



Future of Software Testing

Last Updated: 7th July, 2008



Introduction

Trends in the industry suggest that software testing in the future will look very different than it does today. Businesses are leading change and transformation projects and application development is changing to support it; current trends include:

- ▶ Web based applications
- ▶ Service Oriented Architectures (SOA)
- ▶ Software as a Service (SaaS)
- ▶ Wireless technologies
- ▶ Mobile technologies

Each of these trends is further complicated by a more agile approach to software development and an increasing emphasis on the 4Rs; repeatability, reliability, re-use and robustness.

This paper investigates how testing will need to change to accommodate these trends and become a business led activity.

Predictions about the Future of Software Testing

This paper attempts to predict how testing can become more effective and efficient so that it can keep pace with, and support the changing trends in software development.

In order to drive these changes in effective and efficient application development of quality systems, more emphasis must be placed on improving the Analysis and Design phases and ensuring early engagement of testing in these areas.

To date, analysis and design have been people based activities with Business Analysts and Systems Architects producing a combination of natural language, models and diagramming techniques (such as Data Flow Diagrams, Logical Data Structures, Logical Data Models and Process Data Diagrams) and some structured language techniques (such as Use Cases and UML). This has led to business requirements and functional designs which do not meet the four key criteria which make the resultant systems “testable”; clear, assumption free, unambiguous and complete.

In the future, if we are to enable the production of quality applications, in less time and at lower cost, we will see more structured and more automated analysis and design

techniques, with Static Testing (reviews, inspections and walkthroughs) becoming “robotized” and built in as part of the analysis and design requirements capture tools. This will result in clarity, the eradication of assumptions and ambiguity and a greater likelihood of completeness which will in turn enable:

- ▶ Executable specifications
- ▶ Auto generated code and systems configurations
- ▶ Model-based test generation
- ▶ Model-based bug prevention
- ▶ System simulation
- ▶ Fault tolerance developing into self-testing, self monitoring, self healing software
- ▶ Performance optimization “designed in” and “on the fly”

So what will it mean for Testing and QA? Well the products should be more reliable, more reusable and more robust. Automatically generated test scripts with full branch and decision tree coverage at the early test execution stages; unit test and “integration in the small”, combined with the development of fault tolerance into self-testing, self monitoring, self healing software will result in defects being found and eradicated earlier in the lifecycle. As a result, fewer defects should be found at the System Testing and “Integration in the large” stages and businesses will be able to concentrate on Acceptance Testing (User, Operational and Performance) in a fully integrated target environment and do full end to end testing of business processes and model office scenarios. In this way, businesses will be able to leverage their investment in technology to realize true business benefits and competitive advantage through creative integration and deployment of reliable, re-usable, robust components; the way in which the components are integrated and deployed will be the differentiator, not the components/systems themselves. In turn, this will enable business to concentrate on business propositions, and go to market services and products, supported by the ability to deal with high volumes, provide excellent customer service and quickly enable high margin/low volume services etc.

Testing Robots? What about Professional Testers?

Software testing in the era of mainframe systems was merely executing a series of steps manually and punching various key strokes as an attempt to break the application. This

was followed by the event-driven client-server environment in which applications were deployed on multiple hardware platforms in conjunction with various other applications. As a result, the scope of testing has increased as failures and unexpected behaviors have become more prominent.

The recent development of web applications, where multi-layered designs were architected in order to hold complex business logic, which are to be supported on numerous browser types and various platforms and different network structures, further increases the potential areas of failure. In addition, web applications say something about the integrity of business operations to a large audience of external users (i.e. customers, business partners) and can have a major impact on brand and reputation.

New and emerging technologies, such as Service Oriented Architectures (SOA) are significantly changing the trends in the application testing. SOA in particular has major implications for how tests are designed and performed. Software as a Service (SaaS) also has a totally different view of software testing.

We can conclude that new and more complex technology/architecture creates greater risk.

So, how will the new approaches postulated in the predictions in the previous section of this paper manifest themselves and what will it feel like to be a Test Professional in an increasingly automated and “robot” driven software engineering environment?

Testing mitigates risk and despite automation and “robot” driven testing techniques, the Testing Professional will continue to be pivotal in the Software Engineering Lifecycle. Working in multi-disciplinary teams alongside Subject Matter Experts, Business Analysts, Systems Architects and End Users they will continue to play the roles of; arbitrator, mediator, translator, negotiator between the other parties, ensuring that their ideas are not only effectively specified and designed, but that the criteria required for effective quality and testing are captured and articulated into the

models and tools so that tests are accurately generated by the “robots” to verify the design and validate the requirements.

In this new world, the Professional Tester will concentrate on adding value to the business and the software development lifecycle and will operate in a new and elevated position where the combination of their structured process driven approach, creativity, destructive inquisitiveness and ability to articulate and define quality and testing criteria into the models will be critical to the success of the business implementation of technology.

Summary and Conclusion

Change is a given and we can be sure that it will come in larger amounts and at increasing pace. No industry, least of all software engineering, is immune from change and therefore none is immune from risk. The changes in software development will require software testing to develop in parallel. Of one thing we can be sure, the role of the Professional Tester will become more interesting and more essential in both the definition of model based testing criteria and the facilitation of Acceptance Testing (User, Operational and Performance), testing of business processes and model office scenarios alongside Subject Matter Experts and End Users and ultimately realizing business benefits.

Testing and Quality Assurance will become more important and add more value as we move into architectures and technologies that support the business in their goals of bringing products and services to the market as rapidly as possible, with minimal risk. We will see a shift towards testing the transformed business operation, the business processes, the way people interact with the systems and processes and the information it provides, and therefore mitigating the risks and increasing the benefits of business change.