



Hi. This is Trevor Atkins, Principal Consultant with Silverpath Technologies. In this presentation, I will discuss the importance of involving testing throughout the software development lifecycle.

Organizations developing software have been challenged for decades with attaining effective quality assurance and testing practices. Still testing remains regularly faced with the lack of up-front planning, the lack of trained resources, and most of all the lack of inclusion in the project team as a key stakeholder able to add value throughout the various phases of the project.


For instance, when was the last time your testing group was involved from the beginning of a project, had an influential voice in the development of the project plan, was able to hire based on a need for test planning, or was considered to be a strategic competitive differentiator by the company.

Too often a test team is employed only towards the end of a project cycle. This means the test team is not as fully leveraged as they could be and therefore the project is not reaching as high a quality bar as it should.

Silverpath works with our customers' in-house project teams to integrate our adaptable approach of Thinking Through Testing™ which emphasizes the inclusion of testing activities early in the project lifecycle. Thinking Through Testing enables our customers to achieve superior results over "traditional" testing by increasing the overall effectiveness and efficiency of their quality efforts.

In the next few slides we will review some of the background to the value of this approach and the base principles of the approach itself.

Attributes of Quality Software



- ❖ Satisfaction with the overall quality of the product can be evaluated through customer surveys.
- ❖ For example, the specific parameters of customer satisfaction in software monitored by:
 - ❖ IBM include CUPRIMDSO (capability / functionality, usability, performance, reliability, installability, maintainability, documentation / information, service, and overall);
 - ❖ Hewlett-Packard include FURPS (functionality, usability, reliability, performance, and service)

– Stephen H. Kan, “Metrics and Models in Software Quality Engineering”

©2009 Silverpath Technologies Inc.
2

To start off, let’s consider what is quality? Quality can be elegance, correctness, or fitness-for-use which was used by Joseph Juran as a definition for quality in 1974 and is the one that we subscribe to most.

Why is quality important in the software business? It is true that the degree of quality doesn’t necessarily impact market share directly for the majority of organizations. But, the lack of quality, competitively, will certainly have an adverse impact on reputation and revenues.

However, as beauty is in the eye of the beholder, to tell if a thing is of quality can vary greatly based on market competition or the intrinsic value perceived by the customer. Without further definition, quality is difficult to measure in of itself and to determine what is “good enough” quality, even more so.

Decomposing the attributes of customer satisfaction, with respect to fitness-for-use, into specific parameters can help provide some level of comparative objectivity when assessing the degree of overall quality between products and their benchmarks.


For example, Stephen Kan notes in “Metrics and Models in Software Quality Engineering” that IBM monitors the parameters included in the acronym CUPRIMDSO and Hewlett-Packard assesses FURPS.

But we can’t wait to ask the customer if they like the product after it is built (or even in User Acceptance Testing). We need to determine if the system meets the needs of the users in their environment, if it is fit-for-purpose, if it is of “good enough” quality, as far in advance of the release as possible – when change will cost as little as possible.

This is where testing can really help out. Through effective verification and validation activities, testing can help answer the questions posed by Barry Boehm in “Software Engineering Economics” – “Are we building the product right?” and “Are we building the right product?” at each phase of the project.

Fundamentally, testing serves as an investment in mitigating the risk of poor quality both in the product and for the business.

NIST 2002-10 Report



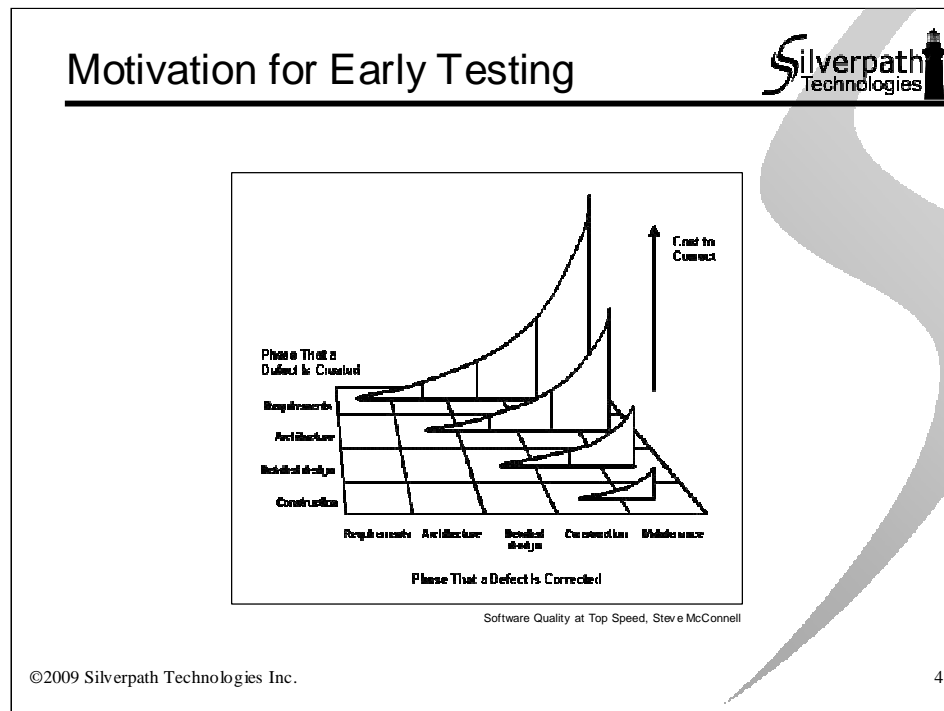
- ❖ According to a study commissioned by the National Institute of Standards and Technology:
 - ❖ Software errors cost the U.S. economy an estimated \$59.5 billion annually, or about 0.6 percent of the gross domestic product
 - ❖ 80 percent of the software development costs of a typical project are spent on identifying and fixing defects
 - ❖ About one-third of these costs, or an estimated \$22.2 billion annually, could be eliminated by an improved testing infrastructure

©2009 Silverpath Technologies Inc. 3

As an industry backdrop in regards to addressing quality and testing challenges, we can look at a study commissioned by the National Institute of Standards and Technology in 2002 that reported that software errors cost the U.S. economy almost \$60 billion annually, that 80 percent of the software development costs on a typical project are spent on identifying defects and performing the associated rework, and, that one-third of these costs, about \$22 billion annually, could be eliminated by putting in place an improved testing infrastructure.

There has been marked improvement in regards to the recognition and acceptance of testing as an integral part of any SDLC and testing is becoming big business with IDC reporting that worldwide and U.S. offshore IT services for 2006-2010 related to software testing are worth \$13 billion. Yet testing is still often an afterthought in many organizations, utilized only to look for defects in a system that is already integrated.

In this context, consider what investment you are currently making in testing and related quality activities. Now consider that they could be made significantly more effective and efficient relatively easily.



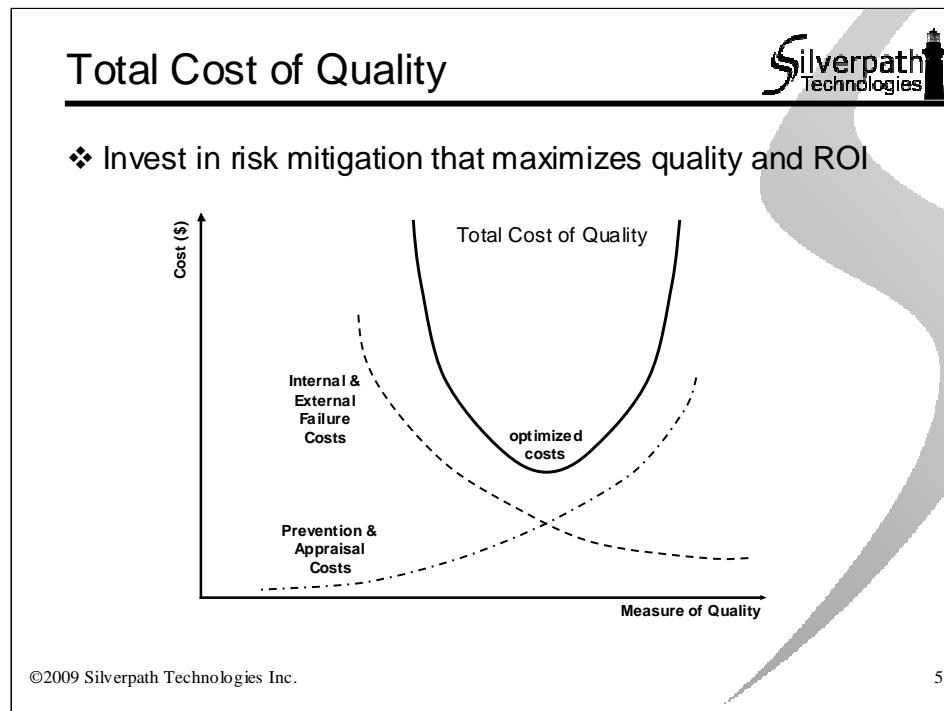
We are all familiar with the longstanding premise that the earlier a defect is uncovered and fixed the less it costs the organization. And, the more complex a system is, the more pronounced this tends to be.

Jeffery Payne comments in “Quality Meets the CEO” that “Corporate management does not care about quality. Management cares about profits, revenues, earnings, and market share. Software is the profit center that makes money. And quality is seen as a cost center that eats money.”

The perspective of this type of corporate management is unfortunately short-sighted, as in fact the flipside is the reality in regards to quality. Just as poor quality in the released product hurts market share, the lack of quality within each phase of the development lifecycle costs the company hard, quantifiable, dollars.

Steve McConnell provides this diagram in “Software Quality At Top Speed” to illustrate the concept. For example a defect introduced and fixed in the requirements phase costs one dollar, left in until design that same defect now costs ten dollars to fix, \$100 in coding, \$1000 in testing, and then much much more once the product is released to the field – a rapidly rising expense curve.

So, how can we get testing to prevent or detect the introduction of issues in each phase as early as possible?



First, let's look at another element of the business case for early involvement of testing. Testing is part of a set of risk mitigation activities where the basic principle is to spend money wisely now in order to avoid greater costs later.

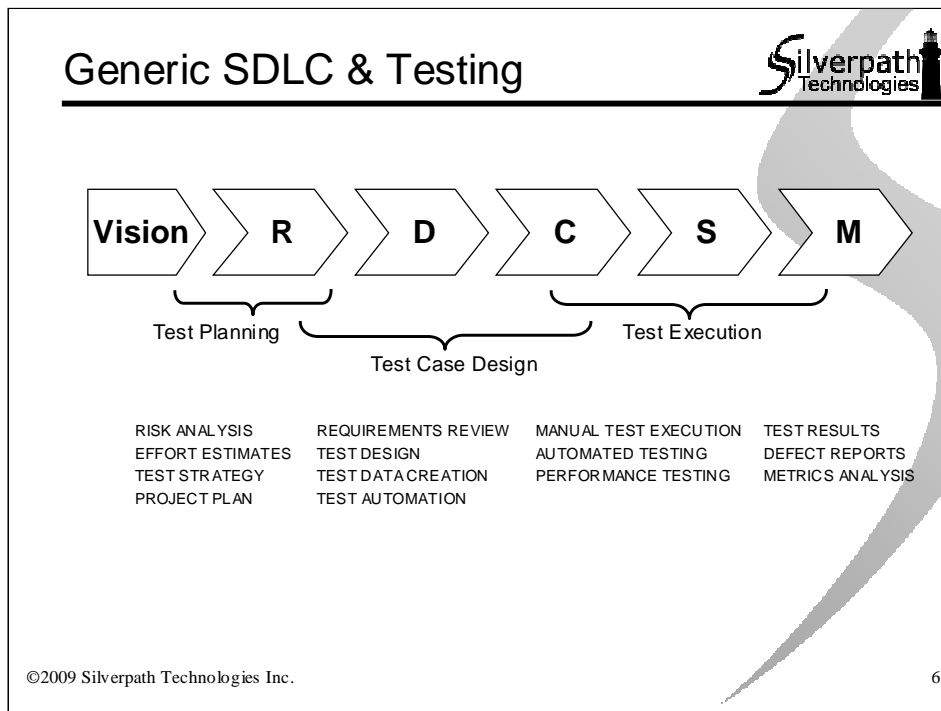
However, goals such as “completely test the system” and “find all the bugs” or even “find all the important bugs” imply an enormity of effort that just isn't reasonable for the typical software project. It is a much more reasonable objective to perform a level of testing that is sufficient to attain an acceptable level of risk in terms of the balance between the quality requirements, the scope, and the constraints for a given product release. Each organization and even each project must find the optimal mix of investment in upfront quality-related activities versus potential costs of inefficiency or failures down-the-line to minimize the Total Cost of Quality.

To help find this balance, let us classify the impacts, costs, and improvement benefits into the four categories of the Total Cost of Quality as adapted from Cem Kaner's outline in “Quality Cost Analysis: Benefits and Risks”:

- **Prevention costs** are related to those activities that are specifically designed to prevent poor quality including coding errors, design errors, mistakes in the user manuals, as well as badly documented or unmaintainable code.
- **Appraisal costs** are related to activities designed to find quality problems, such as code inspections and any type of review or testing.
- **Internal Failure costs** arise before your company supplies its product to the customer including such things as rework, overtime, and wasted/lost time.
- **External Failure costs** are realized after your company supplies the product to the customer, including such things as customer service costs or the cost of patching a released product and distributing the patch. These costs can be extended to include impact on the perceived value of the product delivered and the reputation of your company.

Naturally, prevention and Appraisal costs increase as smart investment increases, but as a result Internal and External Failure costs decrease. In accordance with the law of diminishing returns, where the two lines cross on the example graph above suggests the optimal balance of the four categories.

At Silverpath, we work with our customers to increase the effectiveness and efficiency of their software systems, processes, and teams. We help them incorporate reviews, efficient test planning techniques, risk-based testing, and strategic automation into their projects as a direct investment in quality that maximizes ROI, reduces failures, and achieves a net reduction to the Total Cost of Quality.




In her article, "No More Second Class Testers!", Johanna Rothman suggests that if you don't associate testers with each development function, or you think that testers exist only to find defects, then you are not receiving the full potential value of your testers.

Each project cycle will pass through a set of common phases one or more times, regardless of development methodology. This provides the opportunity to integrate typical verification and validation activities with each stage such that early detection of issues is realized.

Starting test activities early means you can catch small quality problems before they become big quality problems later on.

Further, early involvement allows evaluation of important planning, design and development decisions with respect to how these decisions aid or impair the testability of the system.

Example: Requirements Review



❖ *"The communication of functional requirements and specifications is the most difficult, critical, and error-prone task in IT projects."*

Bill Walton, A Systematic Approach for More Effective Communication of Functional Requirements and Specifications

❖ *"The criticality of correct, complete, testable requirements is a fundamental tenet of software engineering. The success of a project, both functionally and financially, is directly affected by the quality of the requirements."*

Theodore F. Hammer, Linda H. Rosenberg, et al., Doing Requirements Right the First Time!

©2009 Silverpath Technologies Inc.
7

A review of the requirements is a great example. The best time to catch defects in the product is to do so before the first modules are even coded.

Requirements information is a primary input to the software project, setting the purpose and scope for the majority of stakeholders' activities.

However, many projects lack formal requirements. Because of the huge pressures to show tangible progress, documentation tends to fall by the wayside. And, when there is a set of documented requirements, they might be incomplete, lacking enough detail, inaccurate, unapproved, conflicting, out of date, or ambiguous, etc.

Bill Walton claims that the communication of requirements is the most difficult, critical, and error-prone task in IT projects.


Hammer and Rosenberg go on to say that the success of a project, both functionally and financially, is directly affected by the quality of the requirements.

In the article "Inspecting Requirements", Karl Wieggers adds that approximately 50 percent of product defects originate in the requirements. And perhaps 80 percent of the rework effort on a development project can be traced to requirements defects. He further advises that anything you can do to prevent requirements errors from propagating downstream will save you time and money.

Frequently, the impact of these problems surface late in the project at the time of acceptance as discrepancies between what was built and what the customer thought was being built – a clear case of poor quality causing customer dissatisfaction and creating costly rework.


Involving testing early in the project lifecycle provides the chance to review requirements for important quality attributes, to ask questions, and resolve issues before they become much more expensive problems in the code.

Thinking Through Testing™



❖ **Our Approach**

- ❖ Risk-driven, ROI-centric
- ❖ Provide feedback on quality as early as possible
- ❖ Execute the right testing at the right time
- ❖ Leverage standardized light-weight practices
- ❖ Conduct in-parallel test activities
- ❖ Maximize reusability
- ❖ Continuous improvement



©2009 Silverpath Technologies Inc.

8

Testing is a strategic component of any software development process, but typically faces serious challenges both within the project and at the organizational level.


When building the “right fit” test approach with our customers, Silverpath enables their project teams to maximize value to the business and the current release while working within cost and schedule constraints.

Our approach includes:

- Risk-driven testing that considers risk to the business as well as project and technical risk.
- Test activities that start early to give feedback early (eg: project planning and requirements review).
- Executing the right testing at the right time; prioritized and scheduled in consideration of dependencies.
- Standardized light-weight test & defect management practices.
- In-parallel test activities to minimize the critical path in the schedule and find issues sooner.
- Maximizing reusable test artifacts to benefit the next test cycle, the next release or the next project.
- Gathering metrics data for timely decision-making in-project and driving future improvements in the organization.

Thinking Through Testing enables our customers to clearly articulate and successfully execute the quality plan that achieves their individual needs within their unique situation.

Summary



- ❖ Silverpath is founded on the principle vision of providing high value to software companies by raising the quality of their software systems and teams while simultaneously driving down the total costs to achieve that quality.
- ❖ Silverpath engages in a highly collaborative relationship with each customer where focus is driven by the emphasis on improving the effectiveness and efficiency of quality and testing activities.
- ❖ Working together, we obtain the practical results needed from the optimal balance of quality requirements, scope of effort, and project constraints and thereby achieve the greatest return on investment.
- ❖ Visit <http://www.silverpath.com>

*Thinking
Through
Testing*

©2009 Silverpath Technologies Inc. 9

An effective and efficient testing group will provide timely visibility into the quality and readiness of the system and its input artifacts that is necessary to enable effective decision-making throughout the software development lifecycle.

At Silverpath, the core practices and techniques we bring to our customers' project teams are built upon our "fitness for use" philosophy of quality, where the customer satisfaction of both the business and end-users determines whether or not the project was a success.

You may even discover though implementing our recommended practices that, in terms of the bottom-line, quality can be free after all.

Thank you for your time and I encourage you to visit the Silverpath website at www.silverpath.com for further resources related to this presentation and details on our consulting and training services.