The Importance of a Solid Quality Assurance Effort for a Successful Project

The most exciting thing about a systems project is what's new. Is it entirely new functionality that will take the organization into the next decade? Is it replacing antiquated programs that barely work anymore? Is it simply tying together two systems that are fine alone, but will be amazing in concert? These are the things that drive that first presentation, that get everyone on board. They're the reason this project exists.

On the flip side, the least exciting thing about a project is often testing for quality assurance. No one wants to think about what can go wrong. But things can go wrong. Often, things do go wrong. A robust and detailed test plan can mitigate that risk enormously.

Quality assurance cannot be an add-on, an afterthought. The time and resources required to thoroughly test must be factored in as part of the project plan from day one. But what does that look like?

Integrating people, process and IT

www.rpesolutions.com ▪ 813-490-7000 Ext. 7353
info@rpesolutions.com

Four Phases to Success

True quality assurance is more than just turning on the system and looking around to see if anything is broken. The test plan should have detailed scripts, checking each component of the system. Each test scenario will evaluate a specific function of the system, with enough information to alert the tester to inappropriate results. But having a library of scripts is just the beginning. To minimize confusion and rework, break the testing into phases.

Unit Testing: This is the most basic, straightforward part of testing. Does each click of a mouse do what you expect it to do? Does 1 and 1 add up to 2? Sometimes called "four walls" testing, it's to check that what you've added to the system, in a vacuum, isn't broken.

Regression Testing: Similar to Unit Testing, this phase is concerned with the other parts of the system that weren't directly involved in the program change. Just because Function Z was added doesn't mean Functions A, B, and D weren't affected. This is particularly important when installing an upgrade to an existing system. It's important to check the new stuff. It's vital to check the rest as well, in case there's an impact that needs to be addressed.

User Acceptance Testing: It's easy to assume that Unit and Regression Testing will find everything that might need to be dealt with. But getting the users involved is important. They're the ones that use the system, day in and day out. They know every corner of it, and more crucially, they know how they use it. If the script they are handed describes one path to a particular outcome, it's the user's responsibility to question that, to say, "That's not how we do it." Making sure it will work as expected by the end users is the goal here. If they can break it, then it's broken.

End to End Testing: This is where everything comes together. Unit Testing will find actual "bugs", Regression Testing will find unexpected effects, and User Acceptance Testing will find front end inconsistencies. End to End Testing makes sure that this system interacts with the rest of the IT architecture as it should. Are the interfaces sent and received with no issues? Does the data maintain accuracy from point to point? Are the windows of operation within acceptable bounds? Coordination of a multi-system stream from data creation (e.g. creation of a new SKU) to final result (e.g. sale at a store) is complex, involving multiple teams who must stay in contact through the process. Finding an inter-system problem in the testing phase is far preferable to finding it the day of Go Live. Or worse, three weeks later.

Benefits of a Comprehensive Quality Assurance Cycle

Spending the time and effort to complete these phases and test the system so completely has the net effect of saving your organization time and effort in the long run. Here are some examples:

- Finding a "bug" early on in Unit Testing is easier to deal with than finding it when a user logs in on Monday morning.

- Cleaning up an unexpected interaction found in Regression Testing will save the End to End team from wasting time trying to work around it.

- Alerting users early on to how their screens and processes will change lessens the risk of Day One anxiety.

- Finding a potentially catastrophic inter-system issue in End to End Testing makes it less likely that a project will be labelled a failure.

Planning and executing a comprehensive quality assurance cycle will help your project be successful.