The Effect of Highly-Experienced Testers on Product Quality

Alan Page
Director, Test Excellence
Microsoft

Abstract

In order to keep up with improvements in software development approaches and tools, the growing complexity in software applications, and the fast ship cycles of a software plus services world, we will need to advance the state of the art in software testing. Advances in the state of the art will require experience-driven improvements that change the way testers approach testing, and how product teams approach quality. But how can we do this when the testers in most organizations are far junior in experience to their development counterparts?

A relatively small group of highly experienced senior testers at Microsoft are working to do their part of advancing the state of the art in testing. They have solved problems that have enabled entire organizations to improve testing efforts and product quality. They've done it by leveraging years of experience in software testing and the respect of their peers in order to solve extremely difficult testing problems that enabled the success of their teams.

This paper discusses three case studies showing how highly experienced testers solved big problems at Microsoft and helped their teams make improvements on quality that absolutely could not have been made without their efforts, as well as the effects a growing group of highly experienced testers can have on an organization. The paper also shows how an organization can model and groom newer testers for taking on such roles in the future, and discusses what Microsoft is doing along these same lines, and what the future plans are for leveraging senior test talent.

Presented at the Pacific Northwest Software Quality Conference 2009

The Effect of Highly-Experienced Testers on Product Quality

Software testing is a new profession compared to software development, and the roles of testers and the scope of their impact on product quality are ripe with confusion and controversy. For some, software testing is an activity performed by people whose role is to mimic the user (or in some cases by users themselves) in order to understand and report on issues that will affect the end-user usability of the software program. For example, a bank may employ a software development team to write account management software for bank tellers, and use a subset of those tellers to "test" the software before deploying throughout their entire infrastructure. For others, software testing may look beyond user scenarios and attempt to perform a more holistic evaluation of the software under test that also takes a deeper look at functional and structural aspects of the software system.

On many software teams, test roles are less valued than development roles. Indeed, if the traditional "cost of change" curve is true (see figure 1), then the emphasis should be on finding and fixing issues introduced during requirements, design, and coding – but on many test teams, the bulk of bugs are discovered much later in the software lifecycle. Despite the increased cost of fixing issues found during system test, the types of issues found during late cycle testing may not seem to warrant the same respect and pay scale as core development work does.

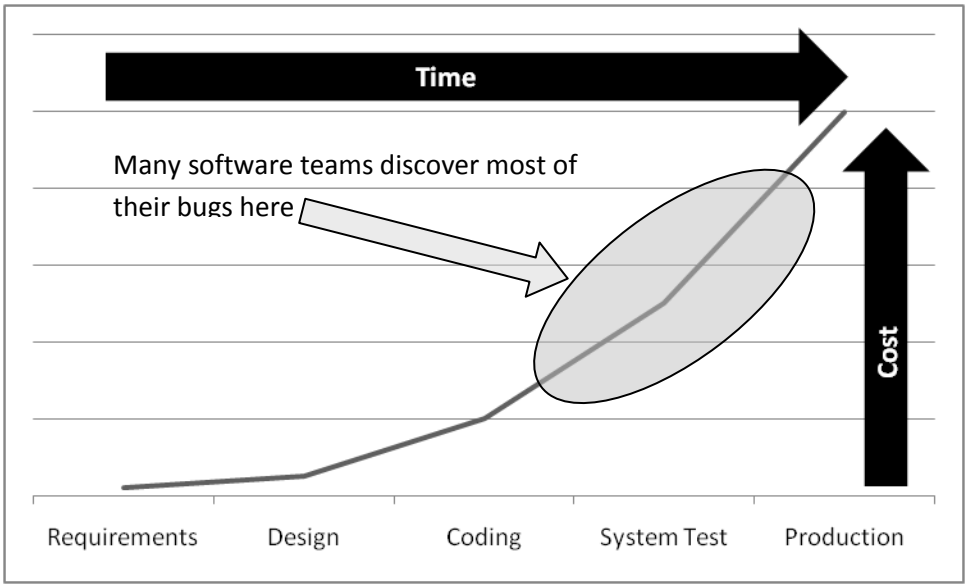

Figure 1 - Cost of change curve[1]

Moving quality up stream is much more than hiring testers earlier in the product cycle. As is well known and generally accepted in the industry if you want to find errors earlier, you need to *look* for them earlier. For most software applications, finding and detecting design issues – especially

[1] Boehm, Dr. Barry, Software Engineering Economics, Prentice Hall, 1981

during their introduction in the design phase, is a more difficult task than finding bugs during testing. Iterative methods may flatten the curve by cycling through code and test phases more quickly, but frequent iterations still often neglect to find fundamental design issues. Regardless of the specifics of the approach, a process that heavily leverages testing and quality throughout the product development cycle is a path towards better software.

The Microsoft Approach

For at least five years, the primary candidate pool for software testers at Microsoft has consisted of graduates from computer science related fields. While we do expect our testers to be able to write tools and automation when needed, a more complete reason why we hire from this pool is that we expect our testers to debug, diagnose, and analyze problems they run into. We also expect them to recognize patterns of bugs, and have insight into how the computer may be using the data they input. These skills decrease or eliminate the amount of time it takes to get bugs fixed (developers often know everything they need to know to fix the bug from the report), and that usually ends up increasing overall productivity. To be fair, we have had great testers without a computer science background who have been able to develop these skills successfully, but we've found much more success in hiring computer science majors and letting them develop into fantastic software testers.

At Microsoft, employees have the option of changing groups and disciplines at almost any time. Given the strong software development background of these testers, some choose to move on to software development engineer (SDE) roles at Microsoft, but most remain in test positions throughout their Microsoft careers.

Career Paths for Testers

An important thing to note is that testers at Microsoft have the same salary structure and promotion possibilities as their counterparts in other engineering disciplines. Over the last two years, a number of senior leaders in software testing at Microsoft have been studying the role of senior testers with the goal of helping to clarify the test career path. One output of this work was the creation of four senior test personas[2] (used internally and shared externally). Each persona describes a potential role a senior tester may play on his or her team. A common theme among the roles described in the personas is leadership. Senior testers provide technical leadership, mentoring and vision for their test teams, and they improve their test teams through this leadership. The current phase in the ongoing investigation into the roles of senior testers is to examine the roles of some of Microsoft's most senior testers and understand not only how they make their teams better, but also how they make their products better.

There has been some success in the work so far. Due partially to consistent hiring practices, and partially due to the work of test leadership, the number of testers in senior positions has grown by over 300% in the last two years. As the number of people in these senior roles has grown, it's important to be able to take a step back and ask, "To what end? Do we just have more people in senior roles, or are they collectively making an impact on product quality?" Measuring the direct impact of the increased population of senior testers is difficult, but we plan to continue investigating this impact over the coming months and years. For example, we are

[2] http://msdn.microsoft.com/en-us/testing/bb414765.aspx

looking at post-release quality measures (e.g. patches, customer feedback, and data from product support), and looking for correlation between perceived quality for certain products (or product areas) and the makeup of tester experience on those product teams.

The author separately interviewed several of Microsoft's most senior testers and their managers in order to obtain qualitative information on the influence of these testers on product quality. The focus of these interviews was on testers in senior non-management roles. Their peers in a development organization are typically Software Architects or Principal Software Development Engineers.

Case Studies / Interviews

Two primary factors helped drive the basis for inclusion in these interviews. First, the testers profiled are in the career stage known inside Microsoft as "Principal". Employees at the Principal career stage comprise approximately the top 4% of all Microsoft engineers, and the top .5% of testers in non-management roles. The second factor influencing the selection of case study candidates was to target testers who most typically represent the role towards which Microsoft expects highly experienced testers to grow. Those selected are strong performers, embody the model of the senior test role and function, and in many ways act as role models for all testers at Microsoft.

The interviews with managers and employees were free form, but included a guiding set of questions. These included questions about:

1) Role on the team
2) Level of influence
3) Influence on improving testing
4) Impact on product quality
5) Estimated impact on testing and quality if the individual were *not* on the team

Interviews occurred in person in the spring of 2009. Three of the roles are profiled in the following section of this paper. Descriptions come primarily from their direct managers. Feedback from peers and colleagues are also included.

The Role of the Highly Experienced Tester

The testers profiled in the following section are in non-management roles. While there is definite need for senior roles in test management, the focus of this paper is on the role of senior testers in non-management roles.

The role and tasks of senior testers vary depending on the skills they possess and the needs of their organization. In some cases, they may be responsible for developing or architecting the test infrastructure used by the team (testers at Microsoft build and maintain most of their own test tools and infrastructure). In other cases, the role can begin to look more like a traditional quality assurance role, where senior testers are responsible for establishing and measuring the quality processes of all disciplines on the team (note that Microsoft does not currently have a quality assurance profession).

In many other cases, senior testers are deep subject matter experts for the technologies they test, or for specific testing methods such as security or performance testing. In all cases, senior testers bear some responsibility for making their teams better – through example, technical leadership, or through direct mentoring or coaching.

Lucretia – Windows Operating System

Lucretia has been at Microsoft for 12 years, and is a Principal Software Design Engineer in Test (SDET) on the Windows team. She is responsible for determining how to test a vast number of Windows operating systems configurations. The testing matrix for Windows SKUs (stock keeping units) is highly complex. Windows Vista, for example, has six primary SKUs available, but those SKUs are available in 32 and 64-bit versions and in dozens of languages. Additional variations are often made available due to regulatory requirements (e.g. offering Windows Vista in Europe without Media Player). A massive number of possible combinations of components could theoretically make up a Windows release. While Windows ends up shipping only a subset of these possible combinations, the ability to select from all possible combinations is necessary in order to respond to issues in a global market. Rather than worry about the implications of how to test numerous combinations of components across Windows, the work Lucretia does takes out any guesswork about what we ship to customers. For example, if removal of a feature or component from Windows is required, Lucretia's work can point out exactly which binaries need to be removed – verifying both that the feature is removed, and that *nothing else is broken* due to dependencies on the removed components. The same rules of dependency analysis are also applicable to patches or hot-fixes.

Her level of influence across all disciplines in the organization is very high. There are teams who base their strategy and workflow on the work she produces. If she were to stop doing her work, the ability to ship a hot-fix would be greatly reduced, and Microsoft's ability to react to high priority customer issues would be greatly impacted. This ability to identify and reduce risk in software is a big part of the test role at Microsoft and this core ability is nurtured and fostered throughout the career of all testers. Software developers will often come to Lucretia and ask for her opinion on how to handle specific issues related to componentization and dependency analysis. Her impact reaches far enough beyond her team to the extent that other organizations have used their own budgets to give her additional rewards. In her manager's estimation, Lucretia has found or prevented thousands of bugs. She is highly trusted and helps build many positive relationships for her manager.

Without her on the team, the cost of testing would rise dramatically. Her expertise, along with the toolset she has developed enables her to accomplish alone in 4-5 hours what used to take a 3-person team an entire week. She enables the team to ship far more SKUs, patches, and service packs than they could without her experience and knowledge of the problem domain.

Ed – SQL Server

Ed is a Principal Test Architect ("Test Architect" is a role of some Principal SDETs at Microsoft) on the SQL Server Team. Ed is deeply involved in identifying and solving the problems of his product unit, but he also has the freedom to help and educate others outside of his product

group, and is extremely influential in advancing the engineering of testing across the entire company.

Recently, Ed helped a team move to a declarative testing model for some of their user interface automation scenarios – moving the team from focusing on implementation details to thinking about the scenarios that drive the test cases. This reduced complexity of verification improved efficiency on the test team and enabled the team to reach levels of test coverage equal to their previous efforts.

Ed is extremely influential in the design of tests across his entire organization. He makes the testers on the team better both through technical leadership, and by teaching other testers to think clearly for themselves and ask the right questions upfront. His approaches to test and test design have enabled the testers around him to write fewer tests yet reach higher levels of feature and code coverage.

Mentoring other testers is a big part of Ed's role. He teaches testers to approach the testing task in a structured and practical manner - making them *think* about the task they are facing and what the full picture of accomplishing that task includes before encouraging them to dive into the technical aspects of the problem. He also is astute at identifying missing subtleties in existing approaches, and constantly helps the testers on his team improve the way they think about testing.

Without Ed on the team, the momentum of advancement in his testing organization would slow measurably. It is also quite conceivable that his business unit would have done less product development, as Ed's approaches have enabled the test team to do more work with fewer people and reach higher quality. He helps testing focus on the areas that are most important.

Jim – Internet Explorer

Jim is a Principal SDET on the Internet Explorer team, and has been at Microsoft for 13 years. He primarily works on site compatibility features in Internet Explorer. He develops systems for identifying the extent and severity of compatibility issues. Although he has no direct reports, he essentially runs a large part of the testing organization through technical leadership and vision. He has created several case studies for the team (and other teams) from which to learn. For example, he would start with a bug report, and then show the team how to tear it apart and find the root issue. Currently, Jim is a big part of the planning for the next version of Internet Explorer.

He worked with many of the functional teams within the organization, and has helped them make adjustments in order to help them test better in areas such as performance and reliability. His peers often cite his experience and technical depth as a key part of his success. In at least two instances, testers were considering leaving the team to take on different roles, but in fact decided to stay due to Jim's leadership and his position as a role model.

Of all the individuals on the test team, he has had the most impact on product quality. Jim personally investigated over 1000 compatibility issues over the course of the release. Some of the issues came back as fundamental design issues in the product – issues that may not have

been found at all without his work. Overall, his investigations and analysis have been critical to the success of the project.

In one example, Jim investigated an issue where menus were not displaying correctly on a web page. After extensive investigation, he discovered that the error occurred due to a design change in the application of DHTML filters when rendering web pages. He then discovered a similar bug in the product bug database initially resolved as “won’t fix / by design”. With Jim’s added debugging information, as well as a real world example, the bug was re-opened, the design was corrected, and in the process, dozens of other rendering problems in prominent sites were fixed.

Jim is also involved in spreading his knowledge across the team to help them increase their own capabilities. Jim creates full write-ups documenting many of his debugging sessions and shares them with the rest of the team. This provides guidance and demonstrations of many important investigation concepts and debugging tools. As a result, the entire team is much better equipped to tackle difficult bug investigations and debugging challenges.

Without Jim on the team, improvements in his area would have been minimal over the previous release. The team would not have been able to scale to the breadth of issues (and depth of critical issues) in a methodic way. Many design decisions and the rendering engine would not be as good without him.

Affecting the Cost of Change

What happens when you have experienced testers on your team who can influence early code decisions and product design, and who can introduce preventive techniques into the development process? The data thus far indicates that the inclusion of highly experienced testers on a team can push discovery of many more bugs earlier in the software lifecycle, therefore saving money and increasing quality. The influence of the experienced tester throughout the product cycle would certainly have an impact on the shape of the curve, and it would be reasonable to expect that the curve would flatten.

Finding a bug late in the cycle is still expensive, but better design decisions early on can potentially lower the cost of post-release bugs. Additionally, more early detection and preventative techniques will greatly reduce the *number* of bugs found post-release decreasing the *overall* cost dramatically. If experienced testers are champions of early detection and prevention, and these practices become more prevalent across a product team the overall cost of change should surely go down. Figure 2 shows an alternate cost of change curve affected by the work of experienced testers.

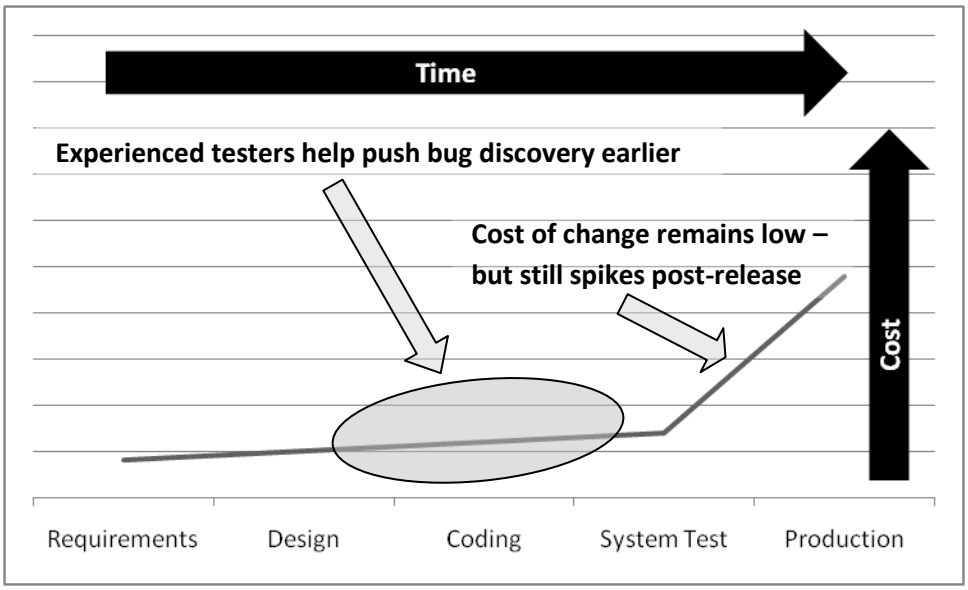

Figure 2 Alternate cost of change curve

How Do You Get There?

Testers with this level of influence and impact don't appear magically, nor are they created through special training. Testers develop into these roles through personal growth and development, as well as simply racking up years of experience shipping products along the way. Some of the common keys to success shared among nearly all of the Principal SDETs at Microsoft include:

- **Technical skills**
  - You can't be technical enough at Microsoft
- **Align your work with business goals**
  - What business problems did your efforts address today?
- **Know yourself**
  - Know the combination of skills that makes you unique
- **Be skilled and creative**
  - Continually learn and push yourself
- **Be dependable**
  - Build a reputation of reliability and confidence
- **Be productive**
  - Do your job – look for ways to be better
- **Be a brand**
  - Be an expert, be consistent, be reliable
- **Help others**
  - Look for opportunities to mentor, coach, and lead. Base your success on the success of those around you.

The above, of course, is not a recipe. Just because someone has experience, is dependable, and provides technical leadership and mentoring for his or her team doesn't mean (s)he will automatically provide the same level of impact as the individuals discussed above.

Another important point to note regarding these roles is that senior testers cannot be the saviors or heroes of their teams. They need to leverage their deep knowledge and experience to solve huge problems on their teams, but the only way for them to scale is to ensure that they spend some portion of their time helping their team get better. It is of little value to have a tester on the team whose primary role is to fight fires and take on challenges independently. Instead, they need to involve the team in order to solve problems collaboratively and give everyone on the team a chance to gain their own valuable experiences.

Organizational Structure

Organizationally, highly experienced testers often are peers of the test management team. Figure 3 shows one possible organizational structure (in this organization, the highly experienced tester has the "Test Architect" title). The experienced tester in this case has influence and leadership responsibilities for the entire organization.

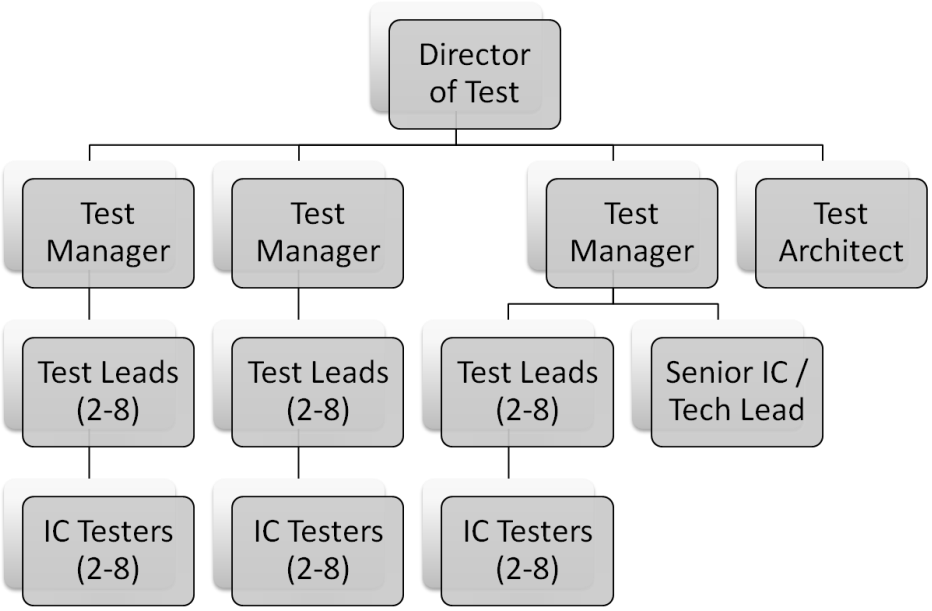

Figure 3 – Example Organizational structure

Figure 3 also shows where a Technical Lead role may be used both as a leadership role for a smaller part of the team, and as a growth path for testers who aspire to grow to a higher level of leadership and influence.

Note that this structure is common in larger organizations (30-40 or more testers). Smaller organizations may not have an individual with enough experience to take on a pivotal technical leadership position within the organization, but they often do have an individual who takes on some of the technical leadership duties in addition to her assigned testing tasks. At Microsoft,

most (but definitely not all) of the Principal SDETs work in large organizations with dozens – and sometimes hundreds of testers.

Hiring and Retention

Most teams would love to have a Jim, Lucretia, or Ed on their test team. Hiring people like these three is, however, nearly impossible. Testers like these become who they are through years of experience and challenging tasks. The role of management in creating testers like these is critical. First, testers need a proper mix of challenging tasks – work that stretches them beyond their current capabilities, balanced with work appropriate for their level of experience so that they can achieve success in their day-to-day work. It's the role of the manager to ensure the proper mix for everyone on the team. Remember that the day-to-day work for one tester may be the stretch assignment that another tester on the team needs.

Experienced testers also need opportunities for leadership. One thing in common with all three of the testers in this paper is that they all have led technical teams. Jim spent some time in his career as a manager, but in all three cases, leading "virtual" teams, e.g. leading a cross-group team in order to solve a common problem for all teams, or solving a problem for an entire test team by working with everyone on the team are necessary and typical examples of leadership exhibited by growing testers.

Finally, these experienced testers have a high degree of trust from their management chain. They're often asked to solve an extremely difficult problem for their organization, but have freedom to experiment when necessary and to choose *how* they solve the problem. Their managers know that they will ask questions or get feedback when they need it, but they also know that they will confidently make most of the decisions on their own, or with the virtual teams they build to help solve the problem.

The traits above are just as crucial for growing the next generation of highly experienced testers as they are for retaining the same. I don't know many people who would leave a job where they were valued, challenged, and trusted.

Conclusions and Next Steps

The influence of these three individuals on product quality is notable. The contributions of each one of these three have enabled their teams to accomplish much more than they could have without them. Additionally, they are leaders for their teams and are influential in helping to grow the senior testers of tomorrow. They all make their teams better through technical leadership, mentoring, and strong vision.

The data thus far is largely subjective, but planning is underway for more in depth case studies. The value of anecdotal data such as described previously in this paper is already having an impact across the company. At a grassroots level, the stories of roles of senior testers are already inspiring more people to stay in test positions. From the other side of the spectrum at the executive management level, there is a growing understanding of the types of value that experienced testers bring to an organization and there is an increasing desire to groom more testers to rise into these positions within organizations and sub-organizations across the company.

The impact of the three testers profiled above is undeniable, yet testers at their level represent just a fraction of a percent of the test population at Microsoft. As more testers, both at Microsoft and in the industry overall achieve extensive experience and technical knowledge, I expect the impact on product quality to become more and more significant. I believe it is just a matter of time before testers such as these become some of the most respected (and sought-after) members of the entire software ecosystem.