The Importance of Test Management in a Heterogeneous Environment
Executive Summary

In today's heterogeneous environments, organizations developing software must channel their requirements, from various sources right across the business, and collate the information from all of those teams, regardless of whether they are in-house or third party. The old adage, “Everything in its place and a place for everything,” rings true, and it is essential that everything is in one place.

Forward-thinking businesses understand that within their software supply chain they need visibility, not just for software developers and testers, but also for the broader business. Gaining visibility requires a consolidated, central view into a quality portal. Not having this visibility from all groups within the organization is akin to driving a car with only one window. To avoid accidents, organizations require openness and transparency to gain an accurate view of what is happening across the entire supply chain.

However, people are individuals with their own ways of working. And many organizations have naturally evolved heterogeneously with different teams in different locations, using their own preferred tooling to deliver their component parts. On top of this, there may be people running manual tests alongside people running automated tests.

Adding to the chaos is the fact that as projects grow in size and complexity, there is greater likelihood that there will be more than one organization providing all the services in the software development lifecycle.

“Even when all roles on a development team work in the same building, testing gets disconnected from other development activities. The ability to deliver valuable features in a timely manner, responding to changing business needs, depends on a short feedback loop.”—Lisa Crispin, Agile Testing Coach, Practitioner and Blogger.

Ultimately, the business need is imperative and gaining a centralized view of all the disparate information produced by different people in different teams spread over different locations is a major software development lifecycle challenge.

1 Software Testing in Distributed Teams—an article by Lisa Crispin, 2013.
   www.methodsandtools.com/archive/distributedsoftwaretesting.php
The Origins of the Heterogeneous Environment

With an increasing number of organizations seeking to outsource elements of their development and testing, distributed software and development teams are here to stay.

Vanson Bourne Research

Over time, organizations are outsourcing greater percentages of their application testing and development. In a survey of 590 senior IT decision makers interviewed in 2013, it was revealed that outsourcing of testing is set to increase from 43% in 2011 to 56% in 2015.²

The world is becoming more and more interconnected. As a result of this interconnection, there are an exponentially increasing number of organizations across the globe that, as well as developing and testing software, provide the entire spectrum of functions and services in the traditional software delivery lifecycle. These service providers are becoming more competitive, often as a result of location, which is why so many of their customers are outsourcing projects to them, especially when these customers do not have the technical capabilities available in-house. (They do, however, have dwindling IT budgets and resources.)

Software testing is considered to be one of the most important tasks for 91% of IT departments, and almost all believe it is crucial to outsource this activity, according to a report from analyst company Pierre Audoin Consultants (PAC). This report also reveals that three-quarters of companies already use service providers with offshore capabilities to provide testing services.³

Research and Markets

According to researchandmarkets.com analysts, “One of the main growth drivers for the global outsourced software testing services market is the need to reduce operational time and cost. This market is witnessing increasing demand for software testing from companies because the quality of applications is highly dependent on testing. Hence, to avoid poor execution, slow turnaround or excessive costs in the software’s lifecycle, companies are relying more on testing services—increasing the adoption of outsourced software testing services.”

Of course, there will always be organizations that are not outsourcing, simply because they have their own in-house capabilities. However, within these organizations the various teams contributing to the software development lifecycle are typically disparate, often based in separate locations, often internationally, so it is not uncommon for organizations that do not outsource to have dislocated ‘in-house’ development and testing teams.

Disparate Teams Using Different Tools

As a natural progression over time, these disparate teams, regardless of whether they are in one location, distributed across the organization, or working within outsourced teams, may have adopted or created their own preferred tools or had tooling imposed. Today, there are a plethora of tools available, ranging from in-house custom tools that are built for purpose to third-party tools and open-source solutions.

The reason teams use these different tools tends to be human nature; people stick with what they know. Employees usually opt for what feels comfortable to them and generally have a negative view of new technology, especially when it causes major changes to their job.

This is partly why the vast majority of organizations are still using Microsoft Office (Word and Excel) tools, because this software is essentially ‘what people are used to’ and easily available. However, this manual approach is steeped in inefficiency, and organizations must adopt an automated tooling solution to remain competitive.

As an application passes through the multiple stages of the software development lifecycle, the information being generated by all these various tools must somehow be brought together to enable the various teams to gain an accurate view of the project at any point in time.

Execution and Value Creation Phases

Within the software development lifecycle, one of the main focuses for outsourcing is the execution phase. This is partly due to the fact that this is usually the point at which complexity and costs are at their highest. If the execution phase is not a company’s core business, then yes, they

---

² Vanson Bourne: The problems of outsourcing application development and testing, 2013. www.microfocus.com/_ex/borland/assets/reports/Vanson-Bourne-testing-research-results.pdf
are likely to outsource it to developers or testers or any other service providers that are competing worldwide for their business, and who can complete the work in any location owing to modern Internet technology.

On the other hand, value creation is a phase that organizations most definitely want to own and manage in-house. They need to keep control of this area because this is where all their ideas and concepts are developed and where the project requirements are outlined. These requirements come in many forms, and from many sources, and organizations must ensure they are effectively gathered so that testing and validation can be correctly performed in order to meet the business requirements. Otherwise, the business will end up having disparate pockets of information with results that are unconsolidated and thus ineffective. All this information eventually needs to be transferred into the execution phase.

According to the Vanson Bourne survey, a staggering 81% of CIOs are not confident that they are communicating clearly with the companies that they have outsourced the execution phase to, which is obviously a major issue.

These CIOs’ organizations are lacking the ability to collate precisely what is being determined by the business, and then cannot connect that information with all the testing information to ascertain a precise level of testing coverage. This in turn would enable the organizations to start to prioritize their tests and make sure all the critical elements are fully tested. (Of course, in reality not every application feature can be tested but the critical ones must be.) So it is vital that organizations can meet quality goals through risk and requirements-based testing.

**Prioritizing Quality Criteria**

For successful application development, businesses require continual quality across their software delivery lifecycle, satisfying the demands of the development and quality testing teams. Full metrics and analytics are the digital paper trail that proves teams met quality goals.

Ideally, these same organizations need to prioritize and control all aspects of their testing by automating collaboration across the software development lifecycle, and using quality goals and risk-based testing can deliver higher quality software faster.

They need to gauge how well they are doing against their priorities to ensure all critical requirements are covered off. This is why it is essential for two-way communication across all the different groups to cover ‘what they want’ (the requirements) and also ‘what is being delivered’ from the testing and development groups. It is imperative that all this information is disseminated from all sides clearly, accurately and punctually.

**User Stories and Unit Tests**

When businesses start to develop software requirements, they build them from a purely business perspective. These requirements then need to be transformed during the software development lifecycle into what the development team needs in the form of a user story, a succinct explanation of what is to be developed. Then the developer takes the user story and develops the project.

While this is being carried out, the developer can also create unit tests, which could be in Java (JUnit) or open source (NUnit) or another unit test tool. So ideally, while the developer is developing the code, testers collect unit tests and run them regularly. Testing ensures that anything the developer builds currently does not adversely impact something that developers created previously.

Unit tests validate modules of code as the developer builds the application and perform in traditional, Agile and hybrid approaches to development. However, continuous integration testing (CIT), which is performed within an Agile approach, involves nightly builds and validation of the code. So the developers check in the code perform a build that is automatically validated. When the developer returns in the morning, the status of the build is ready for inspection.

This is an interesting concept because businesses usually consider test management to be something that happens once the development team passes the project over. In reality, test management can actually happen across the whole cycle, from discovery through to delivery.

“A software tester can begin testing early—very early—before the software has even been built. One of the best times to start testing a product is in the product-discovery phase… Defects found earlier in the development cycle are the easiest and cheapest to fix; imagine finding a defect before the software is even built!”


**The Importance of Being Open**

To enable collaboration, control and visibility, open test management is essential for today’s software development lifecycle. ‘Openness’ is required to handle any tooling, whether it is from a business, development

4 Vanson Bourne: The problems of outsourcing application development and testing, 2013. www.microfocus.com/_ex/borland/assets/reports/Vanson-Bourne-testing-research-results.pdf

or testing perspective. And organizations require a solution that can digest, integrate and communicate with those tools to ensure teams can visualize information at all points in the software development cycle, and at any time.

This openness also needs to extend to the way people work, whether they are using a traditional or an Agile methodology or a hybrid. Ultimately, organizations need a solution with the flexibility, adaptability and capability to support all methodologies from a process point of view.

**An Open Test Management Solution**

To remain competitive, today’s businesses need an open test management solution, one that truly reflects the way that organizations work. Micro Focus can help your business to develop a scalable, flexible test management engine to enable development teams to deliver higher quality software into the hands of your business users faster, ensuring greater value in a shorter timeframe.

**Summary**

This white paper has covered a range of challenges associated with heterogeneous environments and explained why visibility for test management, plus all the other elements within the software development lifecycle, is essential. The paper has also highlighted the importance of openness and transparency in terms of gaining an accurate view across the entire supply chain. Emphasis has been placed on the fact that organizations that are currently managing input from various sources across and outside the company must consolidate this disparate information into a central view accessible by all teams at all times.

Micro Focus’s Silk Central™ provides an open test management approach, enabling strong collaboration and control throughout all stages of quality with centralized visibility. Micro Focus’s solution uses built-in capabilities to capture requirements, create manual tests, manage automation through to execution, track defects and generate reports. Silk Central also has the ability to integrate leading requirements and to source control and virtualized environment tools. Ultimately, Silk Central brings together all testing needs and makes them accessible from a single point of truth.

For more information, visit: [www.borland.com/silkcentral](http://www.borland.com/silkcentral)

![Figure 2. Centralized test management is key](image)