to test the task as soon as the developers have finished implementing it. In addition, we integrated more automated tests in our code to test features. We also develop sets of application regression test documents to allow developers help with regression testing.

To further ensure that we build the product features right, we demo the application features to the business users as soon as they are finished. QA also plays a very important role in conducting and coordinating the user acceptance testing with the business users. We communicate feedback from the business users to the development team. QA responsibility does not end after we finished testing the application features. We co-ordinate the release to the production. QA also provides post-release support to the application support team and the maintenance team.

Our development and QA team have been working very hard to make the Agile Development process work well for us. We attended conferences and workshops every year to bring new technologies, new knowledges about Agile Development back to our team. We constantly slim line our development and testing processes to improve efficiency and accuracy. We will continue strive for improving our QA processes to make our development team more efficiency,

## 7 Appendix

The following are the major tools our development and QA team use for project management, managing project documentations, such as technical requirements, acceptance criteria, test cases and regression test documents, and application deployment tools.

### 7.1 Project Management

- Leankit/Jira
- Confluence (project stories, test cases)

### 7.2 Automation

- Unit Tests
- Integration Tests ( Specflow)
- Postman (WebApi testing)

### 7.3 Logging

- Kibana
- Elk Stack/ Elasticsearch

### 7.4 Deployment

- TeamCity (Build Agent)
- Octopus