Changing Trends in Software Testing

By Ramesh Ramani
The demand for software applications is ever increasing. Software development companies are working hard to keep up with this demand by delivering reliable, bug-free, and high-quality software. There has been a renewed focus on people, processes, and software testing tools. In the last 10 years there has been a paradigm shift within the software testing community.

I’ve picked out some top-of-my-list trends that I see emerging in the near future.

### Open-source Replacing Commercial Tools

The market share owned by open-source test solutions has seen a significant increase over the last few years. Today there is a higher demand for customized solutions. Unfortunately, commercial solutions are often unable to meet expectations. Additionally the cost of commercial solutions has also significantly increased. These companies have tightened their licensing model, which means that the user has lesser flexibility of sharing the licenses between projects and locations.

O’Reilly analysis shows an exponential growth of sourceforge.net in the last 10 years (ref. fig. 1). Innovation, cost, support base, and freedom to modify or build over the existing platform; have all become important aspects of testing and open-source directly suffices these needs. Successful usage of open-source tools like Bugzilla, MantisBT, Testlink, Selenium, Watir, JMeter for enterprise applications has given the confidence and comfort to companies to use open-source tools for testing.

### Test Early and Test Often

Traditionally, the testing is executed after the development team has finished code development.

However, at this point in the product lifecycle, testing doesn’t always get the attention and resources it needs. And the longer you wait to find a problem, the more expensive it becomes, both in terms of money and time. Therefore, more and more companies want to test concurrently with the development phase.

The graph in fig. 2 gives you a good idea about how early stage testing matches up against traditional testing. During the last phase of system testing, errors can be expensive since re-work effort is high. Additionally, minor requirements or design changes can become large development projects.

One of the common challenges in early testing is test au-
Non-functional Focus

Non-functional testing is a broad area that covers the quality attributes of a system. These aspects are not covered by functional testing and are often more difficult to quantify for both the business and development team. They are therefore frequently overlooked, often with catastrophic results.

If you carefully go through the diagram in fig. 3, the non-functional requirements primarily focus on change management, flexibility to port to different technologies, and customer focus. These quality attributes are generally not considered in the initial phase and are necessary to ensure that the product is sustainable and useful to an end user.

Many of the projects do not focus on the non-functional requirements in the initial phase of testing. Most of the time the non-functional tests are carried out after the functional tests. Many of the non-functional requirements are not addressed in the projects as these issues are found late in the cycle and fixing these issues may need design changes that need a lot of rework effort. With the user demanding fast, user friendly, and secured systems; performance, usability, and security testing are considered as focus areas. And many organizations have started building dedicated teams to verify the quality attributes from the design stage of the project.

Change in Domain Focus Strategy

A few years ago, we had a debate on whether the test engineer should test the requirements of the application or functionality of the application. Even though both of them may sound similar, the focus varies.

Domain is the foundation on which functional testing can be done. The credibility of a test engineer lies in the fact that he can test on behalf of business/operational users.

It is impossible for a project manager or test engineer or a business analyst to gain expertise to replace the business/operations users in testing the functionality. Hence focusing on a specific domain adds value and confidence to the customers. There is a definite need and focused test engineers tend to think and work like an end customer.
Tool Driven Process Improvement

Most of the organizations have evolved a unique flavor of SDLC and have branded the same with a unique name. Unfortunately the process still exists only on paper and not with the work.

Unless the process is monitored and tracked throughout the project, processes cannot be measured and improvement cannot be objectively monitored. There is a huge gap between the processes that are defined in the document and what is followed in the project. All the processes what we define need to be automated and managed using the tools.

A few years back, we were looking for process management tools and hardly had any choice. Today, there are ample of open-source and commercial tools that integrate processes with daily work schedules. For example, HP quality center in the commercial world or Testlink in the open-source world, helps us automate the complete process of test management and execution. Most of the process and entry and exit gates can be tracked and managed with the tool. Few years earlier the go-no go decision would be based on the team’s instinct and not a reliable statistical model. Today we are able to take informed decisions based on statistics that are churned out of these tools.

Overall trends seem to point towards significant changes that are in the offing for the testing world. Rather looking at functional specifications, test engineers will be forced to look at the application holistically and then design the tests. There is already a renewed focus on productivity of the people and tools will play an important role in process and product automation.

About the Author:
Ramesh Ramani is a technocrat and has over 14 years of experience in the industry. At the time this paper was written, Ramesh was the Director for IVV Practice at Tavant. He has specialized in product and process automation using open-source and commercial tools. He has customized and built frameworks that are used in testing enterprise-class applications.

As one of the early adopters of open-source in software testing, Ramesh has developed a combination testing tool called Jwrap published with the same name at sourceforge.net. He has participated and published papers in various conferences and international journals including StickyMinds.com and TicketIT.com

About Tavant Technologies

Tavant Technologies is a specialized IT solutions and services provider that employs cutting-edge and emerging technologies to provide game-changing results for its customers. Tavant prides itself on its traditions of engineering and process excellence coupled with high employee and customer satisfaction levels. Its unique best-shore delivery model provides close onsite interaction with customers and a strong process oriented offshore team. At all levels, its employees continually interact to provide a superior outsourcing experience to customers. Tavant specializes in building solutions and providing end-to-end services in the domains of Service Operations, Consumer Lending, eBusiness and Trading & Securities.