

2009

PACIFIC NW SOFTWARE  
QUALITY  
CONFERENCE



MOVING  
QUALITY  
FORWARD

OCTOBER 27-28, 2009

Conference Paper Excerpt

From the

CONFERENCE  
PROCEEDINGS

Permission to copy, without fee, all or part of this material, except copyrighted material as noted, is granted provided that the copies are not made or distributed for commercial use.

# **I HAVE TWO MANAGERS?! : ONE COMPANY'S MODEL FOR A CONSULTATIVE TESTING TEAM AND MATRIX MANAGEMENT**

Amy Yosowitz

Senior Information Technology Manager, Quality Assurance ■ Apollo Group, Inc.

(602) 387-7883 ■ amy.yosowitz@apollogrp.edu

## **ABSTRACT**

Apollo Group, Inc., parent company of University of Phoenix, faces an interesting challenge in providing homegrown, feature-rich, user-friendly software to a staff of thousands who serve more than 350,000 students. With 30+ integrated applications being continuously developed by almost as many development teams within the Business Systems Division (BSD) of Apollo IT, we have chosen to go the route of having a consultative testing team. Our Software Quality Analysts (SQAs) have dual identities. Each SQA is a member of a larger QA organization where individuals learn skills, exchange ideas and business knowledge with other testers, and receive mentoring and direction from QA management. Each SQA is also a highly valued member of a development team where they interact with their development lead, developers, and business contacts. SQAs receive their direction from both their QA manager as well as their development team lead.

This paradigm has resulted in well-rounded SQAs who are fulfilled in their careers and have a deep sense of ownership over the applications they work with. With multiple management escalation paths, quality of our products does not get shortchanged. Our division has little to no “throw the code over the wall to QA” mentality, and development leads each have dedicated testers that they trust in and collaborate with closely in their day-to-day tasks. QA management spends much of its time mentoring, coordinating system-level testing efforts, and propagating best practices among SQAs and development teams.

This paper reports on the many aspects of this model, including: the roles and responsibilities of development leads, QA managers, and SQAs within our IT division; the elements involved, such as meetings, interactions, goal-setting, and reviews; the benefits and advantages of being organized in this manner; and the challenges we face.

## **BIOGRAPHY**

Amy has over 10 years experience in the software field. She is currently a Senior Information Technology Manager for Quality Assurance at Apollo Group, Inc (the parent company of University of Phoenix), where she manages a group of 25+ software testers that focus on testing business applications. She was formerly a Senior Quality Assurance Engineer at Alogent Corporation (now Goldleaf Financial Solutions) and a Consultant at HBO & Company (now McKesson Corporation). During her tenure at Alogent, Amy was a lead tester on a state-of-the-art banking teller system. At HBO & Company, Amy performed implementation, testing, and design tasks for an innovative hospital information system. Amy holds a Bachelors of Arts degree in Mathematics with a French minor from Emory University in Atlanta, Georgia. She resides in Tempe, Arizona with her husband Rob and her toddler son, Blake.

## **BACKGROUND**

Apollo Group, Inc. is the umbrella organization for a number of educational institutions, including its largest subsidiary, University of Phoenix. University of Phoenix is an institution of higher learning geared toward the working adult and offers associates, bachelors, masters, and doctoral programs. University of Phoenix has ground campuses in almost every state in the United States as well as campuses abroad. One half of University of Phoenix students attend the Online campus, where coursework is done in an asynchronous fashion through online forums. Other Apollo institutions include the Institute for Professional Development, Center for Financial Planning, Western International University, and Meritus University. Apollo has over 350,000 active students, more than 400,000 alumni, approximately 25,000 faculty members, and approximately 20,000 staff members.

Apollo Information Technology (IT) is the department of Apollo that develops and tests both internal and external-facing software applications for Apollo's subsidiaries, as well as provides technical support, hardware support, database administration, network infrastructure, and other services to respond to all other technical needs of the company. The software area is split into three divisions: Business Systems Division (BSD) for most internal-facing software that the business employees use, Product Development Division for external-facing software that students and faculty use, and Finance/Human Resources (FHR) Division for finance and human resource applications.

This paper will focus on the software testing model used by the Business Systems Division (BSD) of Apollo IT. BSD has responsibility for over 40 different home-grown business applications. These are in varying stages of maturity; some are legacy applications, some are new applications developed within the past five years, some are brand new applications. All applications continue to receive enhancements and bug fixes on a regular basis to respond to the changing needs of our business. The development staff is split into small teams of typically ten or less developers, with each team taking responsibility for one or more closely-related applications. There is one development team lead ("dev lead") per team.

## **TESTING ORGANIZATION**

The testing team, often referred to as the Software Quality Analyst (SQA) team, works off of a consultative model and has a matrix management reporting structure. Our team currently has more than 25 SQAs who, from a human resources perspective, report up through to me, a senior QA manager. I have another QA manager reporting to me who helps to share the load of this large group.

It is my responsibility, along with my fellow QA manager, to assign permanent testing resources to each development team. It is in this way that we have a consultative model; much like a consulting firm provides the right resources for a job, QA management seeks out the right resources (as well as the right number of resources) to fulfill the specialized testing needs of each development team. It is important to note the differences between our model and another popular model for testing teams:

<b>Common Model</b>	<b>Apollo Model</b>
Pool of resources with generally similar product and testing knowledge	Pool of resources, each with varying testing experience and each with specialized product/business knowledge
Resources are assigned relatively short-term tasks, then when complete, are assigned to new testing tasks	Resources are each dedicated to a development team and perform all testing that team needs on an ongoing basis
QA management distributes tasks among SQAs	QA management mainly serves as a mentor and advocate for quality. SQAs, along with their respective dev leads, delineate the testing tasks that need to be done for a particular release cycle of each product.
SQAs all sit together in a central area	Each SQA sits embedded within the development team they serve
Some SQAs create tests, others execute those tests	Each SQA is responsible for both test case creation and execution for their product

When assigning an SQA permanently to a development team, several items are taken into consideration, including: communication style, business knowledge, technical skill, and the SQA's ability to work independently. Different development teams have different work styles and expectations of their testers, so QA management tries to make the best fit possible.

The other key element to our model is matrix management. The SQAs do all report up through me in the HR reporting structure, but through a "dotted line" in the reporting structure, they also report to the dev lead for their development team. The dev leads, who have overall responsibility for their respective application(s), are the ones that the SQAs interact with on a day-to-day basis. The SQA works with the dev lead (as well as the developers) on understanding requirements, creating estimates, defining the testing approach, test cases, troubleshooting, and raising system issues. The QA manager has a different relationship with the SQAs. The QA manager does more general oversight, ensuring that the SQAs are doing the right things to test their applications. SQAs are mentored by QA managers in areas of testing (including design and execution), technical skills, communication skills, system integration points, and time management. From the point of view of the SQA, the matrix management model provides the best of both worlds: a dev lead who knows the ins and outs of their particular product, and a QA manager with a higher level view of all of our systems.

Matrix management also has the benefit of providing multiple escalation paths for issues, which is always a difficult area in organizations where testers report directly to development managers. If a dev lead is having any sort of problem with their SQA, they can escalate to the QA manager, who has the expertise to mentor and work with the SQA. Development management typically does not have the mindset or the skill set to be able to help an SQA do their job effectively and efficiently. In addition, if an SQA feels the dev lead is repudiating the SQA's ideas and attempts to ensure a quality product (for example, "I know you haven't had enough time to test, but we need to send this to production anyway."), then the SQA can escalate to the QA manager, who can examine the situation and work with the dev lead to assess risk and arrive at the best possible outcome.

## **ROLES**

SQAs, QA managers, and development leads all have important roles in making this model work. Here are some further details about these roles and their responsibilities:

### SQA (Software Quality Analyst)

- Often functions as a "jack of all trades" – part tester, part business analyst, part developer. A well-rounded SQA has excellent analytical and logical thinking skill and therefore first-rate test case

creation skill. The SQA also understands Apollo's business model and is skilled at understanding and dissecting requirements, finding both intended and unintended consequences of new features. Lastly, the SQA has good technical skills with the ability to automate tests as well as write database queries to validate data and create and search for needed test data.

- Helps the dev team to understand the user perspective as well as the core business reasons behind software changes
- Decides, with support from the dev team, what testing is necessary for new changes and bug fixes. Compiles all test cases and test data.
- Maintains and executes manual and automated regression test scripts
- Creates and maintains automated smoke test scripts, used both in the QA environment as well as for production release validation
- Logs bug reports and helps the dev lead to prioritize any issues found and determine any "must fix" items for a particular release
- Makes recommendations to the dev lead as to whether a release is ready to go to production (or not), but does not have the final decision
- Generally has responsibility to protect the production environment
- Often has responsibility to organize, script, and/or perform product demonstrations for business users
- Calculates Defect Detection Rate (DDR) metrics on a regular basis
- Directly observes end users at least a few hours per month

#### QA Manager (Quality Assurance Manager)

- Mentors SQAs in areas of testing (including design and execution), technical skills, communication skills, system integration points, and time management
- Helps SQAs reach beyond the everyday to provide the most value to the organization
- Mediates conflicts between SQAs and development
- Serves as the advocate for the SQA and for quality software in general
- Performs all administrative tasks for SQAs, including: performance reviews, access approvals, hiring, and informing regarding company news
- Addresses any testing or SQA concerns the dev lead raises
- Organizes training and mentoring opportunities
- Manages the SQA team to work toward larger initiatives like quarterly goals
- Coordinates testing large IT projects, such as database moves, upgrades, and implementations for new user bases
- Shares knowledge between development teams. Each dev lead is given quite a bit of freedom to organize and run their team as they think is best. Because of QA management's exposure to many team's practices and methods of solving problems, they are in a valuable position to spread best practices, process improvements, and problem solutions among various development teams.
- Understands the high-level system architecture, data sharing between applications, and dependencies between products. Upon hearing of new features or new types of data interactions in one application, interprets these items and makes recommendations as to what other applications may need testing in conjunction.

#### Dev Lead (Development Team Lead)

- Serves as the leader and single point of contact for an application or a small group of related applications
- Has ultimate accountability for the quality and success of their application(s)
- Represents the application to the corresponding business unit(s)

- Manages and organizes developers and their work
- Ensures adherence to processes and standards that the team uses, including coding standards, software development lifecycle activities, and other best practices
- Interacts with the SQA on day-to-day activities
- Prioritizes, along with the SQA, the testing of new features, bug fixes, and regression tests, as well as prioritization within each of those areas, to be in line with risks, business priorities, and various uses of the application.
- Regularly reviews regression test cases for coverage and depth
- Makes the final go/no go decision on production releases, using information provided by the SQA

## CORE ELEMENTS

There are some principal practices that Business Systems Division of Apollo IT follows in order for this model to be successful.

- The QA manager and each SQA have a monthly one-on-one meeting for one hour.
  - This discussion can include: updates on how their dev team is functioning, follow up on goals and testing initiatives, any obstacles the tester is facing, professional development plans, review of test cases and test executions, metrics, upcoming releases and functionality being added, participation in system test efforts. It is integral to have this one-on-one time; with the sheer number of SQAs sometimes it is the only time these individuals can talk one-on-one.
- The QA manager and each dev lead have a monthly one-on-one meeting for 30 minutes.
  - This discussion can include: testing, process, and general SDLC (Software Development Life Cycle) challenges, upcoming functionality being implemented, a review of the SQA's job performance – where they are succeeding and areas that need improvement, suggestions as to how the QA manager and the SQA can better help that team to succeed, upcoming multi-application initiatives.

→Both of these one-on-ones help the QA manager to have a high-level view of all systems and to be aware of changes in individual applications. The QA manager can then identify both software and data dependencies and pull in multiple testers to collaborate in testing when needed.

- The SQAs and their respective dev leads are encouraged, but not required, to have a one-on-one monthly.
  - This closes the SQA→dev lead→QA manager communication loop and allows the dev lead and SQA to discuss things that may fall outside of normal day-to-day discussions, such as overall test strategy and automation progress.
- Each development team, including developers, SQAs, and the dev lead, has a daily stand-up meeting for 15 minutes per day.
  - This goes along with agile best practices, and allows the team to track progress and issues from day to day. We have found it quite essential for good team communication. Teams that forgo this practice, even for a short while, note that they feel out of touch with each other.
- All BSD SQAs and QA managers have a mandatory weekly team meeting for one and one-half hours per week (or less time, as needed).
  - These meetings typically start out with announcements, which can include:
    - personnel changes (SQAs coming, going, or switching development teams)
    - company news
    - IT department news
    - a “thank you” session where managers thank SQAs who have helped out in a special or extraordinary way, as well as SQAs thank each other publicly for assistance, mentoring, or a good job on collaborative testing

- a “bug of the week” session where SQAs bring up a unique bug that was found that week. It could be a serious bug that impacted our business, a trivial bug that was funny, a bug that was found by accident, a bug found using an interesting testing methodology – anything of note to other testers. We discuss how the bug was found, the impact, and how the dev team dealt with it. We encourage an open forum where SQAs admit their mistakes and share learning experiences with other SQAs.
  - There is also a general topic for each meeting. These can include:
    - An SQA presenting on the application that they test. This allows each team member to obtain knowledge on applications other than their own. Since our applications are all intertwined, this is important for testing purposes. It also helps everyone understand our business better.
    - A technical presentation on HP Quick Test Professional (QTP) or HP LoadRunner (LR)
    - A team building exercise or fun activity
    - Detailed review of a best practice or a specific SQA responsibility, such as metrics gathering or bug reporting
    - An SQA presenting on something learned from a recent conference
    - A focus on another area of IT, such as business intelligence, data warehousing, external-facing applications, support, or database testing
- All SQAs in all divisions and all SQEs (Software Quality Engineers, more technical testers) have an All SQA/SQE meeting for one hour each month, led by the director over QA. Topics are similar to those of the SQA weekly meetings.
- Most development teams are on two-week release cycles. This means that a new software version, with new functionality and bug fixes, is placed into production every two weeks. Development teams do this in one of two ways, depending on the nature of the application:
  - Overlapping cycles – A release goes to production. The next day, the SQA receives a build of the next version of the application to begin testing. While the SQA is testing this version for two weeks, some developers are dedicated to fixing bugs in that version, and some are already coding for the next version in a different branch. The version that the SQA is testing goes to production after two weeks of testing, then the cycle repeats.
  - One week/one week cycles – A release goes to production. The developers spend the next week coding for the next version of the application. While this coding is going on for a week, the SQA is catching up on automated scripting, understanding upcoming business requirements, adding to the regression tests, and doing spot tests on developers’ builds. One week after the previous release, the SQA receives a build of this new version and has one week to test it. Typically, most developers are focused on bug fixing during this week. The version that the SQA is testing goes to production after one week of testing, then the cycle repeats.
- The QA manager performs reviews on each SQA in the form of semi-annual and annual reviews. For both types of reviews, the QA manager obtains extensive feedback from the dev lead that the SQA works with to create the review. For the annual review, feedback from peers (fellow developers, SQAs, management, and sometimes business representatives that this SQA works closely with) is also solicited in the form of an anonymous peer feedback sheet. At each review point, general and professional development goals are set for the SQA including both project-based and personal career goals. Goal setting is a collaborative exercise between the SQA, QA manager, and dev manager.
- Each SQA gathers two flavors of metrics for each release after it has been through QA and also completed its production run (i.e. it has been replaced with another version in production):

- Defect Detection Rate metrics (DDRs) measure the number of defects found in the QA portion of the lifecycle (pre-production) compared to the total number of defects found in that release (including defects found post-production). These are also broken down by severity to better interpret the numbers. This gives the SQA, dev lead, and QA manager a way to measure how well the SQA did at protecting the production environment from significant defects.
- Day-by-day metrics measure the number of defects found each day a version is in QA and in production. It is a goal of all SQAs to provide information about the quality of a build as early on in the cycle as possible, so that the development team has adequate time to react and fix things. This measures how quickly SQAs were able to provide feedback to their dev teams.

→Both types of metrics often trigger interesting discussions of both QA and development practices within a team, whether it is the need for more unit tests, better prioritization of testing, more regression test cases, or development strategies for lowering the total number of bugs found each release.

- The SQAs sit among their development team. This greatly aids in communication, and the developers think of the SQA as a member of their team and not an unknown entity that they throw code over the wall to.
- Most SQAs attend all business meetings that the dev lead has with their corresponding business unit. This allows the SQA to see the decision-making process and ensure the final software changes do indeed fit the users' needs and intents. SQAs also often serve an important business analyst type role in these meetings, translating items between the developers and the business users.
- SQAs do not make the final decision as to whether a build goes to production; this is a decision left to the dev lead, as they are ultimately responsible for the success or failure of the application and its quality. SQAs do have the important responsibility of making a recommendation as to whether the build should go to production or not, and have to gather supporting data to back up this recommendation.
- Business Systems Division (BSD) has a general practice of moving employees between products every few years. This is especially true in the case of the QA department. QA management feels that after two or more years testing the same application, even the best tester can have blinders on to important or interesting bugs. Also, the tester is no longer challenged as much as they should be. In addition, having an SQA being a product expert for too long can result in too much of a safety net for the development team; developers may start taking shortcuts or using poor development practices because they figure the experienced, expert SQA will catch anything they mess up. Sometimes SQA swaps are initiated by the SQAs themselves who want to test something different and/or learn a new part of the business; sometimes they are orchestrated by QA management among two or more people we think need a new challenge. Transitions between testers usually occur over the course of a few months. While sometimes transitions are tricky, in the long run, they typically benefit both development teams. Each dev team gets an SQA with a fresh set of eyes on their application, as well as someone who understands other parts of our business. SQAs also know what development cycle practices have and have not worked on their previous team, and they can bring this experience and some new ideas to their new team.
- The BSD directors define goals each quarter for both development and QA that are good practices that can get left behind in the day-to-day work of building software. Recent quarterly goals for QA have included: ensuring all test executions are done within Quality Center, building system level tests for business processes that involve multiple applications, and improving production application monitoring.

- For most of our applications, we try to stick to a ratio of one SQA for every three application developers on a team.

## MISCELLANEOUS ELEMENTS

Here are some other elements of our environment that are not all directly related to testing, but important to note:

- We have a team of application Software Quality Engineers (SQEs). These are more technical testers that perform static code analysis, load testing, continuous integration tasks, and other white box testing techniques. They use a similar consultative testing and matrix management model to that of the SQA team.
- We have a team of database SQEs. These are testers that perform testing on database code including stored procedures, triggers, packages, and data transfers. They have a slightly different model of serving their customers, but it still can be considered matrix management and consultative testing.
- We have a team of database developers that are separate from application developers. Many applications share databases with other applications. These developers are somewhat independent from the development teams, but obviously have a need to frequently interact and collaborate with application developers.
- We have a team of Human-Computer Interaction (HCI) specialists. These individuals help produce easy-to-use and intuitive user interfaces for all applications. They are also set up in a somewhat consultative manner, although they do not generally follow the matrix management model.
- With a few exceptions, we do not have traditional business analysts for each business unit or application. In many ways, the dev lead and the SQA perform the business analysis tasks for a team. This has benefits and disadvantages. It puts the development team much closer to its business and causes our applications to align quite well with business needs, but it also puts an extra workload on dev leads and SQAs who have other primary focuses.
- We follow a mix of various agile methodologies.

## BENEFITS AND ADVANTAGES

BSD enjoys many benefits of this consultative testing team and matrix management model:

- Each SQA has a great sense of ownership, loyalty, and belonging to their development team and their product. This results in devoted SQAs who always go the extra mile to ensure the quality of the software and make sure it fits the users' needs.
- SQAs are the subject matter experts and are very much go-to people on their respective development teams. They greatly enjoy being able to give this amount of value and get a lot of fulfillment out of their job.
- SQAs are typically the "right hand man" for their development lead. The dev leads have a high sense of trust in their SQAs, and often depend on them to help lead the team when the dev lead is out of the office. This is wonderful when compared to IT organizations where development sees QA as a roadblock to their success. In Apollo IT, it is just the opposite – QA is an essential depended-upon part of the organization.
- The SQAs get a great variety in their jobs: the logic/analysis/test case creation aspect, the technology/automation/database aspect, and the interaction with business and requirements. This makes their job a lot less boring than if they just wrote and/or executed test cases all day.

- Quality does not get shortchanged due to the multiple escalation paths SQAs have (QA and dev management) for getting issues addressed. This is an advantage over a setup where QA reports directly to development.
- With SQAs assigned to work with a team permanently, the team develops relationships and gets the chance to really “gel.” There is little to no “throw the code over the wall” mentality. New employees often find this remarkable.
- Having typically one to three SQAs work for a single small development team helps with determining accountability for testing errors on a particular product.
- We have very few examples of contentious relationships between QA and development.
- As a consultative testing team, we feel having dedicated SQAs provide their services to development teams provides better customer service and visibility to our development leads. This is an advantage over having a centralized QA team where tasks are distributed.

## CHALLENGES

Any organization that uses this consultative testing and matrix management approach will face some difficult aspects to manage.

- While we generally like to allow development teams and SQAs to determine their own best and most efficient ways of doing things, some items must be standardized. This can include standardization of test case documentation, automation scripting standards, and even a standardized telecommuting policy. With each SQA as part of a different development team with different practices and conventions, it makes it difficult to standardize on most things. QA management also finds it difficult to enforce standardization with such a diverse team.
- QA management also finds it difficult in our model to monitor the productivity of the SQAs. Some of this is related to difficulties with standardization. With so many applications and typically only one or two SQAs per application, it is hard to determine if the SQA is slow or fast, efficient or inefficient, productive or unproductive with the time that they have to test.
- In our environment, it is just about impossible for one person to have in-depth product knowledge and business knowledge in all areas. In many cases, SQAs are experts in their own area but do not know about other areas, especially the upstream or downstream applications related to their own application. We encourage informal knowledge sharing as well as formal knowledge sharing through application presentations and organized system tests. In the end though, getting each SQA to know their application as well as other applications in their same business area and/or ones that theirs shares data with is often challenging.
- With typically one SQA per application, we face many challenges when SQAs take vacations or are out of the office unexpectedly. There is no natural person who can perform their responsibilities to their level of expertise in situations like this. Management has set up a system of application backups, where each SQA is a backup for one or more applications. When the main SQA for an application is out of the office, their backup is there to help their development team with testing efforts. This is by no means an ideal solution. When someone is out of the office, their backup must split their time between their main application and their backup application. This requires both development teams to slow down their development somewhat since neither has a full-time SQA during the period. Also, the development team has a difficult time putting their trust in a backup SQA that they do not know and that does not know their application nearly as well as their regular SQA. In addition, it is a challenge to make sure backups stay current on the features of the applications that they back up.
- With SQAs each specializing in one product or a group of small products, they can tend to have blinders on and focus their testing only on their application. Apollo often has business functions that

span multiple applications, and up until recently, these rarely got tested as a system. We have started putting together system tests for these important functions. All SQAs that work with applications involved with this business function meet over a period of months to design both a manual and automated test that covers the main scenarios and business workflows. This has been a slow but successful effort so far.

- We have recently faced an issue where, despite our best time management efforts, the SQAs are not able to keep the automated regression test suite up to date. We have fallen behind in continuously creating automated regression tests for new features, and many existing automated tests have been broken for quite a while. With the constant push to get new releases out every two weeks, automation ends up being deprioritized. It is important, but not urgent. We are considering changing our model so that we have a separate small team of automation specialists that write and maintain automated regression tests for all of our applications. We are also looking toward keyword-driven automation as a methodology that could work in this new model.

## **CONCLUSION**

Using this consultative testing and matrix management model has been very successful for Apollo IT's Business Systems Division. Our SQAs feel very valued and fulfilled in their jobs, and as a result we have very little turnover. Development team leads love the fact that they have a dedicated tester that they can always depend on. We have an organization that from the top down values quality and intuitively easy-to-use software applications that scale with our growing business. The Software Quality Analysts at Apollo Group are excited to take part in creating innovative software that fulfills our business goal of serving students well.