

# Makin' Lemonade!

Maximizing Your Resources  
In a Sour Economy



benchmarkQA

Software Quality Assurance Consulting  
QA/Test Staffing • Training • Local Test Lab



benchmarkQA

## FOREWARD

*“When life gives you lemons, make lemonade!” When forced to do more with less, we have to become more efficient. The current economy requires many organizations to find ways to cut costs. The risk of this “belt tightening” in software development is that quality is all too often sacrificed to “save money” – by short-cutting good quality practices, reducing the test effort, jumping offshore without sufficient preparation... the list goes on.*

*At BenchmarkQA, we provide our clients with alternatives to prevent the potential erosion of quality within their organization – yet help them save money and become more efficient with the resources and funding already allocated.*

*The document, “Maximizing Your Resources in a Sour Economy” is a three-part report outlining key concepts to help you do more with less in your software development lifecycle. It is organized to illustrate the opportunities for efficiencies in three specific focus areas: People, Processes and Tools & Technology. Each focus area is dedicated to a separate white paper released in the fall of 2009.*

- 1. People Efficiencies - how to effectively reorganize the same work load to get higher performance from the great people you’ve already got (released in August 2009)*
- 2. Process Efficiencies - how to apply lean principles and simple process improvements to maximize productivity and reduce waste, resulting in a more profitable software development process (released in September 2009)*
- 3. Tools & Technology Efficiencies – how to reduce the time spent creating metrics and leverage tools & technology such as cloud computing, virtual imaging, and open source automation to get more done in less time (released in October 2009)*

*As a leader in testing and software quality assurance consulting for more than 22 years, BenchmarkQA has an extensive and in-depth understanding of the software industry. Having observed and formally assessed the challenges faced by our clients, we have developed this whitepaper to share our insights with managers and leaders of software quality assurance – to help you find areas to improve efficiency, reduce costs, and develop higher quality software.*

*I hope you find the ideas presented in this report to be very useful in helping you make the most of our current economic challenges.*

*Sincerely,*

*Molly Decklever  
Vice President of Operations  
BenchmarkQA*



# Makin' Lemonade!

## Maximizing Your Resources In a Sour Economy

PART THREE: CREATING EFFICIENCIES WITH TOOLS AND TECHNOLOGY

Betty Schaar, Sr. Process Consultant, CSQA  
Molly Decklever, Vice President of Operations

BenchmarkQA  
Minneapolis, MN

October 2009

### EXECUTIVE SUMMARY

This sour economy is forcing us all to find creative ways to do more with less. Creating high quality software with limited or shrinking resources caused by hiring freezes and down-sizing is a difficult challenge, especially when scope and deadlines aren't typically adjusted to reflect a reduced capacity.

This white paper contains creative ideas and tools used by our top consultants to help companies do a better job of managing their software development lifecycle to maximize efficiency and reduce defects; to cut costs while protecting quality.

We will provide key concepts to help you create tool and technology related efficiencies for your team:

1. Opportunity One – Automation
2. Opportunity Two – Cloud Computing
3. Opportunity Three – Virtualization
4. Opportunity Four – Combinatorial Test Tools
5. Opportunity Five – Metrics
6. Opportunity Six – Licensing



Decisions about which tools and technology to use to support your testing program have a strategic impact. Even when “free,” the technology you select shapes the options available to support your test efforts and establishes a path that requires the allocation of resources and time to follow. This report will provide some tips and ideas to inform your strategic technology and tools decisions and guide you toward the areas our consultants and our customers have found to be the most useful in reducing costs and improving efficiency.

## OPPORTUNITY ONE: AUTOMATION

The concept of automation has evolved and expanded massively in just the last few years. We will attempt to share some of the ways our clients benefit from the wide range of uses for automated testing, as well as the challenges and barriers that exist with automation.

In general, automation has recovered from the early years during which over confident tool sales reps and eager buyers lead to expensive shelf-ware. For the vast majority of those who tackle the effort with forethought and planning, automation has provided real return on investment—reducing test cycles and increasing coverage. However, our experts would caution, automation *rarely* reduces the total cost of testing—just shifts the cost to different resource types and tool licenses, and gives you more value for your investment.

Automation can replace mundane, tedious tasks—freeing up test resource time for more complex test efforts. Open source automation—in capable hands and for precise testing needs—has become a viable alternative to heavy front-end investments in tool licenses. Agile and other iterative development methodologies have forced automation into earlier unit and integration test efforts, delivering value by catching defects earlier in the lifecycle. Unfortunately this scenario will often merge a tester into the role of developer—armed with JUnit, Selenium, and the like to build test harnesses and stubs on an ad hoc basis. These efforts typically lack a long term strategy or automation architecture and are therefore unfortunately thrown away.

While there are reasons and ways to leverage automation for a quick hit return, our clients obtain the greatest success and value through automation efforts with a long-term strategic view, developed and executed as full-fledged projects designed to meet the unique needs, budget, and dynamics of the team, the internal systems, and the applications to be tested. Cowboy development is the bane of successful, maintainable automation. Choosing tools that won’t work well for the development languages and technologies of the organization is quite common.



Good coding standards, stable applications, and existing manual test cases are still the baseline conditions required to play in the big time with automation and win.

---

## OPPORTUNITIES TO AUTOMATE

Have you automated all that you can? Here are some guidelines to selecting the best tests for automation that pays dividends:

- Tests where results can be objectively assessed. If a test requires a subjective evaluation—such as for appearance or ease of use, it can't be tested with automation.
- Tests that are likely to be repeated three or more times with the same functionality. One common opportunity for re-testing the same functionality is to run a quick test of a new build—often referred to as a smoke test. Great return on effort can be obtained by creating this short series of automated tests, developed to gauge the general fit of the application and help determine if a build is sufficiently functional to make subsequent in-depth testing viable.
- Tests which require significant data input. Automation scripts can loop through a file of values with a script to reduce data input and allow for faster and broader data and calculation validation by increasing the number of variables and number of tests that can be run (and re-run) quickly.
- Test which are extremely tedious and/or simple. Simple tests are typically great candidates for automation (and with lower cost or open source tools). Automation of these easy items will help reduce the risk of inaccuracy from manual execution by testers glazed over from boredom, and allow you to allocate their intelligence toward more complex testing.
- Performance and stress tests. It is very difficult to measure performance, load, and stress without using a tool of some sort. It can be done but it's not pretty. We have a client who has had success using an inexpensive on-line tool to record end user functions, then parameterize the scripts and edit as needed for performance testing. These scripts are also executing the functionality of the application, serving a dual purpose of functional testing and performance/load testing for the most common features and end user scenarios. This is something to consider as a low-cost, quick, and relatively easy solution.

Automation scripts can be re-purposed as low-tech performance tests by creating a master script to initiate multiple automation scripts concurrently at the server to

emulate load. It is much more difficult to compile the data into meaningful results, and you will also need someone capable enough to write the scripts. However, it can provide a quick check (without getting a performance tool and creating new scripts) to ensure the system doesn't fall apart and to determine if a subsequent evaluation with a true performance tool is needed.

- Regression suites. A regression suite is a full series of tests which execute the core application functionality and can be run for each new build to ensure new features, enhancements, or any other changes have not caused any damage to the core application. With a defined regression suite that is stable from one build to the next, given the frequency of execution, automating a regression suite is a very common and highly cost-effective use of automation. In developing an automated regression suite, the focus needs to be on long-term re-use and ease of maintenance—therefore an automation strategy, automation architecture, and a seasoned automation expert are often required to ensure the effort is understood and the intended long-term value is obtained.
- Non-verification tasks that can be automated—such as data creation or test setup. One of our clients reduced test “execution” time by more than four hours per test by automating the test setup portion (example: they had to create a doctor, a patient, and a patient visit in order to “execute” the actual test). They used these same tests in the regression suite in each quarterly release—so the effort and frequency of test setup made this a worthwhile automation investment. In total, they reduced the test effort for that regression suite by nearly 50% per run, which allowed them to refocus the test effort on more complex new features that could not be automated, increasing their total test coverage.

---

#### IF NOT AUTOMATING AT ALL, ARE YOU READY?

Automation doesn't have to mean expensive tool suites.

If you need simple functional testing to be automated, consider using shareware/freeware, and open source tools such as Ruby on Rails, and test tools well suited to automating at the unit test level like JUnit, Nunit, and Selenium. A listing of some cheap and free test tools can be found at: <http://www.riceconsulting.com/cheaptools.htm>

Additionally, you might want to consider having your developers create utility scripts that can automate any number of otherwise manual testing related tasks. Some examples include scripts that compare output files, search for and select production data records based on specified criteria, or interrogate data files looking for records with a particular bad value.



If you are interested in building a maintainable test suite for smoke testing or regression testing to avoid buying shelf-ware, conduct a readiness assessment before you invest. BenchmarkQA offers an automation readiness assessment service, which has helped customers develop a strategic approach to automation to ensure they obtain their desired return on investment. For one client, we determined that their lack of consistency in development and coding standards would make automation difficult to maintain and therefore would require an investment in a senior level automation architect to regularly repair broken scripts. We have since guided the client to improve their coding standards and expand their manual test suite so that an automated regression suite is now a viable long-term investment. In the meantime we used inexpensive automation tools to conduct quick hit heavy data testing.

If you don't have an internal expert, engage a third party expert through a company like BenchmarkQA, who knows the tools in the market and can help you build an automation approach from a strategic vantage point.

## OPPORTUNITY TWO: CLOUD COMPUTING

Given that cloud computing is a rather new buzz word with a dynamic range of interpretations, we will establish a working definition of cloud computing and then describe some of the benefits of it related specifically to quality assurance and testing.

Cloud computing: "A model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g. networks, servers, storage, applications, services) that can be rapidly provisioned and released with minimal management effort or service provider interaction."<sup>1</sup>

Cloud computing is a revolutionary shift in infrastructure that can provide massive ROI. Proponents of cloud computing claim to have cut IT infrastructure costs by as much as than 90%.<sup>2</sup> Start-ups and established organizations alike have found they can flip the switch and gain scalability for minimal cost increases. By shifting their infrastructure to an external organization with a dedicated focus and expertise in IT, they gain not only a tremendous cost advantage, but also the potential for increased reliability, scalability, and control not easily obtained in an internal IT organization. When the New York Times sought to provide online access to their archive of over 11 million articles, the high price tag and months of effort they expected would be required to establish nearly four terabytes of storage. Cloud computing enabled them to complete the project in a matter of days at a cost of \$235 using Amazon Web Services instead of their own internal infrastructure and resources.<sup>3</sup>

How does this relate to quality? Cloud computing is an exceptional opportunity to save money and time which can be reallocated to quality improvement efforts (or other critical needs). For



example, the effort and resources that used to be allocated to procuring and maintaining infrastructure can be refocused toward improved build and release management practices (which may be needed to facilitate a successful shift to a cloud environment anyway).

Additionally, by working with external experts who are specialists, and working with a company with a dedicated focus on infrastructure, the potential for improved security, stability, and reliability may reduce the likelihood of quality issues associated with the infrastructure.

On the counter-point, the dynamic nature and lack of total ownership and control over the external cloud environment presents a whole host of new challenges for testing, which need to be factored into a decision to move to a cloud environment. BenchmarkQA's Quality Forum in November 2009 focuses on the implications of the cloud infrastructure on testing. Please look to BenchmarkQA's website for additional information on that topic in late November.

As cloud computing is not yet widely adopted, the following details might be useful in exploring the ways your company might leverage a cloud infrastructure.

---

#### ESSENTIAL CHARACTERISTICS OF "THE CLOUD"

There are several essential characteristics<sup>4</sup> that allow an environment or system to qualify as a "cloud." These include the following:

- **On-demand self-service.** A consumer can unilaterally provision computing capabilities, such as server time and network storage, as needed automatically without requiring human interaction with each service's provider.
- **Ubiquitous network access.** Capabilities are available over the network and accessed through standard mechanisms that promote use by heterogeneous thin or thick client platforms (e.g., mobile phones, laptops, and PDAs).
- **Location independent resource pooling.** The provider's computing resources are pooled to serve all consumers using a multi-tenant model, with different physical and virtual resources dynamically assigned and reassigned according to consumer demand. The customer generally has no control or knowledge over the exact location of the provided resources but may be able to specify location at a higher level of abstraction (e.g., country, state, or data center). Examples of resources include storage, processing, memory, network bandwidth, and virtual machines.
- **Rapid elasticity.** Capabilities can be rapidly and elastically provisioned to quickly scale up and rapidly released to quickly scale down. To the consumer, the capabilities available for provisioning often appear to be infinite and can be purchased in any quantity at any time.

- **Measured service.** Cloud systems automatically control and optimize resource use by leveraging a metering capability at some level of abstraction appropriate to the type of service (e.g., storage, processing, bandwidth, and active user accounts). Resource usage can be monitored, controlled, and reported providing transparency for both the provider and consumer of the utilized service.

---

## CLOUD DELIVERY MODELS

There are multiple delivery models for cloud computing, but three main categories<sup>5</sup> under which delivery typically occurs, as follows:

- Use provider's applications over a network
  - **Cloud Software as a Service (SaaS).** The capability provided to the consumer is to use the provider's applications running on a cloud infrastructure and accessible from various client devices through a thin client interface such as a Web browser (e.g., web-based email). The consumer does not manage or control the underlying cloud infrastructure, network, servers, operating systems, storage, or even individual application capabilities, with the possible exception of limited user-specific application configuration settings.
- Deploy customer-created applications to a cloud
  - **Cloud Platform as a Service (PaaS).** The capability provided to the consumer is to deploy onto the cloud infrastructure consumer-created applications using programming languages and tools supported by the provider (e.g., java, python, .Net). The consumer does not manage or control the underlying cloud infrastructure, network, servers, operating systems, or storage, but the consumer has control over the deployed applications and possibly application hosting environment configurations.
- Rent processing, storage, network capacity, and other fundamental computing resources
  - **Cloud Infrastructure as a Service (IaaS).** The capability provided to the consumer is to provision processing, storage, networks, and other fundamental computing resources where the consumer is able to deploy and run arbitrary software, which can include operating systems and applications. The consumer does not manage or control the underlying cloud infrastructure but has control over operating systems, storage, deployed applications, and possibly select networking components (e.g., firewalls, load balancers).

---

## CLOUD DEPLOYMENT MODELS:

To be considered a “cloud” environment, it must be deployed on top of cloud infrastructure—such as those stated below, which also has the essential *characteristics* of a cloud stated earlier.<sup>6</sup>

- Private Cloud - Enterprise owned or leased
- Community Cloud - Shared infrastructure for a specific community
- Public Cloud - Publicly available mega-scale infrastructure (i.e. Amazon, Google, etc.)
- Hybrid Cloud - A composition of two or more clouds

## OPPORTUNITY THREE: VIRTUALIZATION

Virtualization is directly relevant to quality assurance and testing, as virtual environments allow for faster test environment creation and more efficient test environment management. Virtual environments use software such as VMWare and Hyper-V to mimic or emulate a physical environment, greatly reducing server and storage overhead. Virtual environments can be hosted either internally or externally.

Many of our clients have struggled to establish reliable test environments in an efficient manner. Perhaps few things are more frustrating or a bigger waste of time than completing a test effort only to find that the effort was conducted in an irrelevant environment or with a release or build that is out of synch. This costly mistake is less likely to happen when using test environment management tools.

If you are not already taking advantage of the efficiencies of a virtual test environment and you or your leadership team need more convincing to invest, consider these top three reasons to adopt virtualization software:<sup>7</sup>

1. Reduce physical infrastructure cost
  - Companies achieve an average of 10:1 server consolidation ratios
  - Hardware and software costs can be reduced by more than 50%
2. Reduce data center operating costs—ecologically wise
  - Power – fewer servers demand less power; demand-driven power down/up
  - Cooling – less servers generate less heat
  - Manpower – less time provisioning, configuring, monitoring, etc.

### 3. Minimize lost revenue due to downtime

- Eliminate planned downtime
- Recover quickly from unplanned outages

Virtual test environments are a fraction of the cost of the real thing and the efficiency and ease of use saves additional time and money. However, there are limitations to using a virtual environment for testing, which can make virtual environments an unacceptable substitute. A customer evaluated several defects that were found in production and determined that these defects should have been caught by testing that was conducted in the virtual environment—tests which had passed. These same defects were reproduced as a defect in a physical test environment. The customer continues to use virtual test environments for the vast majority of their testing, but has since isolated the specific characteristics or features of the application which require a physical environment to ensure accuracy.

The decision to go virtual needs to be made carefully, with the constraints of your infrastructure, application, end users, and product in mind. We recommend consulting an expert and working with an experienced tool provider to get you on the right path. The benefits associated with reduced test environment infrastructure overhead and reduced support costs make virtual environments an area of tools efficiency worth evaluating.

## OPPORTUNITY FOUR: COMBINATORIAL TEST TOOLS

“Combina-what” you say? We mention combinatorial test tools to bring light to a largely unknown tool type which supports efficient test coverage, thereby making testing more efficient and effective. These tools are algorithm-based decision table tools which return a set of the highest impact combinations of values, when testing for multi-value scenarios. The goal of the tool is to give the user the test set which will provide the best coverage with least number of tests.

These tools are based on “Pairwise” theory, which asserts that most software faults are caused by the interaction of, at most, two factors.

Many tools are available and several are free. BenchmarkQA evaluated three free tools—Jenny, PICT, and FireEye. In our analysis, we obtained three slightly different combinations for a simple MM, DD, CCYY test of boundary values. While none of the resulting sets were “wrong”—the variance in outcome indicates you will likely need to apply your own logic to validate the resultant set and get comfortable with the outcomes of the tool before you rely on it for critical efforts. However, given the exponential nature of testing multi-value scenarios, these tools



might be worth looking into to determine if they can help you quickly develop more efficient test coverage.

#### OPPORTUNITY FIVE: EVALUATE METRICS AND THEIR SOURCES

A good metrics program (meaning the tool, the processes, and the management of the entire effort) can help coordinate and inform team members, support good decision making, assist in future project planning and task estimating, and aide process improvement efforts. Bad metrics programs can inhibit team work, reduce productivity, and create an expensive distraction from the real work at hand. Here are some questions we invite you to consider to ensure your metrics program is on the right track, based on common trouble spots we have found.

1. Do your tools support easy metric collection? The time spent to collect, scrub, and report metrics is lost productivity. If metrics are important to your organization, having tools that support efficient collection and management of quality metrics is typically money well spent.
2. Are the tools and/or metrics used driving unwanted behaviors? We have witnessed “loophole” field values that allow a bypass of the standard process (such as a value “already tested” or “emergency fix” which can be entered by a developer to circumvent testing). If you have loophole field values, make sure they are not being abused.
3. Are the metrics feared as individual performance measurements? If you have a culture where people are scared of metrics, they will find ways to manipulate the data.
4. Does each metric provide critical information? Who uses the information and how does it help your company? In asking these questions, you may find there are a few metrics that are “interesting” but not truly useful to the success of your efforts. The effort to collect and manage these metrics could be significantly streamlined to focus on the metrics that are used to manage your success, drive positive change and allow you to shift the effort from collection to analysis, conclusion, and action.

#### OPPORTUNITY SIX: LICENSING

Licensing is an area with great potential for “found money.” Here are some ideas and tips from our experience helping customers reduce license costs.

1. Multiple teams who purchase the same product separately might be able to get a volume discount if they combine their purchases into a larger bundle.
2. Evaluate the tools you are actually using to see if you have tools that are no longer used and for which you are still paying license or maintenance fees.

- Website that has a software management guide and links to license audit tools, some of which are free:  
<http://www.bsa.org/country/Tools%20and%20Resources/All%20Tools.aspx>
3. Evaluate if you are using all the licenses you purchased. If you have fewer users than the number of licenses paid, request a reduced fee.
  4. Ensure that licenses associated with decommissioned hardware has been accurately accounted for and released.
  5. Different tools from the same vendor often have “package” discount rates, which you can request to be retroactively applied. In this economy, tool vendors are highly motivated to keep customers happy.
  6. It is a great time to buy. If you are in the market for a new tool or expanded license set, ask for discounts and negotiate on the rate. If you missed a promotional period, ask them to extend it. You might be surprised how much they will accommodate a buyer in times like these.
  7. If you do not have budget for a tool but really need one, consider shareware/freeware tools.
    - Websites that have tool listings, including freeware/shareware tools:  
<http://testingfaqs.org/>, [www.stickyminds.com](http://www.stickyminds.com)
  8. Re-evaluate the price tag on your tools and licenses. Are you getting the value you are paying for? If you are not confident that the investments you have made in application licenses are paying off, make some noise. Make your vendor work for you and help you get the value you expected. If the lack of value lies in your internal use, consider investing in training or mentoring to support effective use—but set specific goals for the training (i.e. after training we will be able to use the advanced features including report creation with features x, y, and z) so you can ensure the training closes the value gap. Sure, the training is an additional expense but if it helps you get the value and results you need from your software, it is money well spent. If you can’t swing the cost, make sure you at least take full advantage of the vendor’s help desk and user manuals and see if there is a local user group or product blog. These can be great resources and ways to gain issue specific assistance.

## CONCLUSION

Using tools and technology to enhance efficiency is an obvious solution, but we all know it is not the silver bullet we often want and need. The tools are only as useful as the processes and people that use them.



We hope we have given you some new ideas to help you find the loose change in the couch cushions, and perhaps pointed out some technology road blocks you can now avoid.

Do your homework, research your options, or consider using a third party expert to do the legwork. It would be our pleasure to help you gain the greatest possible efficiency from your investment in tools and technology to support your commitment to quality.

## FINAL SUMMARY

This is the third and final release of a three-part whitepaper. All three releases and valuable supplements and reference materials are available to our clients, as well as those who attended our August 2009 “Makin’ Lemonade” Quality Forum. For additional information, or to request access to these supplemental materials, please contact our office (contact information on the last page).

Supplements and reference materials provided include:

- Organization Skill Analysis
- Candidate Interview Evaluation Form
- Common Test Case – Boundary Value
- Test Checklist – Generic Window
- Document Reviews as a Form of Static Testing
- Requirements Checklist

There is a silver lining in a recession. There are opportunities to take advantage of—such as having motivated tool vendors ready to provide great discounts to win your business and being able to gain exceptional talent at good rates.

Thankfully, these economic conditions don’t come along too often—but now that it is here—don’t let a good recession go to waste. Use this time as an opportunity to become lean, streamlined, and more efficient at bringing good quality software to market. Get rid of the waste in your processes, and reapply those resources toward improving your product and getting to market faster.

We hope this whitepaper series has provided useful information. It is our goal to be of service to our clients and our industry and it is our privilege to help companies become more efficient and effective.

We wish you the very best in your efforts to “make lemonade!”



benchmarkQA

**Notes**

<sup>1</sup> <http://www.csrc.nist.gov/groups/SNS/cloud-computing/index.html;%20NIST-cloud-def-v14.doc>

<sup>2</sup> <http://www.itworld.com/saas/56362/whos-getting-roi-cloud-computing-now?page=0%2C3>

<sup>3</sup> <http://www.itworld.com>

<sup>4</sup> <http://www.csrc.nist.gov>

<sup>5</sup> <http://www.csrc.nist.gov>

<sup>6</sup> <http://www.csrc.nist.gov>

<sup>7</sup> <http://vmware.com>

**Disclaimer**

*The information contained herein is provided by BenchmarkQA and is intended to provide general information on a particular subject and is not an exhaustive treatment of this subject. Accordingly, the information is not intended to constitute professional advice or services. It should not be relied upon as the sole basis for any decision which may affect you or your business. Before making any decision or taking any action that might affect your business, you should consult a qualified professional adviser.*



### **About the Authors**

*Molly Doyle Decklever, Vice President of Operations*

*With a comprehensive background in the people and process side of technology, Molly Doyle Decklever brings her expertise in project management, process development, training, and IT staffing to this BenchmarkQA Quality Forum event. As vice president of operations, Molly is responsible for managing in-house testing projects for BenchmarkQA's testing lab and recruiting consultants and employees of BenchmarkQA. Prior to BenchmarkQA, Molly held various process management and IT talent management positions at Nash Finch Company.*

*Betty Schaar, BenchmarkQA Senior Consultant and Training Practice Lead*

*Betty Schaar is a 23-year veteran dedicated to improving quality assurance. A highly respected leader, Betty has earned the designation of Certified Software Quality Analyst (CSQA) through the Quality Assurance Institute and has held QA leadership positions in major Twin Cities corporations including United Healthcare, Kodak Health Imaging, and Imation. Betty has also consulted at a number of companies spanning a variety of industries. Betty's QA expertise has been highlighted through speaking engagements at national conferences including the Practical Software Quality Techniques (PSQT) and Software Testing Analysis and Review (STAR) conferences, and through her leadership and participation in TCQAA and PSQT.*

### **About BenchmarkQA**

*BenchmarkQA is a software quality assurance consulting firm in Minneapolis, Minnesota. For over 22 years, BenchmarkQA has been helping software project teams deliver higher-quality software through consulting, staffing, training and testing services.*

*BenchmarkQA's Quality Path™ service is a start-to-finish quality assurance strategy. It begins with a comprehensive Quick Audit™ assessment which evaluates how your people, processes and tools & technology are utilized and managed, and identifies issues and opportunities to enhance the quality and reduce the cost of software you create.*

*For additional information on BenchmarkQA services, please visit our website at [www.benchmarkQA.com](http://www.benchmarkQA.com) or contact us at 952.392.2400.*