

# **MERCURY CUSTOMER PERSPECTIVE WHITEPAPER: BEST PRACTICES FOR IMPLEMENTING MERCURY QUALITY CENTER**

## ABOUT THE AUTHOR

Michael Giacometti is a Mercury TestDirector® CPC and a Mercury Certified Instructor with more than seven years of QA experience. He is currently a QA architect for a Fortune 500 financial services company. His responsibilities include creation and implementation of a company-wide QA process, creation and technical management of an Automation Center of Excellence, and the overall administration and implementation of one of the largest Mercury TestDirector sites in the world, with more than 2,500 users. Michael has created more than 25 applications to support Mercury TestDirector and has presented at conferences such as Mercury World and QAI USA. He resides in Southeastern Pennsylvania with his wife and daughter.

## TABLE OF CONTENTS

About the Author .....	2	Post-Purchase .....	10
Introduction.....	3	Implementing Mercury Quality Center .....	10
When to Make the Move to Mercury Quality Center .....	3	Recommendations and Advice .....	12
Mercury Quality Center: A Brief Overview .....	4	Training .....	12
Best Practices, Recommendations, and Advice .....	5	Political Issues and Considerations .....	13
Pre-Purchase .....	5	Summary .....	15
Surveying the Current Test Management Environment .....	5		
Evaluating Infrastructure Requirements .....	6		
Planning.....	8		

**Introduction**

The quality of your company's software has a direct impact on the quality of your company's financial results. You know it. Management knows it. And the importance of quality will only continue to grow with the need for 24x7 operations, high availability requirements, aggressive service-level agreements, and the need to roll out innovative new web-based services.

Yet at many companies today, there is still an enormous amount of inertia around implementing the tools, technologies, processes, and training to meet increasingly stringent application quality requirements. The evidence? According to the National Institute of Standards and Technology (NIST), 80 percent of the software development costs of a typical project are spent on identifying and fixing defects. And according to the Gartner Group, 50 percent of deployed applications are rolled back.

The core issue is not the tools companies are using to increase software quality – it's the lack of structured, consistent processes and best practices for implementing and using them. Even if your company has purchased the right tools for the job, those tools may be sitting on the shelf because the implementation process has not been clearly defined or agreed upon. Or you may be using only a small fraction of the features and functions available. Or you may be experiencing pockets of resistance to new approaches among certain individuals, groups, or even whole lines of business that insist on doing things "their way."

This paper cannot address every technical and political issue relating to application quality. What it can do is offer a structured approach to planning and implementing Mercury Quality Center™, an integrated, web-based suite of tools and best practices for automated software quality testing. The paper offers practical advice, based on my own experiences as a QA architect for a Fortune 500 company, for each phase of implementation. My goal is to facilitate broader acceptance of better tools and processes for delivering high-quality applications. I hope you'll find the paper useful as a first step on your road to improved application quality.

**When to Make the Move to Mercury Quality Center**

It's easy to determine whether your company has a significant issue with application quality. For example, you clearly need to make some changes if you have:

- Applications that are consistently late to market and over budget.
- Frequent application outages or missed service-level agreements (SLAs).
- A high percentage of development costs allocated to finding and fixing bugs.
- Problems with applications that result in constant development patching.
- Frequent rollbacks of applications from the production environment.

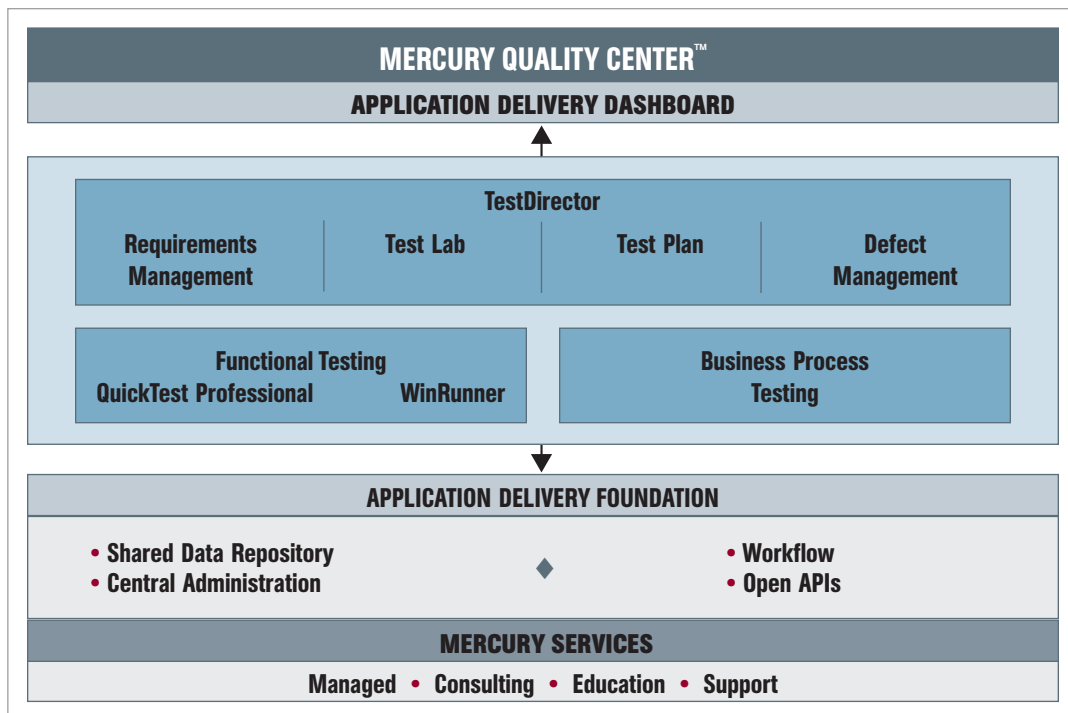
There are a number of other symptoms of poor quality management practices that, while less obvious, are equally significant to the organization:

- High levels of frustration/complaints from end users or customers.
- Low or declining morale among quality managers, QA professionals, and testers.
- Inconsistent use of tools and processes across organizations, leading to inconsistent quality levels in production applications.
- Excessive use of manual processes, signaling inefficient testing practices.
- Lack of alignment between quality practices and business goals.

**Mercury Quality Center: A Brief Overview**

For companies experiencing any of the problems or symptoms listed above, Mercury Quality Center offers an integrated suite of products and services that provide a solution.

Mercury Quality Center is a web-based system for automated software quality testing across a wide range of IT and application environments. It is designed to optimize and automate key quality activities, including requirements, test and defects management, functional testing, and business process testing. At all points in the process, you have visibility into where your quality project stands, so you can optimize software quality while managing and controlling risk as you develop and test an application. Mercury Quality Center includes industry leading products such as Mercury TestDirector, Mercury QuickTest Professional™, Mercury WinRunner®, and the new Mercury Business Process Testing™.



Mercury Quality Center performs automated software testing and QA across a wide range of IT and application environments.

### **Best Practices, Recommendations, and Advice**

Implementation of any new testing tool or process requires much thought and planning, and no matter how good the product is, it will quickly become shelfware unless planning starts during the sales process and extends well beyond the software delivery date. So whether your company is an existing Mercury customer or a prospective customer, it makes sense to work closely with Mercury sales engineers to make sure your requirements, questions, and concerns are understood and addressed at every stage. For the purposes of this paper, topics to be considered are classified as pre-purchase and post-purchase.

#### **Pre-Purchase**

##### *Surveying the Current Test Management Environment*

Mercury Quality Center installation and implementation may be less successful if the current situation is not surveyed, evaluated, and if needed, changed. One of the most common misconceptions when purchasing a test management or automated testing tool is that simple installation and minimal configuration of the tool will ensure successful use and implementation. The following questions will allow you to gain a clear understanding of your company's current test management environment and allow you to identify any changes that need to be made before proceeding with the purchase of Mercury Quality Center:

- **Which group, team, or organizations are involved with requirements, test, and defect management today?** For small to mid-sized companies, this may be a single centralized team. In larger organizations there may be separate teams. If each of the management types is handled by a separate team (with or without overlap) then identifying current methods of communications is critical to implementing Mercury Quality Center. Teams may have different communication standards. One of the largest challenges when implementing Mercury Quality Center is having all teams standardize on one communication type (wording, format, and use of items such as screen shots or other types of attachments).
- **How are items such as metrics or other types of reports gathered or communicated today?** Mercury Quality Center provides basic reporting capabilities. These are greatly enhanced with use of the Advanced Reporting Tool or Mercury Quality Center Dashboard™. External reporting tools can also be used with Mercury Quality Center (be sure to check with the Mercury Customer Service Organization (CSO) to determine if the tool is supported). If your company is using an external reporting tool (such as Actuate or Crystal Reports), investigate the amount of effort required or the feasibility of continuing to generate these reports with the external reporting tools after Mercury Quality Center has been implemented.

- **What quality, test, or defect management methodology does your company use today?** One of the most powerful features of Mercury Quality Center is that it is not tied to any quality, test, or defect methodology. While the chances of your company having to change its quality methodology are slim if implementing Mercury Quality Center, be sure to ask your Mercury sales engineer about creating rules or workflow to support your company's quality methodology. Can it be implemented? What type of resources and time would it take to implement? How hard will it be to maintain in Mercury Quality Center once implemented?

#### *Evaluating Infrastructure Requirements*

Mercury Quality Center increases quality and is ultimately more cost-efficient for your organization, but it does require an investment. Investing in Mercury Quality Center requires more than simply purchasing licenses for the application itself. Funds may have to be budgeted to create the environment and infrastructure that can support Mercury Quality Center according to your company's needs. The following checklist can be used to help determine not only your company's Mercury Quality Center licensing needs, but environment and infrastructure needs as well.

- **Evaluate your current test assets.** Test assets include not only the total number of test cases, but also the total number of test runs per testing cycle, the number of defects found during the application lifecycle, the number of requirements, and the number of users that are involved with the software development process (developers, analysts, testers, automators, management, etc.).
- **Use test asset data to determine your environmental needs.** The testing asset data that is gathered can be used to determine how many Mercury Quality Center licenses need to be purchased. A general rule of thumb is one Mercury Quality Center license for every five to six people using it. This license-to-user ratio may increase or decrease depending on the number of people using the tool and the times that the tool is going to be used. In other words, you may need fewer licenses if Mercury Quality Center usage is going to be distributed throughout a work day or more if usage is going to be concentrated during certain time periods. Test case, test run, requirements, and defect data will be used to help determine the type of operating system, database, and hardware to be used when implementing Mercury Quality Center.
- **Use test asset numbers to determine hardware, OS, and database types.** The Mercury CSO knowledgebase contains many articles and tools that can be used to determine the type of physical environment needed when implementing Mercury Quality Center. Mercury TestDirector was created for use on a Windows platform. There are two key criteria to keep in mind when designing your

physical environment for Mercury Quality Center: cost and performance. The physical environment used to implement Mercury Quality Center will more than likely be the same type that your company uses today. UNIX shops tend to want to install on UNIX while Windows shops will generally install on a Windows environment. Sticking to the minimum system requirements will ensure that Mercury Quality Center will perform adequately. The best practice is to use the Mercury Quality Center Installation Guide to identify the type of hardware, OS, and database you wish to use. Submit this environment scheme to your Mercury sales engineer or Mercury CSO. Send them not only your environment scheme, but also the test asset data gathered during your discovery process. Mercury can use this data to determine if the environment you are scoping will meet your needs.

- **At what times during the business day does the environment need to be available and does it need to scale?** Another common mistake that companies make when implementing Mercury Quality Center is to not take into account when the software is going to be used and if the team or organization is going to grow in the foreseeable future. When identifying the number of users during the test asset information gathering phase, also take note of when testing, development, or analysis occurs. A typical usage day for smaller companies would be between eight and 10 hours. However, for larger companies, these activities may go on 24x7. System stability and scalability is critical if the system needs to be available at any point during the day.
- **Where is Mercury TestDirector for Quality Center going to be used?** Mercury TestDirector for Quality Center is the first version of Mercury TestDirector that can truly be used on an enterprise basis. Companies have recognized the advantages of using Mercury TestDirector in a decentralized testing organization. As a result, implementation of Mercury Quality Center may differ for a company that will be using it throughout various sites as opposed to a single site. While some aspects of Mercury Quality Center usage may be the same for all organizations (e.g., allowing all users to access the client GUI from their workstations, or ensuring that corporate intranet policies allow for ActiveX execution on client PCs), there are some very distinct differences between small companies and large companies that use Mercury Quality Center. The more decentralized and distributed the usage of Mercury Quality Center, the more support and coordination is going to be required to ensure it is successful. When choosing to implement Mercury Quality Center at a large client with various locations, be sure to consider how it is going to be accessed. Is some sort of remote access service going to be required? Is the Mercury Quality Center web/application server going to have to be exposed on the Internet so workers can get to it no matter their location? What changes or

exceptions are going to be needed so workers can access Mercury Quality Center if they are not in the same Windows domain? Are you going to use LDAP? Will Mercury Quality Center allow LDAP access across these various domains? If Mercury Quality Center is going to be used in a distributed fashion across various domains, then it is best to use the center authentication instead of LDAP. It is important to remember that LDAP authentication cannot be used out of the box if the Mercury Quality Center web/application server is going to reside on a UNIX platform.

#### *Planning*

In addition to recognizing the type of physical environment and infrastructure Mercury Quality Center is going to require for implementation, anyone planning to implement the solution must also consider how it is going to be used and supported. Those familiar with Mercury TestDirector will recognize the classic features such as requirements management, test planning and remote execution, and defect management. Mercury Quality Center takes these classic features and extends them with not only the cross-module workflow scripting that first appeared in TestDirector 8.0, but also introduces the concept of Mercury Business Process Testing. This is a powerful feature that can allow resources to link requirements to test plans and then, with the use of Mercury QuickTest Professional, quickly automate selected test cases. Proper planning and use of these features will allow any user to take full advantage of Mercury Quality Center and recognize the full benefit of a centralized test management solution.

One word of advice: Mercury Quality Center provides many features and benefits that you may or may not need at your company. Plan or identify how you would use these features, but do not implement too much too fast.

- **Defect Management.** Identifying how defect management is performed at your company is one of the first steps in considering Mercury Quality Center. While Mercury Quality Center does not require a specific defect management methodology, it does require at least some basic planning and organization. Defect workflow or transition rules are not required for basic implementation. However, take advantage of the “System Fields” such as Defect Status, Project, Application, Assigned To, and Detected By fields. Use “User Fields” for items or classifications that are specific to your department or company. *Use of dropdown fields provides basic reporting capabilities and immediate return on investment.* Workflow or transition rules can be incorporated into Mercury Quality Center at a later point after it has been rolled out and used for a period of time.

- **Deciding whether to automate test plans – and how to go about it.** Automating test plans is usually the ultimate goal of anyone investing in Mercury's products and solutions. While determining what functionality or which application to automate is a major decision, deciding on storage and execution is just as critical. As with Mercury TestDirector, Mercury Quality Center provides a centralized repository that can store all automated tests and report on their execution status. Make one of the first tasks after implementation of Mercury Quality Center to move or save these automated test plans in the Test Plan Module.
- **Workflow: now or later?** The ability to place workflow in each module of Mercury Quality Center is its most powerful feature. Workflow can be created via Mercury Quality Center wizards, or by someone who is well versed in VB Script. Workflow should only be done during implementation if current defect or test transition rules exist at your company. Implementation can be delayed or negatively impacted if the decision to introduce and use transition rules is made while the transition is made to Mercury Quality Center. If your company chooses to introduce defect or test transition rules, then implement it after the center has been implemented and your user base has had time to adjust.
- **How is Mercury Quality Center going to be supported?** Mercury Quality Center is a rather easy application to administer, use, and maintain. However, a dedicated full- or part-time resource is going to be required to perform administration, maintenance, and support tasks. The qualities of this resource will be discussed later in this document. However, if this resource does not already exist at your company, you may have to hire people. In my experience, a part-time resource is required for any Mercury Quality Center environment up to 500 users. A full-time resource should be used for sites larger than 1,000 users. An additional full-time resource may be required for any 1,000 users added thereafter. In terms of concurrent licenses (assume a one-license-for-every-five-users ratio), a part-time resource should be used if your site has fewer than 100 concurrent licenses. We observed that full-time resources are generally required for every 200 concurrent licenses purchased. Multiple resources may also be required if Mercury Quality Center support and administration is required on a 24x7 basis. If Mercury Quality Center is going to be used in a large environment (greater than 1,000 users), then investigate if some of the more basic and frequently executed administrative tasks, such as password resets, can be assigned to a centralized helpdesk.
- **What not to do.** Mercury Quality Center can adapt to any defect or test management process very easily. Plan on what customizations will need to be made to the tool to allow it to adapt to your process. Any automated testing, defect, or test management process should be tool-independent.

**Post-Purchase***Implementing Mercury Quality Center*

Mercury Quality Center implementation is very much the same for small or large companies. This section deals with the common implementation tasks and then focuses on differences for small and large shops.

- **Common Tasks.** All implementation tasks break up into three areas: pre-installation, installation, and post-installation. The most important piece is to map everything out before you begin. Create a master task list that can be used to track tasks and ensure that no steps have been missed.
- **Pre-Installation Tasks.** As with the implementation of any piece of software, pre-installation planning and task execution is important. Mercury Quality Center's documentation does a good job of stating what the pre-requisites are. Make sure that the environment Mercury Quality Center will exist on is configured and operational before installing the Mercury Quality Center software. The system (front-end servers, back-end servers, load balancer, etc.) should be installed and configured. Database instances should also be loaded, configured, and if possible, tuned. If using a web/applications server other than JBoss, then ensure that is installed correctly before proceeding with the Mercury Quality Center installation. Before proceeding with the actual installation, review the Mercury CSO website for any last-minute updates to documentation or patches that could affect the installation.
- **Installation Tasks.** Installation tasks are items that are executed to install Mercury Quality Center software. These items are covered in the Mercury Quality Center Installation Guide. The time immediately following installation should be used for shaking down the environment to ensure that there are no issues with the installation. You may want to install Mercury Quality Center with the demo project as any problems found with the environment can be quickly diagnosed and addressed by either your internal experts or Mercury CSO. Performing shakedown testing after migrating your projects from a previous installation of Mercury TestDirector, or creating your own projects if this is a new install, may create added complexity and may not accurately reflect a problem with the Mercury Quality Center installation should you encounter any problems.
- **Post-Installation Tasks.** Post-installation tasks are performed after the Mercury Quality Center installation and shakedown testing have been completed. They include: the installation and testing of any add-ons or third-party tools (such as any external tracking tool plug-ins), automated testing tools (such as Mercury QuickTest Professional), or advanced reporting plug-ins. The largest task involved for users upgrading to Mercury Quality Center is the migration, modification, and updating of Mercury TestDirector projects. Depending on the number or complexity of the projects (workflow), this may be the most time-consuming step.

- **Small Shop Implementation.** Companies that are centralized with a small number of Mercury Quality Center users are considered small shops. The person who installs Mercury Quality Center is most likely the onsite expert, with access to all of the hardware and software in the environment, and this person generally does not support Mercury Quality Center 24x7. Small shops should concentrate their implementation plans on the actual installation and migration of Mercury Quality Center. Other environments that can be used for full-scale pre-installation testing or development may not be an option. If that is the case, keep in mind that Mercury Quality Center and Mercury TestDirector can exist on the same server, although they will need different IIS directories (TDBIN vs. QCBIN & SABIN). Any installation should be done during off hours and the installer should have a solid back-out plan in case there is a problem with the implementation.
- **Large Shop Implementation.** Large shop implementations require much more coordination of resources. Large shops may be centralized or decentralized, have large numbers of concurrent and named users, and want to take advantage of Mercury Quality Center's support for highly scalable systems with fail-over and load-balance capabilities. Large shops should take advantage of multiple environments if at all possible. Mercury Quality Center should first be installed in a staging environment where projects can be created (or migrated) and the implementation plan can be practiced and revised. Resources should be coordinated well in advance of the implementation date. As large shops may have specific departments dedicated to certain job tasks such as database engineering, system engineering, or system administration, it may become necessary to engage a project manager who can coordinate resources and budget. Environments in large shops will more than likely be more complex than their small shop counterparts; therefore, you should allocate more time to shakedown testing and installation testing.
- **Upgrading from Mercury TestDirector 7.x/8.x to Mercury Quality Center.** Companies are usually upgrading from Mercury TestDirector to Mercury Quality Center to take advantage of the expanded operating system and web server options. Regardless of whether or not Mercury Quality Center will exist on a new environment or replace the existing Mercury TestDirector environment, efforts need to concentrate on the upgrade and migration of the TestDirector projects. Mapping out and identifying tasks in the migration plan is just as critical as mapping out the implementation plan. Project migrations could be as simple as keeping the projects on the same server with the same database, or as complex as changing file system types and databases (e.g., Informix to Oracle, NTFS to UNIX file system). Map out your migration strategy and test it on copies of your largest projects. Be sure to not only consider the size of the project repository on the current file system, but also note the number of database records as this may extend the time it takes for the Mercury Quality Center migration tool to migrate the Mercury TestDirector projects to Mercury Quality Center. Research the Mercury CSO database for tips and updates on a Mercury Quality Center migration and open Service Requests with Mercury CSO as questions or issues arise.

*Recommendations and Advice*

Implementing Mercury Quality Center encompasses much more than installing the actual Mercury Quality Center software. Environment, resource, installation, and migration planning are critical to a successful implementation. Try to use the following tips:

1. Put the supporting hardware/software in place before installing Mercury Quality Center.
2. Map out and identify all implementation and migration tasks before starting the Mercury Quality Center install.
3. Try to have a test environment and a production environment for Mercury Quality Center. Use the test environment to practice installation and project migration before the actual implementation. Identify any issues and have workarounds ready so there are no surprises during the actual production implementation.
4. Create Mercury TestDirector 7.x/8.x copies of the projects to be migrated. Migrate the copies to Mercury Quality Center instead of the originals. This way, the original TestDirector 7.x/8.x projects can be used in case the implementation is delayed or cancelled due to unforeseen circumstances. The original projects will also be unaffected in case of problems during the project migration process.

*Training*

Mercury has made significant changes to the underlying Mercury Quality Center architecture, but few changes have been made to the end-user experience. Companies that are updating from Mercury TestDirector 7.x may notice more changes in the administration and client pieces of Mercury Quality Center. It then becomes important that the basic pieces of functionality are implemented first and the new functionality (traceability, follow-up flags, cross-module scripting) should be implemented over time.

Training is a very important part of the Mercury Quality Center implementation process. Money or time should be allocated to properly train users on the functionality before implementation or shortly thereafter. There are three types of training that are recommended depending on the number of users.

- **Send users to Mercury-sponsored training.** For companies with small numbers of users it may be more cost-effective to send a few of the users to Mercury-sponsored training. After training, the users can then be used to train their coworkers on proper Mercury Quality Center usage.
- **Bring a certified Mercury instructor onsite.** Companies with fewer than 100 users should bring a certified Mercury instructor onsite. The Mercury instructor can then hold training classes over a period of time. This is more cost-effective than sending a large number of users to Mercury training classes. Keeping the certified instructor onsite for a few days after training can ensure that the lessons learned are being correctly applied after Mercury Quality Center implementation.

- **Certify internal employees.** The most cost-effective option for companies with large numbers of users (> 100) will be to send one of their users to Mercury Quality Center certification and then to have this person certified as a Mercury instructor. Doing so may not only help the instructor's career path, but will also allow the users to be educated at a more reasonable pace at a lower overall cost.

If your company chooses to bring a certified Mercury instructor onsite, be sure to consider Mercury's partners in addition to the Mercury Professional Services Organization as it might save your company money.

#### *Political Issues and Considerations*

No Mercury Quality Center implementation will be successful unless it is a joint venture between your company and Mercury. The following tips can be used to ensure that the budget and resources are available to properly and successfully implement the solution.

- **Acquire the support of a senior manager early.** Identify a senior manager at your company who believes in the vision that Mercury Quality Center provides and can overcome any initial hurdles with funding or resources. If you have identified such an individual, set up a meeting between him or her and your Mercury account manager. If the meetings are successful, that senior manager can recruit the support of other senior managers.
- **Find an internal advocate.** Despite a well-thought-out implementation plan and flawless installation, Mercury Quality Center can become shelfware unless your company has an internal advocate onsite who can evangelize the center and Mercury's Business Technology Optimization (BTO) vision. Like any technology solution, Mercury Quality Center requires monitoring and maintenance to keep everything operating properly. The internal advocate serves two purposes. First, he or she can talk to any person in the company to clearly communicate what Mercury Quality Center is and how it can provide ROI or cost benefit. Second, the advocate has the technical prowess to identify possible problems and address them early. The earlier problems are identified and addressed, the more stable the system will be. The more stable the system, the more people will adopt it. Mercury Quality Center can sell itself if given the chance. An internal advocate is a person who is familiar with development, testing, and automation methodologies. This person should have a good amount of experience with Mercury TestDirector and/or Mercury Quality Center. If possible, this person should be certified as a Mercury TestDirector or Mercury Quality Center consultant.

- **Get onsite Mercury representation as much as possible.** If an internal advocate is not immediately identified – and even if one is – bring a Mercury account manager or Mercury authorized reseller onsite as much as possible. The Mercury account managers and resellers are easy to work with and will want to partner with you to ensure that Mercury Quality Center is being used properly and successfully. They are also experts and can quickly identify solutions if they perceive any problems or issues. The goal of a Mercury account manager is to build a relationship and a partnership, so use this relationship to the best of your advantage. The Mercury account manager is also good at creating visibility within Mercury CSO should your company encounter any problems with Mercury products.
- **Overcoming resistance.** Any shift in paradigm or new tool such as Mercury Quality Center is bound to cause some sort of resistance within a company. It is very important to overcome this resistance if Mercury Quality Center is to be successful in the long run. There are quite a few ways to deal with this. The first is to use the leverage of the senior manager. The second is to use the Mercury account manager. The third, and perhaps most compelling, is to install Mercury Quality Center and use it in a small setting (maybe a small team at first). Acquire a number of successes and tell everyone about them. Gain momentum by continuing to introduce Mercury Quality Center to more small pockets. Persistence is key in any strategy you choose when rolling out Mercury Quality Center.
- **Getting maximum bang for the buck (or quick victories).** The best way to show the benefit of Mercury Quality Center in your organization is to simply use it. Consider doing an internal proof of concept. While doing this proof of concept, you will want to do two types of reports. The first is a “user experience” report. This will show the benefits and disadvantages your internal users see between Mercury Quality Center and the old way of doing things. The second, which will get the interest of management very quickly, is to do a cost-benefit analysis of using the center vs. doing things the old way. Create categories such as:
  - Time it took to create a defect before and after Mercury Quality Center.
  - Time it took to create and execute a test plan before and after Mercury Quality Center.
  - Time it took to create reports before and after Mercury Quality Center.

Associate these time values with monetary values. The first trend you should see is that more work can be done in one hour with Mercury Quality Center vs. without. The second is that you should be able to associate a cost with adding/modifying a defect or test plan with and without Mercury Quality Center. Getting the numbers and figures in front of management will go a long way toward convincing them that Mercury Quality Center can provide the cost and time savings they have been seeking. Mercury account managers can provide you with cost/benefit analysis worksheets and templates at your request.

### **Summary**

All of us are interested in improving the quality of the applications we produce on behalf of our respective departments, lines of business, and enterprises. The best tools alone aren't enough; well-defined processes aren't enough; insightful plans aren't enough. What's required is a lifecycle approach to quality, structured planning and design of the environment, consistent use of best practices, state-of-the-art tools such as Mercury Quality Center, and of course a healthy dose of sensitivity to those sometimes-nebulous but ever-important factors known as "political realities."

The guidance and advice offered in this paper is far from exhaustive; it is simply intended to give you the benefit of my own experiences as a QA architect. I hope it answers some of your questions, sparks some new ideas, and takes you a step or two further on your road to a successful Mercury Quality Center implementation.

**MERCURY**<sup>™</sup>

Mercury is the global leader in business technology optimization (BTO). We are committed to helping customers optimize the business value of IT.  
[WWW.MERCURY.COM](http://WWW.MERCURY.COM)

This document is protected by U.S. and international copyright laws, and may be used only with Mercury Interactive Corporation permission. Features of the Mercury Interactive Corporation software, and of other products and services of Mercury Interactive Corporation, may be covered by one or more issued or pending United States or foreign patents.

Mercury, Mercury Interactive, the Mercury logo, LoadRunner, SiteScope, TestDirector and WinRunner are trademarks of Mercury Interactive Corporation and may be registered in certain jurisdictions. The absence of a trademark from this list does not constitute a waiver of Mercury's intellectual property rights concerning that trademark.

All other company, brand and product names are registered trademarks or trademarks of their respective holders. Mercury Interactive Corporation disclaims any responsibility for specifying which marks are owned by which companies or which organizations.

This document and its content are provided for informational purposes only. The opinions expressed in this document are those of the author(s)/contributor(s) and do not necessarily represent those of Mercury Interactive Corporation, its affiliates or its subsidiaries ("Mercury"). This document does not create a contract between Mercury and any person or entity ("You"). Mercury shall not be directly or indirectly liable to You for the document's content including, without limitation, any claims arising from 1) errors in the content, or 2) your use of or reliance on the content. Mercury shall not be liable for any damages or costs of any type arising out of or in any way associated with Your document use. Mercury retains all rights, title and interest in the document. The document and its content are presented "AS IS" with no warranty.

© 2005 Mercury Interactive Corporation. All rights reserved. [www.mercury.com](http://www.mercury.com).