Offshore Software Test Automation: A Strategic Approach to Cost and Speed Effectiveness

Synopsis
The “Offshore Software Test Automation” white paper presents an executive overview of the innovative approach to integrating global outsourcing and the latest test automation methodologies and tools. This approach can effectively help any software development organization meet their goals of time-to-market, cost containment and quality.

To provide the proper background for this discussion, the paper also summarizes the software development process, the software testing process, global resourcing and the automation of software testing based on a structured approach and methodology known as Action Based Testing (ABT).

This paper is based on the principles examined in the book Global Test Automation: A Discussion of Software Testing for Executives (http://www.logigear.com/about_us/books/gta.asp).

Introduction
Two industry trends — automating software testing and moving software testing offshore — encourage savings in both cost and time. Bringing software to the market faster and reducing costs are very high on any organization’s software development wish list.

However, many of these efforts yield results that fall far short of expectations as demonstrated by the media reporting such failed offshore attempts. But despite such failed exposure, offshore software test automation does succeed under a coordinated management effort with sufficient planning and preparation.
The competing goals of delivering a quality software product, reducing costs and meeting deadlines to market targets often lead management to take a tactical approach, focusing entirely on one goal or approach while neglecting to consider the impact of their decisions on other parts of the process. They may simply focus on one dimension such as automating testing to improve time to market or offshoring testing to drive down costs.

Taking a tactical or one dimensional approach can very often lead to undesirable or unforeseen results. Management needs to consider the interplay between quality, cost, and time-to-market goals to construct a more strategic approach and develop an overall strategy and finally select the tools and partners.

An effective test automation effort requires:

- Thoroughly comprehending and appreciating the software development process.
- Recognizing software testing as a strategic effort that can provide both cost and speed benefits, as well as providing management with visibility into the quality of the software under development.
- Budgeting and funding software testing separately from development.
- Prioritizing a structured test automation approach based on a methodology known as Action-Based Testing (ABT), which in turn creates a hierarchical test development model improving the quality and speed of testing and reduces costs.
- Appreciating the strategic importance of global outsourcing strategies to further drive down costs and impact the bottom line.

This white paper will:

- Provide an executive overview of the software development process and when to perform testing and automated testing for management to attain informed and effective decisions on test automation and global outsourcing.
- Discuss the strategic integration of the latest test automation methodologies and technologies such as Action-Based Testing (ABT) to fully capitalize on the speed, cost advantages, and best practices in automation and global sourcing.
- Demonstrate the importance of selecting an experienced strategic partner such as LogiGear, who can provide an integrated testing solution based on ABT and a global outsource strategy.
Executive Overview of the Software Development Process

At its most basic, there are three general categories in any software product development life cycle:

1. Specification or requirements definition
2. Development
3. Testing

This simple list and the prevailing view of the process, implies a linear relationship between each of the components - the process moves from specification to development to testing as illustrated in Figure 1.

![Typical View of Development Process](image)

This is an overly simplistic view of the process of developing software that focuses on the tasks while ignoring the strategic interplay between the three parts of the process. Unfortunately, this is how the process may be implemented in many companies. There are several problems with this task-focused view:

- It assumes that the specification delivered will not change.
- It assumes engineers will deliver software to testers on time and that there will be no changes.
- It involves testing far too late in the process.

It is essential to understand that each step in the overall product development life cycle is its own discipline and process. Further, each process is substantially enhanced if it is informed by and continuously involved with the other steps in the process. In reality, the steps in the development life cycle should both overlap and provide feedback to each other. For example:

As product marketing/planning develops a requirements document (hopefully with extensive customer and user input), engineers investigate alternative ways of implementation, develop initial implementation estimates, identify critical paths, and translate requirements into a workable engineering plan. A non-linear approach grants clear advantages for all departments involved to efficiently and effectively collaborate throughout the entire process.
- Engineers continuously communicate requirements on feasibility and risks, as well as deadline estimates for implementing certain features and functionality which may trigger alterations to the requirements specification.

- Testing at early stages provides visibility into what will need to be tested expediting test planning and test cases development.

- Early collaboration grants testing to design and automate test cases in advance ready for implementation once development delivers code.

- Testing in the initial stages enables feedback, time estimates, risk assessments, and it identifies essential areas to implement the requirements. Testers can even ask development to put hooks into the software that will make the test process easier and faster.

Figure 2 illustrates the more parallel nature of the process with the invaluable feedback loops. The strategic integration of the three disciplines into a more cooperative process helps to improve quality, improve delivery time and reduce costs.
How Quality Fits

Many mistaken quality assurance with software testing—it is possible to do an excellent job of software testing and still deliver a poor quality product! Poor requirement definitions or implementations can lead to a product with unwanted or unusable features. Testing can verify that such a product is relatively reliable and works as intended, but no one may want it or use it. Such a “working” product will be seen as poor quality by users and potential buyers. Delivering a product like this can have a very negative impact on an organization in driving up its costs and potentially impacting current or future revenue streams.

It is important to understand that quality entails efforts at every step in the development life cycle. It is also important to understand the costs of quality throughout the process, and that costs differ depending on where in the process they occur.

Quality cost, or the total cost to deliver a quality product, is simply the sum total of all quality related costs incurred at every phase of a project. There are four broad categories of quality costs:

1. **Prevention costs** - Prevention costs represent costs spent to prevent errors in software, documentation, and other product related matters. These costs include staff training, requirements analysis, early prototyping, defensive programming, usability analysis, clarity of specification, and more. Prevention quality costs are the most cost-effective quality dollars that a company can spend. In essence, it costs much less to avoid errors up front than it does to later identify them, determine how to fix them, and then develop new code to rectify them.

2. **Appraisal costs** - Appraisal costs represent funding that is spent on testing activities including any and all activities associated with searching for errors in software and associated materials. The cost encompasses testing done by developers themselves, as well as the testing performed by the software testing organization. It is one of the largest costs associated with quality.

3. **Internal failure costs** - Internal failure costs are associated with the expenses fixing bugs found prior to release.

4. **External failure costs** - External failure costs are costs associated with defects and errors that are discovered after the product is released. These costs include all of the direct costs of identifying and fixing the problems under pressure, as well as all of the sales and marketing costs associated with damage control. There are also intangible costs such as loss of goodwill, low customer satisfaction, and impact on future sales. External failure costs are very costly, and are typically much higher than internal failure costs.

As demonstrated above, it is much less expensive and disruptive to fix problems internally prior to product release than it is after the product is released to customers. External failures present the very real risk of negatively impacting an organization’s profitability. The challenge to management is to do a cost-benefit analysis to achieve an optimum balance between prevention and appraisal costs to help to minimize failure costs, reduce overall costs, and positively impact the bottom line.
Executive Overview of Software Testing

There is the common assumption that software testing is a small extension of development and that testing is performed exclusively by software engineers. Another misconception is that software testing occurs at the end of the development process after the software is developed. The software is then tested and shipped for release. Others equate quality assurance and software testing. All of these conjectures are incorrect.

Software testing is a highly strategic and specialized discipline. It is very different from—yet related to—software development (just as sales and marketing are very different but related disciplines). At its most basic, software testing is a concerted attempt to break software so that bugs may be identified and fixed before end-users encounter them. Testing uses skills, methodologies, insights, and creativity that are very different from those used by software developers.

Software testing is an important part of the overall development process. Because of its strategic significance, software testing should be its own organization separate from development with its own budget. This setup enables:

- Effective training
- Continuous skills and career development
- Efficient implementation of testing tools
- Increased effective communication among the testing, development and management team

The software testing process has four main activities:

- Designing software tests
- Running tests
- Identifying problems and defects
- Reporting to management and development on key metrics

Testing is an iterative process of identifying, fixing and retesting, as well as reacting to design and code changes. Software testing consists of both manual and automated testing. While it is true that manual testing should be kept to a minimum; there will always be tasks, such as usability testing, that require manual tests.

Software testing is not quality assurance; it is a part of quality assurance. All steps in the development process, from requirements definition to design and development share quality as an objective. The role of software testing in the process is to identify defects so that they may be fixed. You cannot,
however, test quality into a poor design (as the saying goes, “bugs may be tested out, but quality must be built in”).

Software testing is also not something that simply happens at the end of the development process. To be effective, software testing needs to be a strategic partner with product marketing and development from the beginning. Software testing needs to have a clear and full understanding of the goals and objectives of the software under design and development. Having such knowledge will help the department design better tests.

In addition to being a process, software testing generates products that can be viewed as valuable assets to an organization for future projects:

- Test cases can consolidate the intellectual property of team members.
- Automated tests that can be re-used, becoming assets that reduce costs.

Software testing is also a strategic resource that provides a major benefit to senior decision-making executives. Software testing supplies the product development and its maturity status for release or production. In essence, software testing is an information service for the executive team allowing them to make more informed and effective decisions regarding the product’s quality and readiness. Software testing provides these measure using small number of well chosen metrics to help track the testing progress and the quality of the software.

### Testing Metrics Provide Visibility

Testing Metrics provide visibility into a product’s quality. Metrics are only useful if they help to make sound business decisions. Metrics fall into two categories:

1. Project management
2. Process improvement

Some metrics can measure the output of your test team, such as:

- Bug reports
- Test cases written or test cases executed

Other metrics, such as coverage metrics, exhibit how much of the software has been tested.
Executive Overview of Test Automation

Test automation provides great benefits to the software testing process and improves the quality of the results. The reasons to automate software testing lie in the pitfalls of manual software testing:

- Slow and costly
- Scales badly
- Inconsistent and repeatable
- Difficult to manage

While these factors may drive the desire and need to automate testing, it is important to take the right approach to test automation. There are several basic steps to automating testing:

- Implement the Action Based Testing (ABT) methodology, a keyword-based, object oriented approach that provides visibility, reusability, scalability, and maintainability ensuring speed and cost efficiency.

- Select appropriate enabling technologies that support the methodology. The tools need to support extensibility and a team-based global test automation framework with a solid management and communication platform.

- Designate team members with the proper skills and training in methodology, tools, and domain knowledge (i.e. knowledge of the software to be tested, its industry, and end-user expectations).

- Separate test design from test automation so that automation does not dominate the test design. Action Based Testing (ABT) creates a hierarchical test development model that allows test engineers (domain experts who may not be skilled in coding) to focus on developing executable tests based on action keywords, while automation engineers (highly skilled technically but who may not be good at developing effective tests) focus on developing the low-level scripts that implement the keyword-based actions used by the test experts. ABT allows an organization to spend more time developing tests and less time actually coding the test cases. This speeds up the whole testing process and helps reduce costs.

- Lower costs by using less expensive labor than your local team.

- Jumpstart the process with a pre-trained outsourcing partner that knows more about test automation success than you do, and that has a competent, well-trained staff of software testers, automation engineers, test engineers, test leads and project managers.

Automated Testing: Productivity Objective

No more than five percent of the effort surrounding testing should be expended in automating the tests.
The most essential element is methodology. The methodology is the foundation upon which everything else rests for it drives tool selection and the rest of the automation process. The methodology encourages offshoring the appropriate pieces of the testing process.

**Executive Overview of Offshore Outsourcing**

Most companies are convinced of the need to offshore some or all of software testing. Offshoring offers the promise of significant cost savings. However, offshoring is more than simply moving existing software testing efforts to an offshore outsource partner. Executives making the decision to offshore testing must understand the possible challenges of outsourcing and offshoring, and must prepare strategies to combat these potential problems. Executives also must stay focused on the fact that an effective offshored testing effort is based on the strategic integration of methodology, the latest technologies, and an effective global resource strategy that supports the methodology and tools.

Some of the major pitfalls of offshoring include:

- Problematic communications due to language and cultural barriers, mismatched or miscommunication of expectations, poor metrics selection, and unresponsiveness.
- Insufficient or contradicting skill sets in the software testing organization, such as using entry-level development engineers as software testers, lack of knowledge of the software being tested, and lack of domain knowledge.
- Management issues from lack of a workable test management process and associated methodology.
- General offshoring risks such as security and protection of intellectual capital.

Some suggestions for dealing with these obstacles include:

- Finding a trusted testing partner or building trust in a testing partner. You need to work with a colleague that you know has testing experience, an experienced staff, an understanding of current methodologies, and competent domain knowledge.
- Train the test organization. Provide them with knowledge of your application, your expectations, your communication/management platforms and expected domain expertise. Discuss how to recognize and deal with cultural issues.
- Adopt a process and toolset that support the overall methodology to improve testing, defect tracking, automation, and communications management, focusing on excellent and correct methods, team communication, accessibility, and useful measures.
- Choose carefully what work is sent offshore and what remains “at home.” Often it makes sense to keep user-focused scenario development and business process testing at home where you have more knowledge of the domain and the user.
• Get someone local to manage the offshored test effort. A local lead that is part of your team, who understands the culture and communication nuances of the offshore team, can lead the project, effectively communicate progress and metrics, and help streamline the process.

The most effective way to avoid the dangers of test automation and offshoring, and to realize the full time and cost saving benefits of test automation, is to implement a strategy of Global Test Automation.

Global Test Automation strategically integrates the latest test automation methodologies based on Action Based Testing with an effective and balanced global resource strategy that makes use of both onshore and offshore resources and leads, and effective team-based management tools and methodologies.

The benefits of such an approach are scalability, reusability, visibility, and maintainability that ultimately will allow an organization to achieve both time and cost savings while delivering a quality software product. The bottom line is a higher quality product that is delivered faster and more cost effectively.

Choosing a Global Test Automation Partner

Selecting the right testing partner to implement this strategy is more than simply selecting a vendor to develop and run test cases. It entails selecting a partner who fosters innovation to improve test productivity understands Global Test Automation, recognizes that the methodology drives the use of appropriate tools, possesses an effective management and communication strategy, and optimally uses global resources to provide the appropriate mix of on and offshore resources to drive down costs.

The process of test automation can be expedited with the selection of a pre-trained strategic partner that knows more about test automation success than you do and that has a competent, well-trained staff of software testers, test leads, and an in-place global resourcing strategy.

LogiGear has the people, technology, and methods to help you double your test coverage, decrease testing time, improve product quality, and cut costs!

An innovative turnkey global test automation solution ... LogiGear is not only such a strategic partner; LogiGear represents a best-of-breed partner for software test automation. LogiGear’s strengths as a test automation partner include:
• Innovation - A thorough understanding of Global Test Automation integration with industry recognized leadership on the topics of Global Test Automation, Action Based Testing, and all aspects of software test automation. LogiGear’s senior management and staff are published professionals, active instructors at major universities, and regularly present at test industry events.

• Transformation experience - Extensive experience implementing Global Test Automation with both small and large software development organizations.

• Test productivity optimization - Experienced in the challenges and resolving such obstacles for test automation.

• Software testing expertise - The ability to implement a global outsourcing strategy that makes use of staff expertise fully trained in all aspects of software testing, coupled with outsourcing locations that minimize cost and risk. LogiGear software testers, both on and offshore, are all LogiGear employees and software testers by training. They continuously develop skills and are efficient in the implementation of test tools.

• Global delivery services - The ability to implement a blended onshore/offshore test lead approach helping to mitigate communications and cultural issues, as well as improve both project management and reporting.

• Experience - Experience working in a globally distributed testing environment, including use of advanced automation for internationally distributed teams.

• Collaboration for success - Experienced in the two-way exchange of methodology, tool, and management knowledge to the customer, and the transfer of product and domain knowledge to the offshore testers.

In LogiGear, you will be selecting a strategic test automation partner that will help you hit the ground running, implementing a global test automation solution that delivers both cost and time savings.