Association for Software Testing

CAST 2006: Influencing the Practice

June 5th-7th, 2006 – Indianapolis

http://www.associationforsoftwaretesting.com/conference/index.html

Tutorial

Strategies and Tactics for Global Test Automation

Hung Q. Nguyen

Monday June 5, 2006
9:00am - 2:30pm

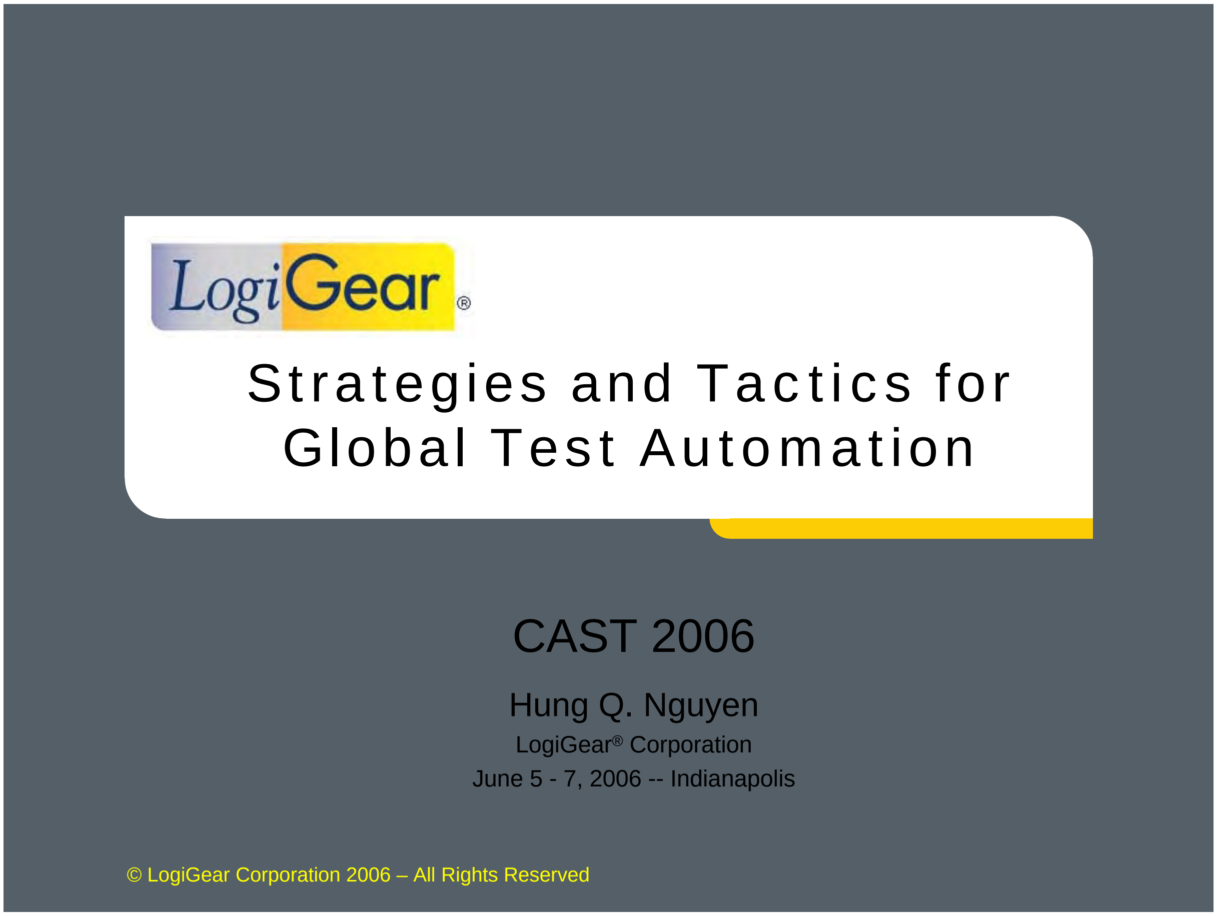
Room: Victoria Station C

About Hung Q. Nguyen

Hung Nguyen. CEO, President, co-founded LogiGear in 1994, and is responsible for the company's strategic direction and executive business management. He leads the company's innovative approach to software testing, test automation, methods and tools, and education programs. Currently with over a hundred engineers, LogiGear provides testing solutions to clients ranging from the Fortune 500 to early-stage startups. It works closely with its customers to determine their exact software quality testing goals and challenges, then design unique solutions based on LogiGear's

- Onshore/Offshore Testing & Test Automation Services
- Testing & Test Automation Methods and Tools
- Software Testing Training
- Software Testing Process & Strategy Consulting

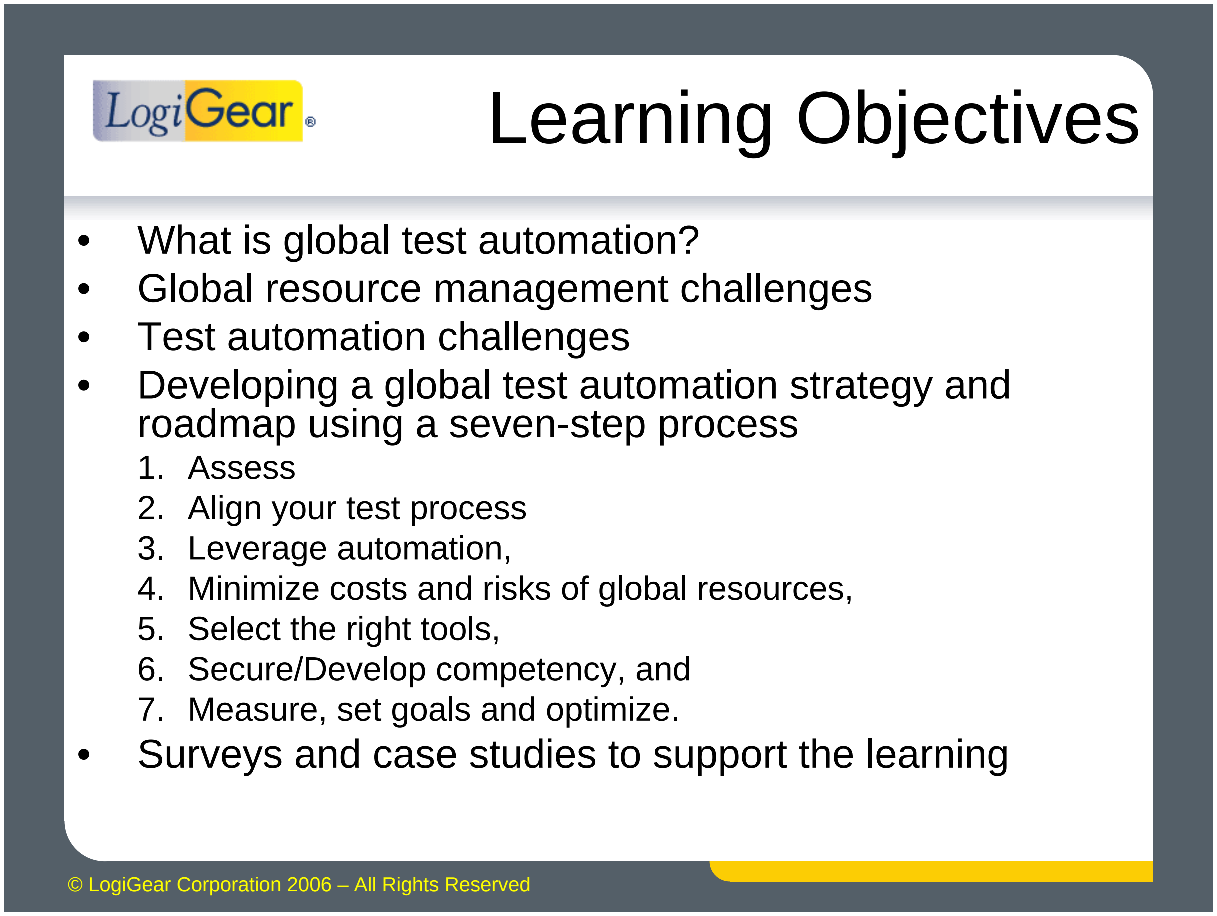
Mr. Nguyen is coauthor of the top-selling book in the software testing field, Testing Computer Software (Wiley, 2nd ed.) and other publications including Testing Applications on the Web (Wiley, 2nd ed. 2003). His experience over the past two decades includes leadership roles in software development, quality, product and business management at Spinnaker, PowerUp, Electronic Arts, Palm Computing and other industry-leading companies. A frequent speaker at industry events and a contributor to industry publications, Nguyen also teaches software testing at LogiGear University, and the University of California Berkeley Extension and Santa Cruz Extension. He holds a Bachelor of Science in Quality Assurance from Cogswell Polytechnical College. Nguyen co-founded the Association for Software Testing and currently is its Director At Large.



Strategies and Tactics for Global Test Automation

CAST 2006

Hung Q. Nguyen
LogiGear® Corporation
June 5 - 7, 2006 -- Indianapolis



Learning Objectives

- What is global test automation?
- Global resource management challenges
- Test automation challenges
- Developing a global test automation strategy and roadmap using a seven-step process
    1. Assess
    2. Align your test process
    3. Leverage automation,
    4. Minimize costs and risks of global resources,
    5. Select the right tools,
    6. Secure/Develop competency, and
    7. Measure, set goals and optimize.
- Surveys and case studies to support the learning

Before We Start

About surveys and case studies:

We surveyed senior managers and engineers in a wide range of software companies and inquired about their views and war stories related to the topics. We will share some inputs we received on each subject. We are sharing with you the benefit of learning from their experience.

What is Global Test Automation?

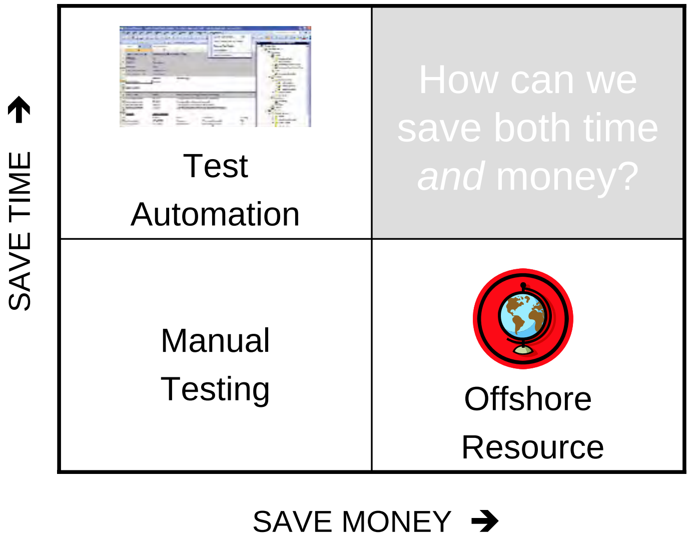
Global Test Automation (GTA) is an integration of the latest test automation methodologies and technologies with global resource strategies to fully capitalize on the speed and cost advantages of best practices in automation and global sourcing.

What is Global Test Automation?

GTA structured approach is based on a methodology known as Action Based Testing (ABT) which creates a hierarchical test development model. ABT 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) to focus on developing the low-level scripts that implement the keyword-based actions used by the test experts. This seamlessly makes for the best utilization of the skill sets of your staff back home as well as offshore staff. The ABT methodology has been proven for over a decade in Europe, and has now gained traction and popularity in the US. We will discuss more later.

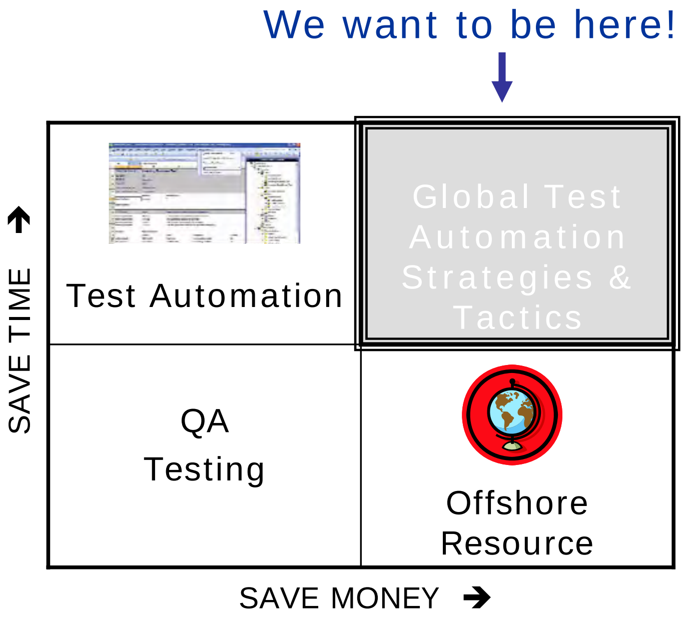
The Challenge

1. Software testing is necessary.
2. Software testing takes time.
3. Software testing costs money.



The Objective

1. High Speed.
2. Low Cost.
3. Uncompromised Quality.



Global Resource Management Challenges

Top 5 Offshore Testing Challenges

- Problematic communications.
- Insufficient or mismatched skill sets at software test organization.
- Management issues.
- Vendor and infrastructure problems.
- Offshoring risks.

Global Resource Management Challenges

- Problematic communications
  - – Cultural (both ways)
    - Non-responsive
    - Non-communicative
    - Creativity or lack there of
    - Versatility or lack there of
  - – Time zone
    - Slow speed of communication
    - Interminable workday
  - – Language barrier
  - – Poor management of two-way communication and expectation

---

Surveys & Case Studies

Good communication and people management are keys to success.

D. Bechtel, Intellisync Corporation

Overview

I manage a development organization responsible for several product initiatives at Intellisync. I have teams located in San Jose, CA and in Eastern Europe. I am part of the senior management team that first brought the notion of outsourcing/offshoring to Intellisync.

Our own experience began by using a small number of contract offshore QA and Development resources. This has evolved into a fully operational and integrated development, QA, and support organization in Eastern Europe with more than 100 employees capable of supporting several key products. We've learned a great deal in the past several years. Clearly a lot of what we have learned is of a technical nature; however, a significant amount of what has been learned (and the area I believe impacts most organizations) relates to cultural, communication, and planning areas of the operation. Intellisync embraces this concept. In today's competitive landscape this is just one more tool that allows us to remain competitive while we continue building award-winning software along the way!

Surveys & Case Studies

Case

Communication is vital in any organization. Everyone in the Software industry has experienced situations where a lack of communication with the person just down the hall leads to delays and additional expenses incurred during the product development lifecycle. Worse yet, sometimes you are not even aware there is a communication failure until bugs appear during testing. Even the most experienced US-based software professional may miss or lose track of important issues due to miscommunication with other stakeholders or team members. Now, take this scenario and expand it to one or more remote teams, in different time zones, with varying skills and English comprehension. You are looking at a real potential challenge. For example, a situation where the US-based product manager assumes those testing the product have a strong understanding how the product will be used by the end-user lead to several key failures in how the product was being tested by this group. In yet another example, a team assumed the specifications were complete when in fact they were continuing to evolve. Things tend to change rapidly in our industry. Failing to keep key product requirement and design documents up to date resulted in quality issues being discovered very late in the schedule. Desired quality levels were eventually achieved but the delays in the release cost the company real dollars in sales and additional staffing costs.

There are also cultural issues to contend with. For example, working with a contract organization in Eastern Europe, I experienced several instances of team members at this site simply telling you what you wanted to hear in order to avoid any sort of direct conflict (slipping schedules, disagreement on approach, etc.) with other team members or even the local management this person might work for. This resulted in unnecessary delays. Being the bearer of bad news is not comfortable (and in some cases simply not possible) for someone whose culture has experienced life under communist dictators like Stalin, Zhivkov, and Ceausescu.

Surveys & Case Studies

Case (cont.)

I understood the significant competitive value this operation could bring Intellisync, a public company, through lower cost of development and increasing our ability for “around the clock coverage” testing of our products. As such, communication and cultural challenges were taken very seriously by our organization. We didn’t always “see it coming” but over time our organization developed a sense of what it takes to manage these issues. This isn’t simply a new “task item” for your offshore team. This is real change for your US-based folks as well. Learning to ask questions in a non-threatening way helps with true disclosure and building trust. Willingness to adjust your own work schedule to help accommodate a remote team, and setting aside time for daily (yes, daily) status calls, IM chats, and other forms of communications helped reduce this churn and miscommunication. It is important to note that part of our action plan was and is to ensure we have adequate US representations working with any offshore or remote team. Lastly, since changing the culture of any organization is difficult, we felt it was extremely important to establish a long-term presence and commitment to the folks we were working with by establishing a formal entity in the region.

Our local and remote teams are working well together at this point. We believe there is always room for improvement and our teams support and encourage this. Frequent visits between local and remote groups have further increased the ability for these teams to work together (really, the ability to “read” reach other) and solve difficult challenges. Our quality levels continue to increase, our costs are under control, and we continue to expand our capabilities into other areas such as automation using these existing, well-trained teams.

Surveys & Case Studies

How likely would you be to drop a service provider if they use offshore service reps with poor English skills? (345 responses)

Not likely: 9%
Somewhat likely: 9%
Very likely: 39%
Done deal, they're dropped!: 43%

Source: Information Week's Outsourcing Pipeline newsletter

Global Resource Management Challenges

- Insufficient or mismatched skill sets at software test organization.
  - Lack of QA/test competency (1)
    - Test design skills
    - Bug finding skills
    - Bug reporting skills
    - Bug analysis/isolation skills
  - Domain expertise
    - Knowledge of product under test (1)
    - Domain knowledge in the category of the product under test (2)
  - Technical knowledge (1)
  - Others (Varies among regions) (2)
    - Lack of versatility or not multitasking-aware
    - Lack of systematic approach
    - Lack of creativity
  - Skill mismatch—Want to code; don’t want to test! (1)

(1) Not unique but the problem is exacerbated
(2) Unique

Surveys & Case Studies

Carefully assess the testing staff against the requirements for domain expertise and industry experience.

Mark Tezak, Adobe Systems

As some of our projects are being outsourced, the tester will need experience in two industries in order to properly test my particular areas. There are admittedly very capable engineers overseas schooled and experienced in computer science, but their domestic industries, in this case printing and publishing, has not yet been able to provide them with a comparable level of sophistication found in the western economies.

Surveys & Case Studies

Competent staff is a necessity.

Steven Yang, MathScore

While at Ask Jeeves, I worked in the Boston office, but we had QA resources in Emeryville, CA. As every developer knows, nearly 100% of talented software professionals avoid QA jobs, which means few QA professionals are very competent. As a result, interacting with QA staff can be very time consuming, and sometimes negatively productive by dragging down otherwise-productive development resources. In my personal experience, there was one particular QA professional who reported lots of bugs and assigned them to me. Unfortunately, most of the time, the bugs really weren't bugs, and the QA person simply didn't understand the product well enough to know why they weren't bugs. In many months, this QA person never found a bug of significance. I wasn't the only person frustrated by this person. The consensus among developers was that the company would have been better off if this person had never been hired in the first place. If I were to hire QA resources, I'd find a talented developer and pay him a disproportionate amount of money in exchange for the misery of doing QA. One talented developer on QA is probably more valuable than a conventional team of 10 QA engineers who aren't very competent.

Global Resource Management Challenges

- Management issues
  - – Lack of visibility and control over the business process and distributed test teams
  - – Poor management of test products or artifacts
    - Share and control the integrity of test artifacts including test requirements, test designs and test scripts
    - Share test results
  - – Lack of a test management platform and methodology to:
    - Track and manage the work done by the remote team
    - Easily generate progress report metrics to control testing activities, schedules and risk

---

Global Resource Management Challenges

- Management issues (cont.)
  - – Big management drain on US staff
  - – Lack of trust
  - – Savings not worth the hassle!
    - It's not a one-to-one saving
    - Volume does not reach critical mass to achieve meaningful savings
  - – No clearly set goals
    - Lack of measurements
    - Lack of analysis to evaluate effectiveness
    - Lack of improvement initiatives
  - – Inadequate preparation and training for both US and offshore staff

Surveys & Case Studies

"Outsourcing software testing is like any other kind of outsourcing: if you use a careful selection process and you are willing to invest in managing the relationship, it will work well. If you toss the problem over the wall and expect the outsourcer to 'do the right thing,' it will fail."

— Francoise Tourniaire, owner of FT Works

Surveys & Case Studies

Close collaboration is needed.

Shyamsundar Eranky, Symbol

Most of the QA work that I have seen in software projects is contracted out – either outsourced to another company that supplies resources to do in-house testing or completely moved to another country. The latter case is not always effective because QA engineers need to collaborate closely with the development team.

Surveys & Case Studies

US companies have not properly anticipated and planned for how much oversight and management is needed to make offshoring positive.

The savings gained through offshore outsourcing are worth the additional management challenges.
(118 responses)

Strongly Agree: 14%
Agree: 17%
Unsure: 11%
Disagree: 26%
Strongly Disagree: 31%

Source: Information Week's Outsourcing Pipeline newsletter

Global Resource Management Challenges

- Vendor and infrastructure issues
  - Few best-of-breed offshore testing vendors
    - There are many all-in-one outsourcing shops but few are specialized in software testing with credible expertise and a track record.
    - Development and testing activities are not the same. Many shops try to do everything, and one can't be expert on everything.
    - Many offshore shops take pride on being CMM-certified
      - Being CMM certified is a plus but does not reflect strong testing capability. Good testers are required to have strong testing skills, domain expertise, technical and communications. Acquiring the right skill sets is essential.

Global Resource Management Challenges

- Vendor and infrastructure issues (cont.)
  - Poor bandwidth
    - Infrastructure, from uninterrupted power supply to network bandwidth is essential for any offshore facility.
    - There have been numerous stories of builds taking so long to transfer to an offshore facility due to lack of bandwidth that time efficiencies have been lost.
  - Service disruptions dues poor back-up plans for outages.
  - Attrition problem drives higher cost.
    - Attrition is a growing problem, particularly at the middle manager level.
    - Staff turnover is/can significantly set back any project and team.

Global Resource Management Challenges

- Offshoring risks
  - Time to productivity
    - Team integration takes time
    - Ramping up takes time
    - Training takes time
  - Security
    - Protection of intellectual property
    - Protection of confidentiality
    - Protection of physical property

Surveys & Case Studies

Outsourced testing strategy needs to be thoroughly assessed.

Martin Zwilling, Director of Technical Services, Fujitsu Software

For outsourcing of software testing, if manual testing is proposed as the primary methodology, the effort will likely to fail or be ineffective at any price. Manual outsourced testing is inherently inconsistent, non-repeatable, and too slow to be competitive in today's rapidly changing technology. Look for automated test tools, organizational certification in a recognized process and strong employee credentials in software development and testing technologies.

Top 5 Recommendations

1. Build trust or use someone you can trust.
2. Train the test organization or make sure that they are competent to begin with.
3. Get a methodology or tool to improve communications.
4. Choose carefully what work to send offshore and what to keep at home. In most cases, you have a higher level of programming skill in your offshore team than in the domestic team.
5. Get someone else, local and part of your team, to manage the offshored test effort and to focus on what each team does best.

Test Automation Challenges

Top 5 Test Automation Pitfalls

1. Uncertainty and lack of control.
2. Poor scalability and maintainability.
3. Low test automation coverage.
4. Poor methods and disappointing quality of tests.
5. Technology vs. people issues.

Overview

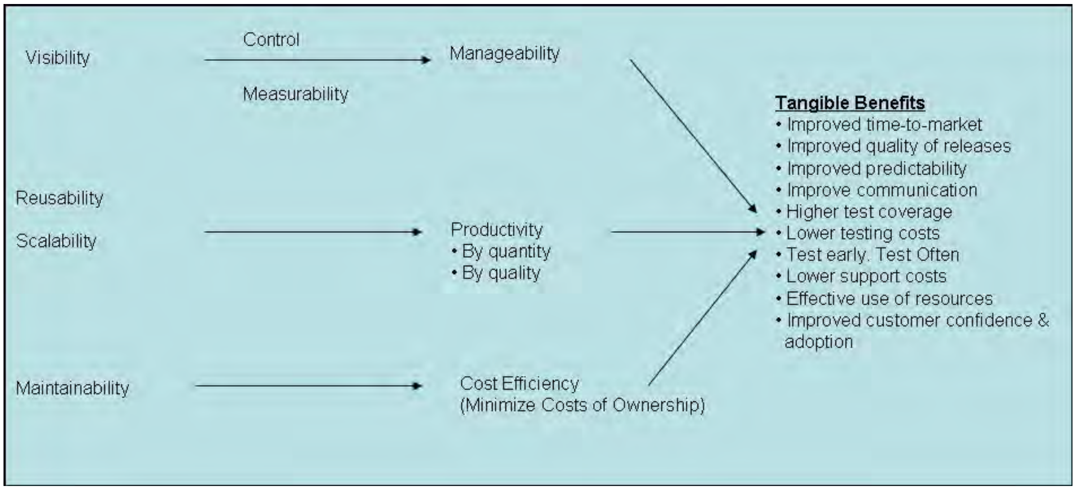
Test automation provides great benefits to the software testing process and improves the quality of the results. It improves reliability while minimizing variability in the results, speeds up the process, increases test coverage, and ultimately can provide greater confidence in the quality of the software being tested. But there are some pitfalls to be aware of.

No Silver Bullet

- Automation is not a silver bullet. It also brings some problems. The key to success is first define the test methodology, then choose the right enabling technology to help you implement the methodology. The chosen methodology should provide the following:
  - Visibility
  - Reusability and Scalability
  - Maintainability

  More discussion methodology later.

Outflows of Test Automation



Visibility

Control

Measurability

Manageability

Reusability

Scalability

Productivity
- By quantity
- By quality

Maintainability

Cost Efficiency
(Minimize Costs of Ownership)

Tangible Benefits
- Improved time-to-market
- Improved quality of releases
- Improved predictability
- Improve communication
- Higher test coverage
- Lower testing costs
- Test early. Test Often
- Lower support costs
- Effective use of resources
- Improved customer confidence & adoption

LogiGear® Test Automation Challenges

- Uncertainty and lack of control make automation a risky investment.
  - Companies know that they need to automate as much as possible but aren't sure how to make it successful.
  - Companies want to achieve high-degree of automation without incurring high total cost of ownership but aren't sure how to make it happen.
  - The uncertainty of how to effectively implement automation coupled with the wishful thinking of a plug & play tool normally leads to an ineffective plan. Cost savings are not realized.

LogiGear® Test Automation Challenges

- Uncertainty and lack of control (cont.)
  - Most automation tools lack an easy reporting mechanism with visibility into what automation is doing, what tests are being written, run, how often and what are the results.
  - Testing and test automation is a process. To successfully manage a process, you need to be able to quantitatively and qualitatively assess the activities and progress throughout the testing lifecycle. You need a well designed yet simple process, as well as a tool that offers the appropriate metrics to bring visibility of testing and test automation activities.
  - Management of technical teams. Many test leads lack the technical skill sets to both provide technical leadership to test developers and solid proficiency in the process to manage and control production. Test Leads trained in managing manual test projects need different skills, not only technical but management to run and communicate an automated test effort.

Surveys & Case Studies

Communicating the value of test automation is both challenging and critical.

William Ufheil, Manager of Quality Assurance, CDW

One of the challenges a QA team faces is the need to *constantly* sell the 'value added' proposition that automated testing can bring to the IT organization. The value in test automation has proven to go far beyond just testing. As IT teams have begun to realize what the QA team can do, we have used our automated functions as the building blocks for constructing training regions, populating data for developer testing, and even running scripts for the input of larger amounts of data to resolve production issues.

Surveys & Case Studies

Build buy-in from the rest of the organization.

Eric Rupprechtmindex, Quality Manager,
Mindex Technologies

Test Automation needs to be continually sold to internal groups (management, development, support, etc.) to educate other teams on the benefits and methodologies used to develop and maintain automation. Once all teams are educated on the cost of delivery and the automation team is staffed appropriately, automation will become a valuable tool in your software lifecycle.

Surveys & Case Studies

Test automation must be carefully planned. A hasty test automation effort will increase, not decrease, your testing headaches.

Lee Cunningham, Release Manager, Beeline

In our organization, poor project planning and cavalier attitudes in development resulted in a huge backlog of testing issues on the back end of the release cycle. When quality issues went undetected into production, the testing group was blamed, and was given the ultimatum to 'automate your testing'. What resulted was a library of hastily-written scripts, the effectiveness of which has yet to be proven. After going through several 'QA Managers', it is finally beginning to sink in that automated testing can't reverse the effects of a defective process.

Surveys & Case Studies

Test automation must be carefully planned. A hasty test automation effort will increase, not decrease, your testing headaches.

Justin McKanna, QA Manager, MYOB, LTD

Test automation adds real value. Unfortunately, upper management thought it would be as simple as making a cup of tea; add hot water, stir and 10 minutes later , voila - instant solution to all of our quality issues. You have to ask yourself if you are prepared for the investment of time, resources and money that automation requires. There is a large planning exercise that needs to happen. Jumping in at the deep end is not the time to START gathering your requirements - that should have happened a long time before you begin the execution.

Surveys & Case Studies

Test automation must be carefully planned. A hasty test automation effort will increase, not decrease, your testing headaches.

Garry Batt, Principal Software Engineer, EMC

Automation engineers are always under pressure to produce test scripts and very often fail to architect a test environment. This usually leads to little re-use of code and very little synergy within a team. Sit down with your team and come up with a strategy and plan. Set a vision. Architect an environment that you can deliver quickly, but is extensible over time.

Test Automation Challenges

- Poor scalability and maintainability makes test automation a costly investment.

  A simplified method for calculating the cost of test automation:

  Cost of test automation
  = Cost of test tool
  + Labor costs of script creation
  + Labor costs of script maintenance

  Labor costs of script creation is much more difficult to quantify because it depends on

  - The automation framework and methodology
  - Having the right balance of people who design tests (good testers may not necessarily have good programming skills) and people who create automated test scripts (good programmers don't necessarily have good test domain expertise and test design skills)
  - How likely the created test scripts will break when the application under test changes.

Surveys & Case Studies

Start with a clear definition of what successful test automation means.

Phil Woollven, QA Manager, FTSE, The Index Company

Automation provides a value added service to the test cycle, but should never become a dependency. Combined with strategic manual testing the quality of the test cycle is greatly enhanced working with automation.

Test Automation Challenges

- Low test automation coverage
  - When asked how many test cases are actually automated, most organizations will report figures in the range of 20-30%, or less. This has to do with how much work it is to automate a test case, and to keep the automation up to date with the latest system changes, as well as the sheer amount of test cases with a script for each test case. Even though some common functions, like “logon”, might be common to more than one script, having many test cases means a lot of scripts to create, manage, and maintain.
  - The low automation coverage is generally perceived by management and other participants as disappointing, in particular when much time and money have been invested and expectations, often based on shiny sales demonstrations, were much higher.

LogiGear® Surveys & Case Studies

Automate as much as possible.

Shyamsundar Eranky of Symbol

Wherever possible, automated test suites have to be created. This not only ensures that your software/product does not regress across releases but also helps in improving code coverage ensuring better quality.

LogiGear® Test Automation Challenges

- Poor methods and disappointing quality of tests
  - An unscalable automaton framework or architecture such as “record and playback” as well as high-maintenance scripted test automation leads to poor reusability.
  - Software Development cycles are so fast the team has no time to effectively automate, especially not reusable scripts.

Surveys & Case Studies

Test automation success starts with good process.

John R Lee, Director, System Test,
Salesforce.com

Most companies miss the most valuable component of automation: The process.

Surveys & Case Studies

Pick an appropriate methodology for your unique testing needs.

Mukesh Jain, Quality Manager, Microsoft

It's hard to test every possible combination. With Model Based Testing, we were able to find the right set of combinations and wrote automated for it—which helped us find the right bugs before our customers bumps into it.

Test Automation Challenges

- Technology vs. People issues--The tool will not solve the problem
  - Automated testing is software development. It needs process, procedures, design, architecture, standards, conventions, reviews. Few organizations have the availability of resources to commit to this type of effort.
  - No good tool for all of your needs
    - The promises of what automation tools can do, particularly in the self-running nature of test creation is grossly misleading.
    - Even some of the better automation tools have bad communication features, with reporting mechanisms that need the purchase of other tools to clean up their lack in reporting. Therefore the automation and communication need different tools.
  - Tool is expensive comparing to benefit it delivers—with limited benefit and high maintenance costs substantial benefit from automation programs is rarely realized. This would be bad for business as is but limited success is better than tools winding up as shelfware, which is the most common result with a total loss of benefit. With such limited benefit, the cost of the tools is never recovered.

Surveys & Case Studies

A testing tool alone won’t solve your testing problems.

Gael Le Bihan, QA Manager, Coradiant

We spent a few months evaluating various test automation tools. We didn't find any that works perfectly for our needs but settled on a good price/feature compromise. With deeper development, we ran into many dead-ends. It seems that our developers always found a better automation tool that ended up with a different dead-end after a few weeks of trial.

Surveys & Case Studies

A testing tool alone won't solve your testing problems.

Ronald Fierens, Product Testing & Services Manager, DYMO Corporation

Don't expect that one tool can do it all. In many cases you will need a 'test automation toolbox' with a wide variation of tools.

Test Automation Challenges

- Technology vs. people issues (cont.)—Test engineers are not using the tool.
  - Lack of training or fear of new technology,
  - Many companies use external, pre-trained teams or individuals.
  - Many teams never get the opportunity to implement an automation program no matter how well designed because of the time investment needed and the adverse effect this would have on current projects.
  - With the high maintenance nature of most automation programs there is often a defeatist attitude on automation teams.

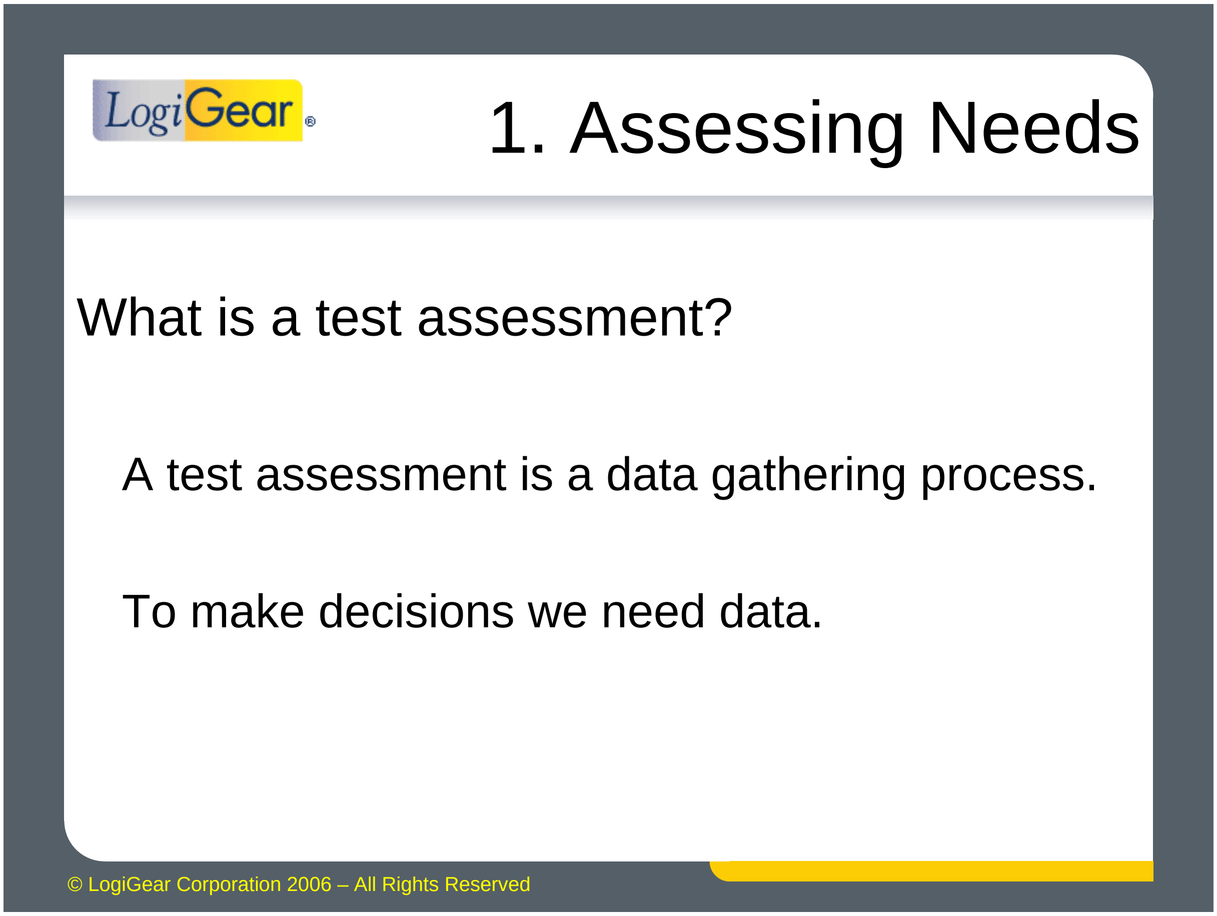
Top 5 Recommendations

1. Focus on the methodology, not the tool.
2. Choose extensible test tools: Select a test tool that supports extensibility, team-based Global Test Automation framework (team members are distributed), and offers a solid management platform.
3. Separate test design and test automation but make sure that tests are *automation-ready*.
4. Lower costs.
5. Jumpstart with a pre-trained team.

Strategies and Tactics

Developing a global test automation strategy and roadmap using a seven-step process

1. Assess
2. Align your test process
3. Leverage automation,
4. Minimize costs and risks of global resources,
5. Select the right tools,
6. Secure/Develop competency, and
7. Measure, set goals and optimize.

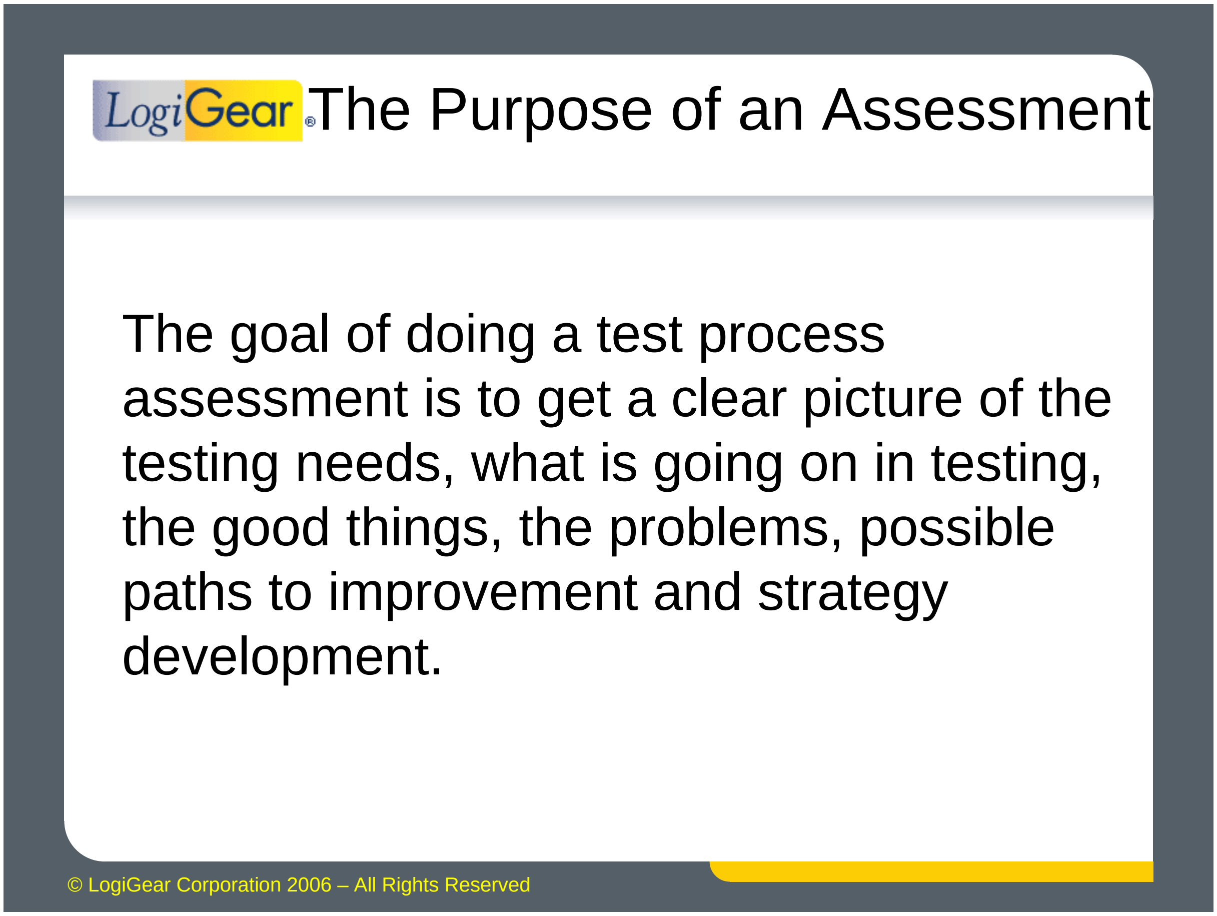


1. Assessing Needs

What is a test assessment?

A test assessment is a data gathering process.

To make decisions we need data.



The Purpose of an Assessment

The goal of doing a test process assessment is to get a clear picture of the testing needs, what is going on in testing, the good things, the problems, possible paths to improvement and strategy development.

Important Note

While the focus is on testing, this effort is much larger than only the test team.

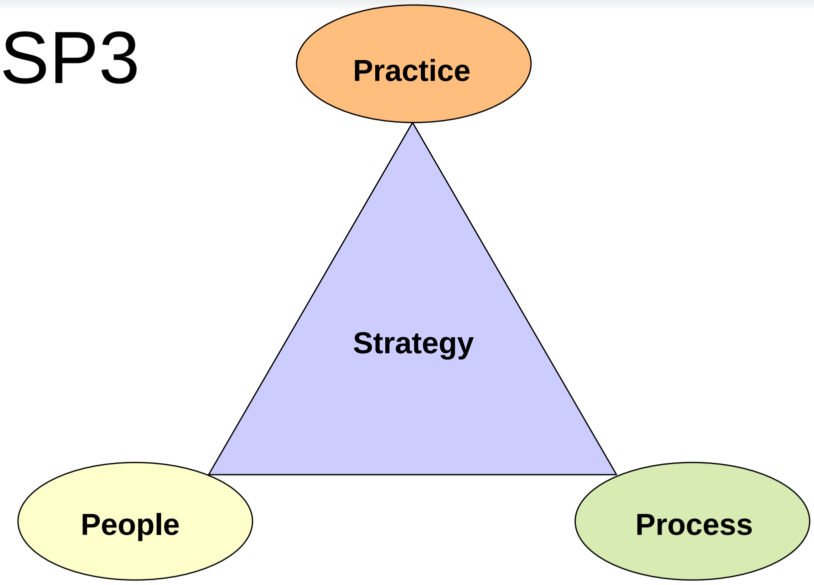
Issues will arise over:

- Who owns quality?
- What is the goal of testing?

If this assessment is done well you may step on some toes!

Testing Strategy in Context

SP3



Source: I developed this model by learning from Mitchell Levy, Value Framework® Institute

Examples of Possible Improvements

Possible Corrective Action	**Faster**	**Better**	**Cheaper**	**SP3**
Automate test execution	Yes	Maybe	Maybe	Practice
Focus manual testing on bug-finding rather than documentation	No	Yes	No	Process
Improve meaningful metrics with well-defined correlation/corrective action	Yes	Yes	Yes	Practice
Improve visibility of testing/QA activities	Yes	Yes	Yes	Practice
Reduce automated test script maintenance	Yes	Maybe	Yes	Practice
Test earlier	Maybe	Yes	Yes	Process
Upgrade talent through training and/or churning	Maybe	Yes	Maybe	People
Leverage outsourcing including global resources	Maybe	Maybe	Yes	Practice
Improve test design	Maybe	Yes	Maybe	People

A Common Theme of Findings

Your test teams might be doing things right, but not necessarily the right things!

Test Assessment Ingredients

- This assessment should produce a report that reveals the feedback on the maturity of the QA/Testing group as a whole, which includes evaluations of
  - QA/Testing human capital,
  - processes,
  - test strategy,
  - methods and tools,
  - project scheduling and
  - the overall effectiveness.

Test Assessment Ingredients

- The evaluation process should be based on the review and analysis of
  - The current quality related documentation
  - Surveys and interviews
  - Knowledge of best practices
- The content of the report should include:
  - Data collected through surveys and interview sessions.
  - Discoveries: Information that offers feedback on the current state-of-the practice contrasting with standard processes such TPI (Test Process Improvement), TMM (Test Maturity Model) including
    - A Process Scorecard: A metric-based report card for the overall test process assessment.
    - Staff Competency Scorecard: A metric-based report card for the overall human capital assessment.

Test Assessment Ingredients

- More on the content of the report
  - A Roadmap—A plan consisting of recommendations for improvement.
  - Implementation activities.
  - Implementation plan and execution.

Test Assessment Ingredients

- Assessment methodology
  - Phase I - Intake
    - Become familiar with the testing organization.
    - Identify the full scope of the project.
    - Determine which individuals within the organization should be involved.
    - Determine what quality criteria should be assessed and what questions to ask of each person involved.
    - Determine what documentation to be gathered from the organization for assessment.

Test Assessment Ingredients

- Assessment methodology
  - Phase II – Data Collection
    - Surveys given to personnel involved in the testing and development effort.
    - Face-to-face interviews with personnel involved in the testing and development effort.
    - Assessment of testing environments, office environments, and testing processes.
    - Review of existing documentation.

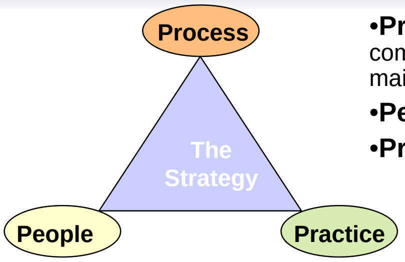
Test Assessment Ingredients

- Assessment methodology
  - Phase III – Report of findings and action plan
    - Actions and measures
    - Transition paths
    - Planning, timely, costs, and benefits

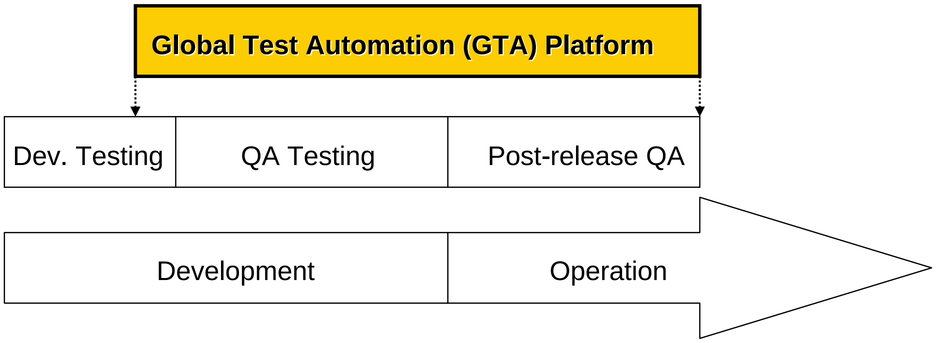
2. Test Process Alignment

Armed with data collected from the assessment, the next step is to look at the development and business process as a whole to help align the testing process.

The Process-Driven Test Strategy



- **•Process:** Treating testing and automation as a complete sub-life cycle with a focus on control, scalability, maintainability and reusability.
- **•People:** Focusing on role-based distributed teams.
- **•Practice:**
  - – Method: Focusing on process and people.
  - – Tool: The technology that supports the method.

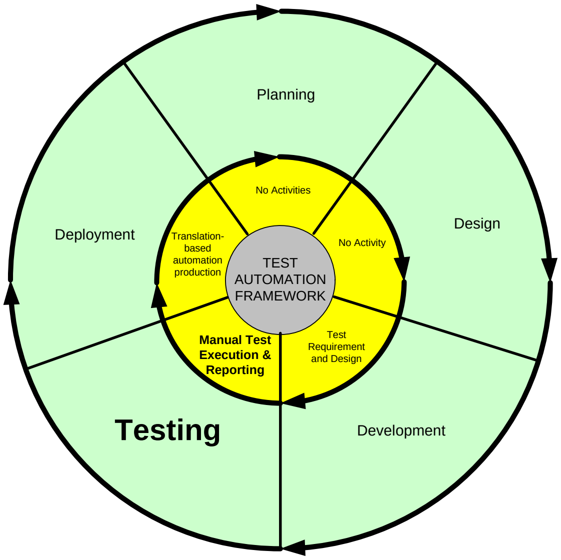


Process-Driven Test Strategy

- Process
  - Activities: Test Requirement | Test Design | Test Automation | Test Review | Test Execution—Manual and Automated | Test Report | Test Maintenance | Test Asset Handoff.
- People
  - Roles: Distributed teams of business analysts, test engineers, automation engineers, leads, managers, and sustain-engineering staff.
- Practice
  - Method: Keyword-driven
  - Tool: Keyword-driven enabling technology

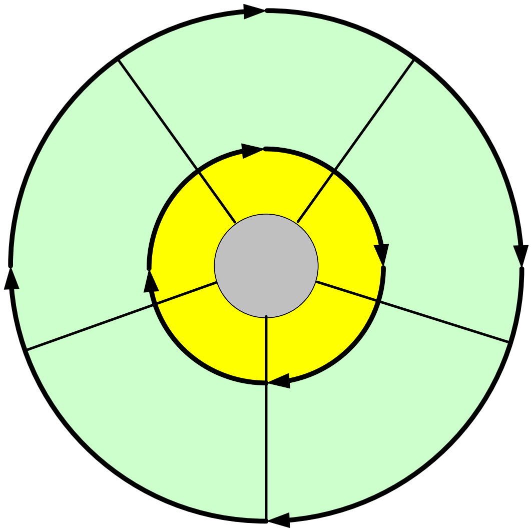
Test Framework & The Application Lifecycle

Integration of test and application development lifecycles without the Global Test Automation strategy.


Planning
Design
Development
Testing
Deployment
No Activities
No Activity
Test Requirement and Design
Manual Test Execution & Reporting
Translation-based automation production
TEST AUTOMATION FRAMEWORK

Test Framework & The Application Lifecycle

Integration of test and application development lifecycles with the Global Test Automation strategy.



---

3. Leveraging Automation

- Key Success Factors for Automation
  - Accessible and maintainable test structure
  - High degree of automation
  - High re-use
  - At least as maintainable as the system under test
  - Test specification separate from automation
    - Making sure to avoid double work between tester and engineer

The 5% Challenge for Test Automation

- No more than 5% of all effort around testing should involve automating the tests.
- No more than 5% of all tests should be executed manually.

Manual Testing Roles and Tasks

- *Roles*
  - Test engineers: Design/write and execute test cases; report bugs
- *Tasks*
  - *Test Case Production*
    - Design/Compose
    - Write/Transcribe
  - *Test Execution*
    - Inputs (Run the test cases)
      - Setup environment
      - Generating inputs
    - Outputs
      - Problem identification or examine outputs
  - Analysis
    - Problem analysis
    - Bug identification
    - Bug isolation
  - *Bug reporting*

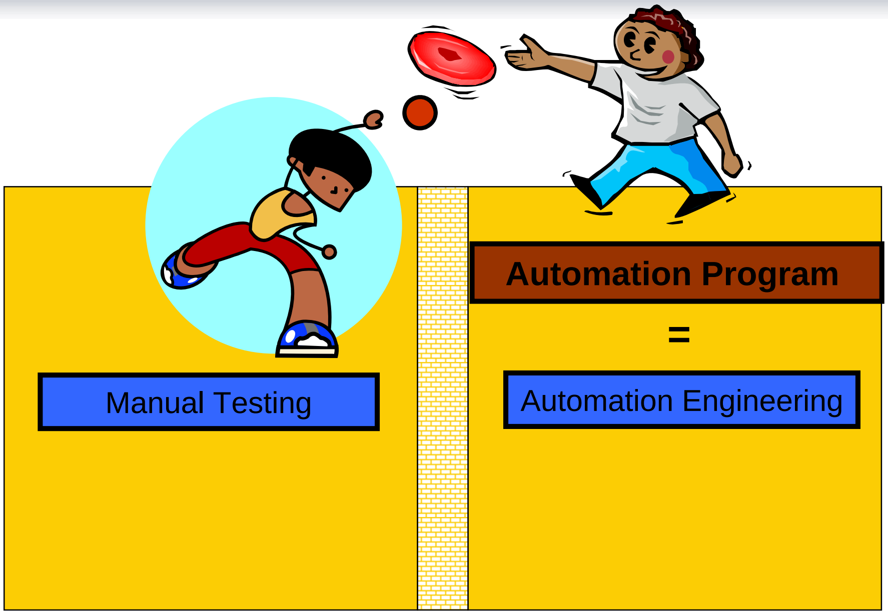
Automation Engineering Roles and Tasks

- Roles
  - Automated Test Engineers (ATE)—Creating (usually with a tool-specific programming language) automated tests or keywords (when keyword-driven framework is employed)
  - Automation Framework Engineers (AFE)—Maintain the framework and support ATE's in the application of the framework.
  - Automation System Support Engineering (low-level)
    - Support the integration of the framework with its native test automation agents or third-party test automation agents
    - Provide support of custom interfaces, objects, and/or platforms
- Tasks—Automate Test Case Execution
  - Coding approach without framework—Code or program the written tests to automate the test execution.
  - Coding approach with framework (hybrid)
    - *Use a framework (such as keyword-driven)*
    - *Code or program the keyword*
    - *Code the written tests but leverage the pre-created keyword to boost automated test productivity and minimize automated test maintenance*
  - Non-coding approach with keyword-driven framework—*Code or program and maintain libraries of keywords*

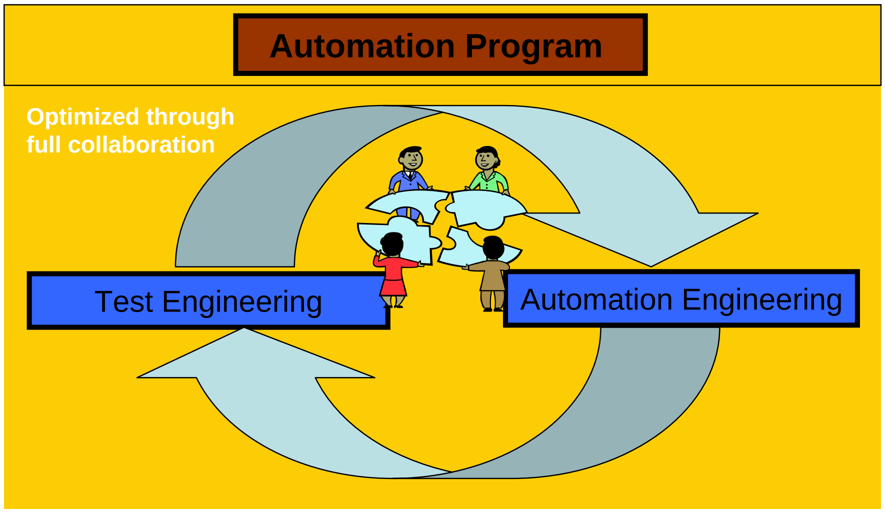
A Successful Automation Program

- *Objective*
  - *Achieve one or more of the following benefits: Reduce test cycle time, improve test coverage, and reduce overall testing costs*
- *Characteristics of a successful program*
  - *Fully utilize and leverage the talent and resource pool*
  - *Maximize team adoption and collaboration*
  - *Optimize all four areas of manual testing including test case production, execution, analysis and reporting*
  - *Optimize control over the testing and test automation process*
  - *Minimize scripting or programming*
  - *Maximize reusability*
  - *Minimize script maintenance*
  - *Fully capitalize testware as an asset*
  - *Fully scalable*

Example of a Partially Completed Program



A Complete and Scalable Program



Surveys & Case Studies

Case Study: Centrify Corporation

- US and offshore teams.
- Windows/Unix/Linux/Solaris 1.0 release.
- Custom automation platform built in 1 month.
- Global team designing tests within 3 months.
- 25 million tests run in less than 6 months.
- 80% of tests automated.
- Full test suite runs in 8 hours.
- Currently supports 35 operating platforms, automated tests run against every platform

---

4. Minimize Costs and Risks

- Build trust or use an offshore vendor you trust!
- Train the offshore staff or work with vendor who provides competent staff.
- Train all US leads who manage offshore resources as well as offshore leads.
- Address all of the communication issues including cultural and time zone differences, technical, English and project communication skills.
- Get a management tool or enabling framework that helps you facilitate project and task management.
- Get someone local who can facilitate.
- Get someone else to manage and be part of my team.
- Increase the productivity or savings from offshore team & reduce the headaches of my team in US.

5. Select the Right Tool

- You want to create the requirements for the tool based on your strategy of how your organization will optimally leverage
  - test automation,
  - the talent pool of test engineers and automation engineers,
  - distributed teams such as offshore teams,
  - and the process-driven strategy.

Tool Options

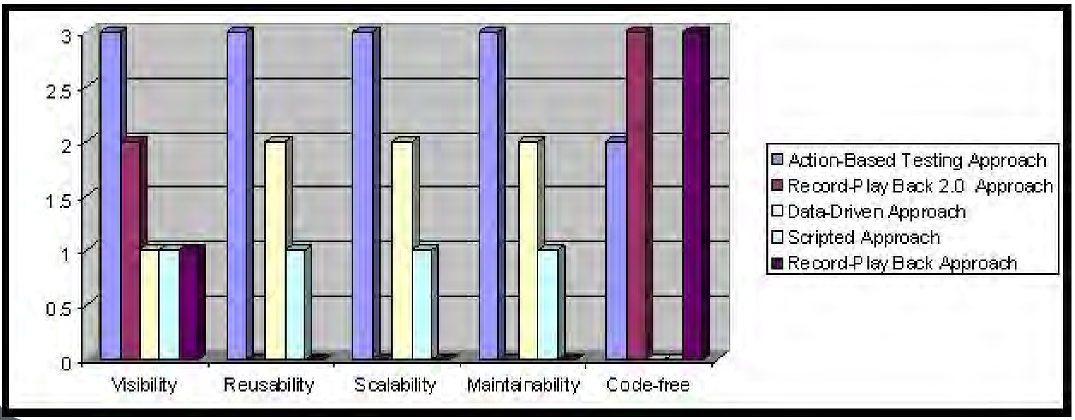
There are several different types of test automation tools available. They are identified below.

1. Action-Based Testing Approach – tools that utilize the hierarchical ABT techniques described in this chapter that focus on embedding the methodology for team-based test automation with great care on maintainability, scalability, reusability and visibility.
2. Record-Playback 2.0 Approach – tools that partially apply the keyword or similar approaches which focus on making it easy for non-technical engineers rather than maintainability, scalability and reusability.
3. Data-Driven Approach – tools that make use of captured data to execute the tests.
4. Scripted Approach – tools that use scripts to program and execute the tests.
5. Record-Playback Approach – tools that use a GUI to record mouse movements and keystrokes, and play them back automatically to run the tests.

Tool Options

	Visibility	Reusability	Scalability	Maintainability	Code-free
Action-Based Testing Approach	3	3	3	3	2
Record-Play Back 2.0 Approach	2	0	0	0	3
Data-Driven Approach	1	2	2	2	0
Scripted Approach	1	1	1	1	0
Record-Play Back Approach	1	0	0	0	3

Legend
0: Very Poor
1: Marginal
2: Good
3: Very Good



Questions to Ask

- Does the tool support role-based test design, automation, execution, and management?
- Does the tool support traceability throughout the testing and test automation life cycle?
- Does the tool provide tracking and reporting mechanism for globally distributed testing teams' activities?
- Does the tool support seamless test asset transferring?
- Does the tool provide complete test automation development and QA testing life-cycle management capability?

Questions to Ask

- Does the tool support a complete integration testing and test automation while leverage the talent pool to focus on separate activities such as test design versus automation engineering?
- Does the tool leverage an embedded test and automation methodology?
- Does the tool help maximizing test and automation productivity?
- Does the tool help minimizing the maintenance effort?
- Does the tool provide accessible, visible and maintainable test structure?

Questions to Ask

- Does the tool support scalable production or a high degree of automation (the 5% rule)?
- Does the tool support a high degree of reusability?
- Does the tool provide comprehensive built-in playback support for many common application platforms such as Web, Window, .NET, UNIX, Linux, etc.?
- How flexible is the tool to be extended for uncommon platforms and/or interfaces such as custom software and embedded software (non-UI)?

About ABT

The ABT methodology improves on traditional test automation techniques by enabling non-automation experts to create automated tests, decreasing the amount of automation scripting required, and significantly reducing the amount of work necessary to update tests after a revision of the application under test.

To understand Action-Based Testing, it is beneficial to look at it in two ways: First, as compared to software development, and second, as compared to traditional test automation. By using analogy as well as comparison and contrast, it will become clear how ABT can produce superior results.

About ABT

In software development, it is now accepted practice to use object-oriented software development methodologies. The central theme of object-oriented software development methodologies is packaging code and related data structures together into “objects” with associated “methods” that perform functions utilizing the data structures associated with those objects. There are numerous benefits that come from this approach. For one, if there are changes in the data structures, only the code associated with the object needs to change. The object has an interface to all other code which may use it that can remain the same. Furthermore, an object can be used by any number of other objects, creating a hierarchical software structure. Low level functions can be developed once and used an infinite number of times by other higher level functions. Then, if there is a change to a low level function, all the higher level functions that utilize it inherit the change without any additional re-coding on their part. This is a huge benefit for the development of large software systems, and is one of the key reasons that software development has continued to get faster and more advanced. Before object-oriented programming practices were commonly used, seemingly small changes required in software may have required a great deal of effort because the impact of the changes could be widespread. It was a maintenance headache.

About ABT

ABT is to testing what object-oriented programming is to software development. Rather than create a single test script for each test to be run, there are test scripts created for low-level actions that are then utilized by higher level test modules. The test engineer will develop tests that utilize actions based on keywords such as “login,” and then the automation engineers will develop test scripts that implement “login” with low-level steps such as “click,” “find text box A in window B,” “enter username,” etc. A relatively small number of low level test scripts are required compared to the total number of tests that are developed. If a change must be made to a low-level script, say because a button’s behavior changed, only that low-level script needs to be changed. Every test module that utilized it will then inherit the change. ABT brings the benefits of object-oriented technology to testing.

Keyword-based test design can actually begin based on documents developed by business analysts or the marketing department, before the final details of the software to be tested are known. As the test automation process proceeds, bottlenecks are removed and the expensive time of highly-trained professionals is used more effectively.

About ABT

One way that the ABT methodology introduces cost savings over traditional test automation is by decreasing the number of scripts that must be written by automation engineers. In the traditional test automation approach, test engineers document tests, and then automation engineers create scripts that implement those tests. In the ABT method, test engineers create executable “test modules” based on a library of “actions”, and automation engineers create a relatively small number of automation scripts to implement the “actions” used in the test modules.

Since the test modules created by the test engineers are executable, the ABT methodology can result in lower costs for each test because they don’t need to be converted to executable modules by automation engineers in a second step. And since the executable test modules are written in a clearly readable language, it is not necessary for test engineers to create additional documentation, such as verbose MS Word documents. This can also result in lower costs for each test.

About ABT

In the ABT methodology, automation engineers create "low-level" actions that handle discrete controls in the application, such as buttons, text fields, dropdown lists, radio buttons, etc. The number of low-level actions needed for testing is directly related to the complexity of the platform the application is using. For example, if the application is a Windows GUI application, it will need more low-level actions than if the application is a command line application. In practice, you might expect that a typical GUI-based application will need 30-50 low-level actions.

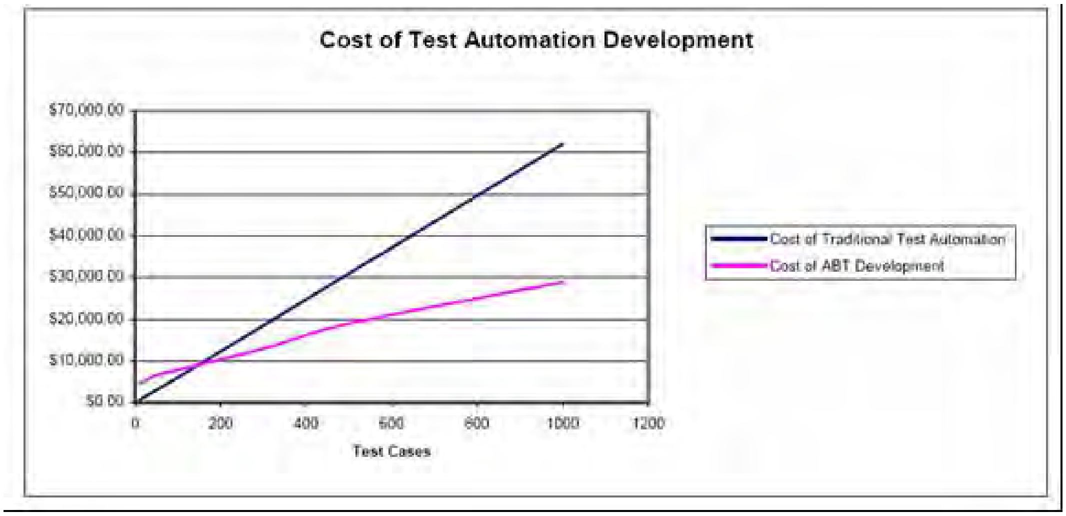
About ABT

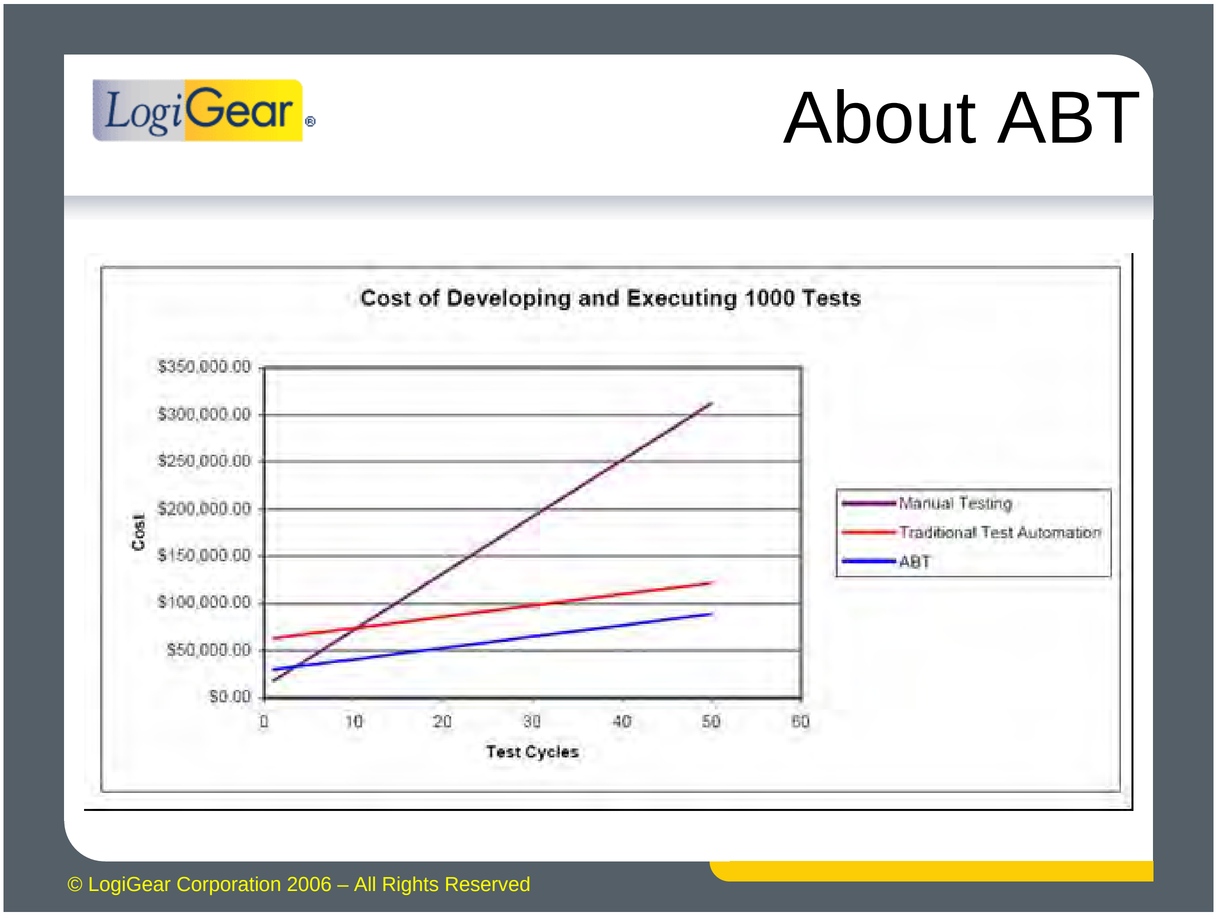
Once the low-level actions have been created, automation engineers can use these to create "intermediate-level" and "high-level" actions. Intermediate level actions typically deal with a single window or dialog in an application; high-level actions typically will move across several windows in an application, and represent business functions of the application. In general, there is a direct relationship between the number of windows/dialogs in an application and the number of intermediate-level actions. Similarly, there is a relationship between the number of use cases and/or functional requirements and the number of high-level actions.

About ABT

When comparing the ABT methodology to a traditional test automation approach, we see a cost savings as the number of test cases (or in the case of ABT, “test scenarios”) increases. The cost of test execution for ABT and traditional automation is roughly the same. A test engineer will typically start the test, perform some other tasks, then come back and analyze the results.

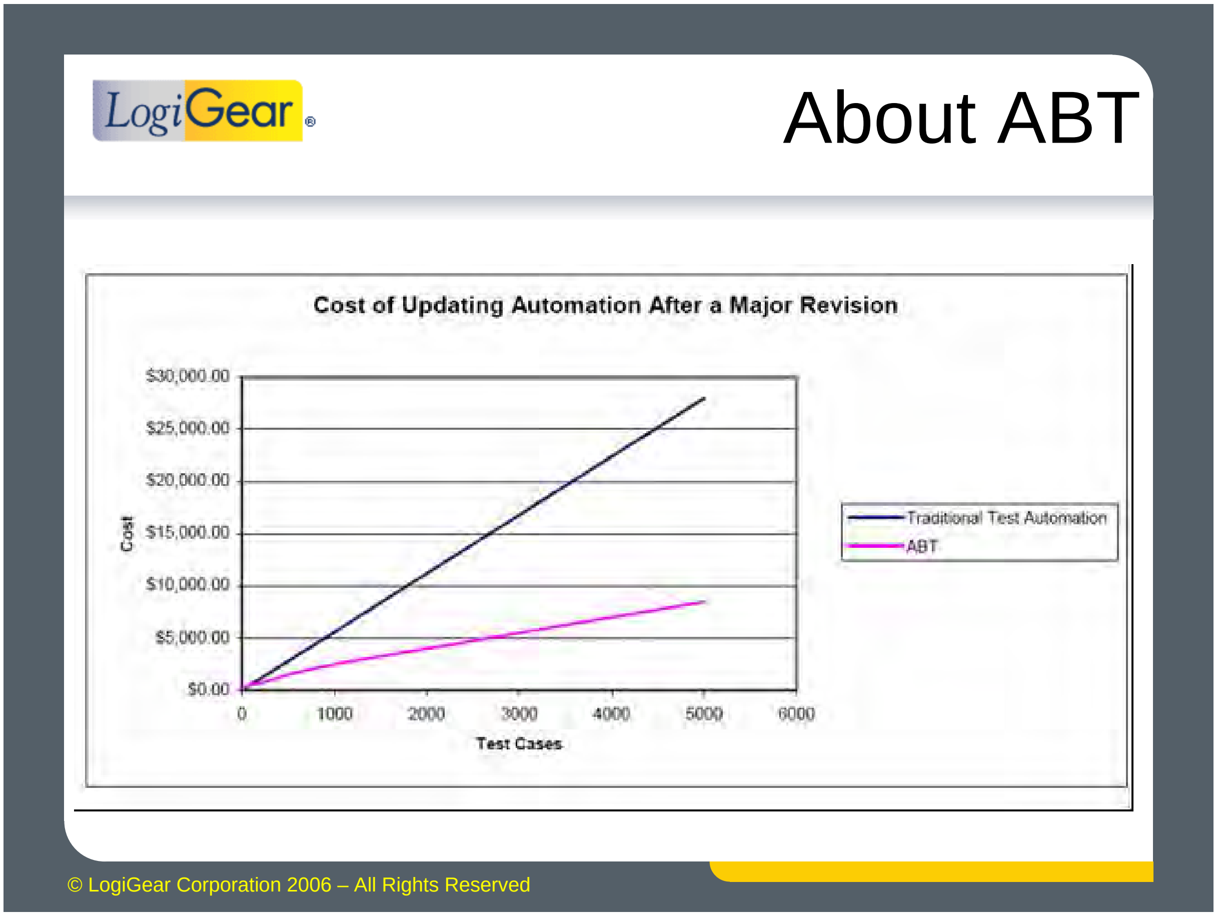
About ABT





About ABT

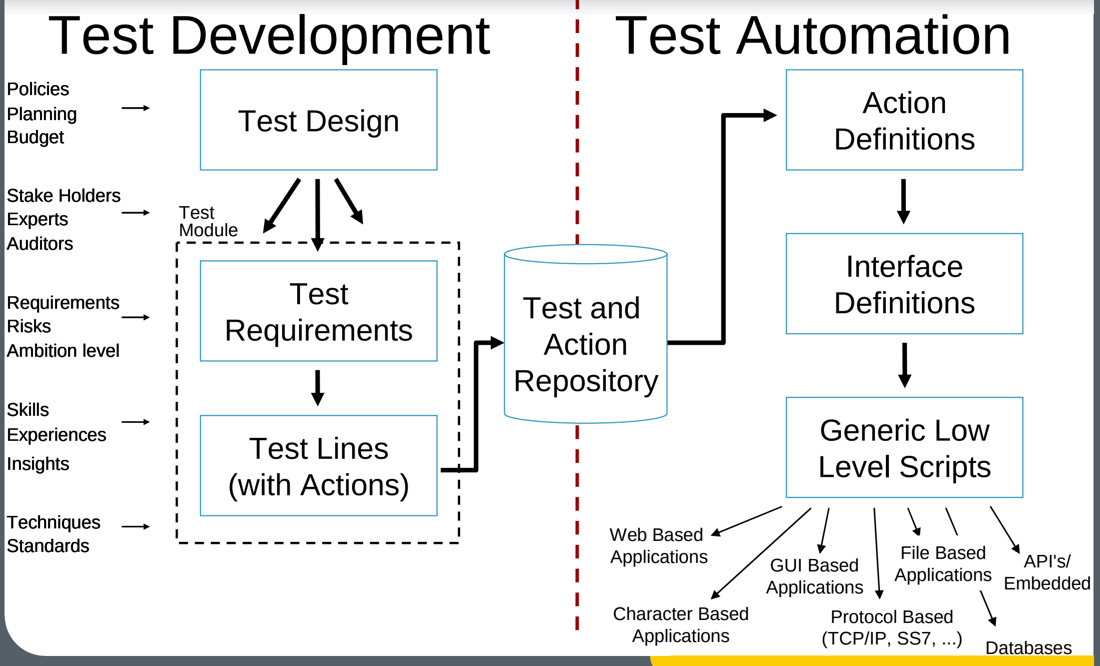
Cost of Developing and Executing 1000 Tests



About ABT

Cost of Updating Automation After a Major Revision

The Action Based Testing Model



Test Development

Policies
Planning
Budget

Stake Holders
Experts
Auditors

Requirements
Risks
Ambition level

Skills
Experiences
Insights

Techniques
Standards

Test Design

Test Module

Test Requirements

Test Lines (with Actions)

Test and Action Repository

Test Automation

Action Definitions

Interface Definitions

Generic Low Level Scripts

Web Based Applications

GUI Based Applications

File Based Applications

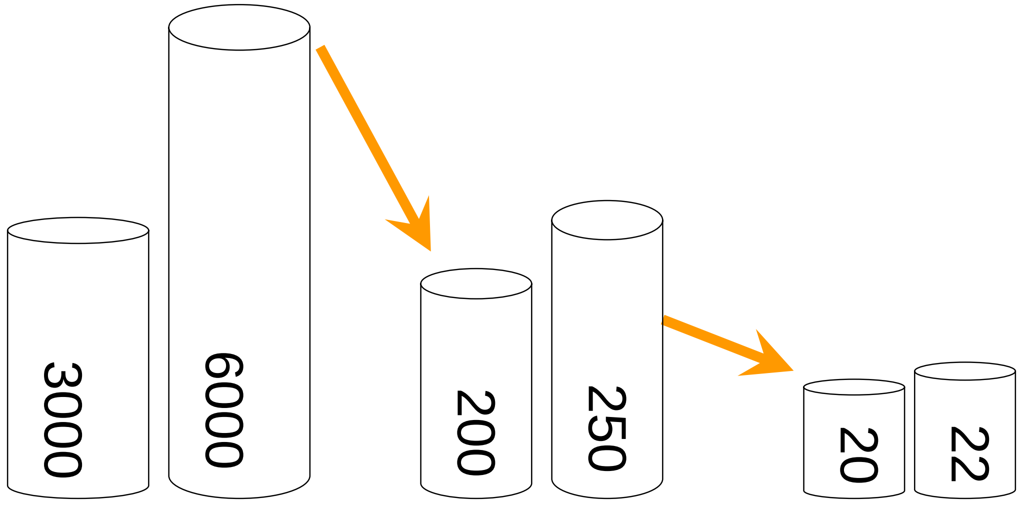
API's/ Embedded

Character Based Applications

Protocol Based (TCP/IP, SS7, ...)

Databases

Scalability Illustration



Test Cases

ABT Actions

Low-Level Actions

Surveys & Case Studies

- Case Study: A Fortune 1000 Company
- US and Offshore teams.
- Windows/Unix/Solaris cross-platform
- Reduced test creation time by 67%
- Test execution time is now only 7 hours
- (compare to 36 hours for the previous scripts)
- Testing coverage is doubled
- Expanded test suite runs 500% faster
- Overall testing and test automation costs reduced by 31%

---

6. Secure/Develop Competency

- Your global staff must be equipped with adequate competency to do the assigned work.
- Your US staff must be equipped with adequate competency to collaborate and manage the offshore team.
- The global teams must seamlessly integrate.

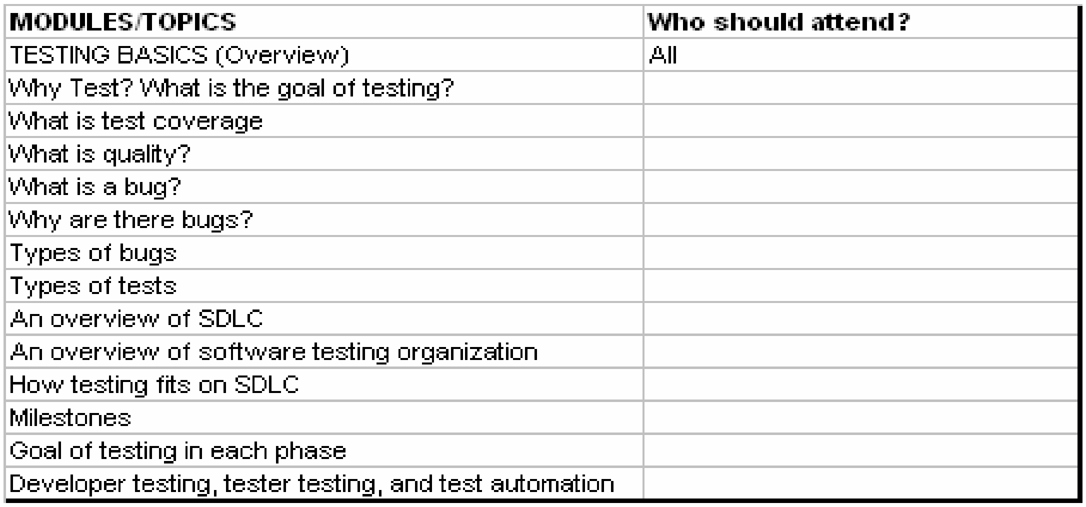
LogiGear®

6. Secure/Develop Competency

- Have a comprehensive training program in place for your teams.
- If you use an offshore vendor, make sure that their staff is fully trained, and a comprehensive training program is in place.

LogiGear®

A Basic Curriculum Example



MODULES/TOPICS	**Who should attend?**
TESTING BASICS (Overview)	All
Why Test? What is the goal of testing?	
What is test coverage	
What is quality?	
What is a bug?	
Why are there bugs?	
Types of bugs	
Types of tests	
An overview of SDLC	
An overview of software testing organization	
How testing fits on SDLC	
Milestones	
Goal of testing in each phase	
Developer testing, tester testing, and test automation	

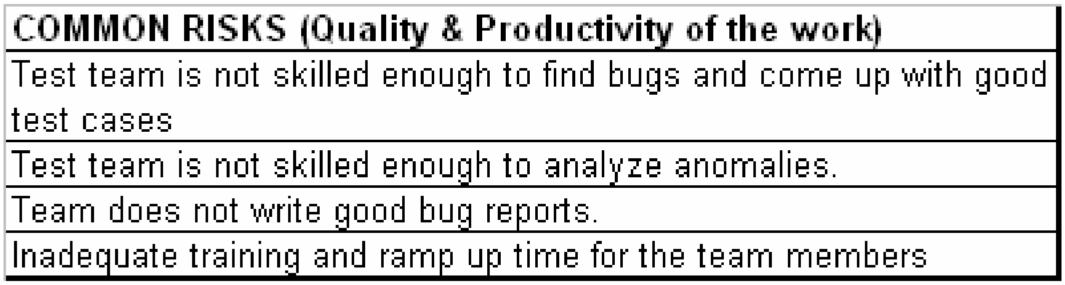
A Basic Curriculum Example

MODULES/TOPICS	Who should attend?
SOFTWARE TESTING SKILLS (Hard skills)	All
Test execution (running pre-written test cases)	
Bug identification	
Bug finding	
Bug analyzing and reproducing	
Bug reporting	
Bug reporting with TRACKGEAR	
Test case designing and writing	
Test case design with TestArchitect	
Status reporting	
Other reporting and communication	
Test documentation	

A Basic Curriculum Example

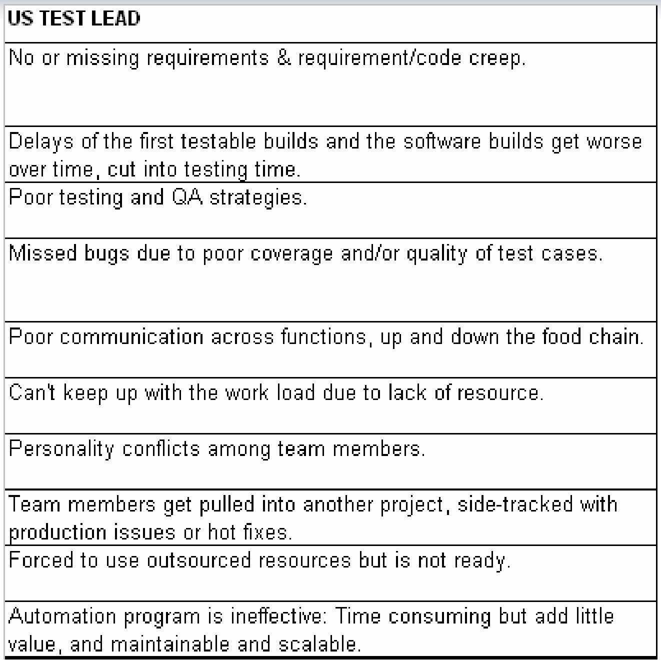
MODULES/TOPICS	Who should attend?
OUTSOURCING (Soft skills)	All
Understand the deliverables	
Understanding customer expectations	
The psychological issues of outsourcing	
Outsourcing models	
Roles and responsibilities	
Exceeding customer's expectation	
Delivering customer satisfaction	
Working in a distributed environment	
The LogiGear/LTRC difference	

Test Lead Training—Common Risks



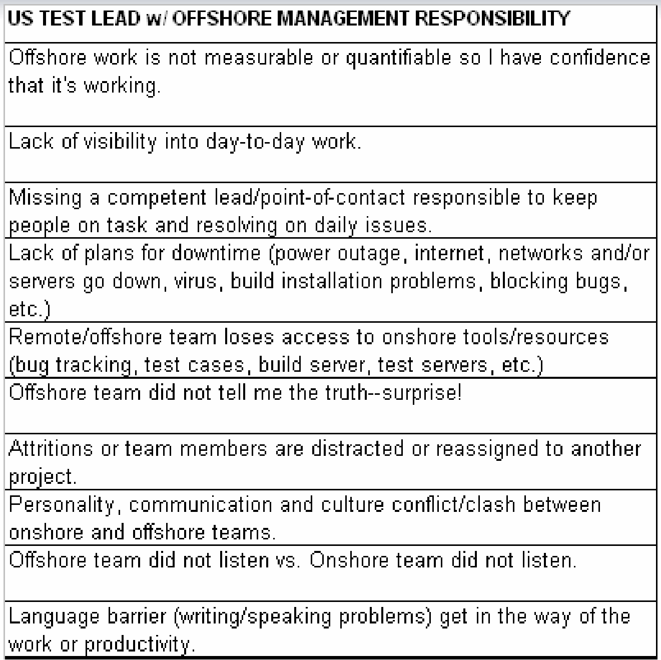
COMMON RISKS (Quality & Productivity of the work)
Test team is not skilled enough to find bugs and come up with good test cases
Test team is not skilled enough to analyze anomalies.
Team does not write good bug reports.
Inadequate training and ramp up time for the team members

Test Lead Training—Top 10 Risk



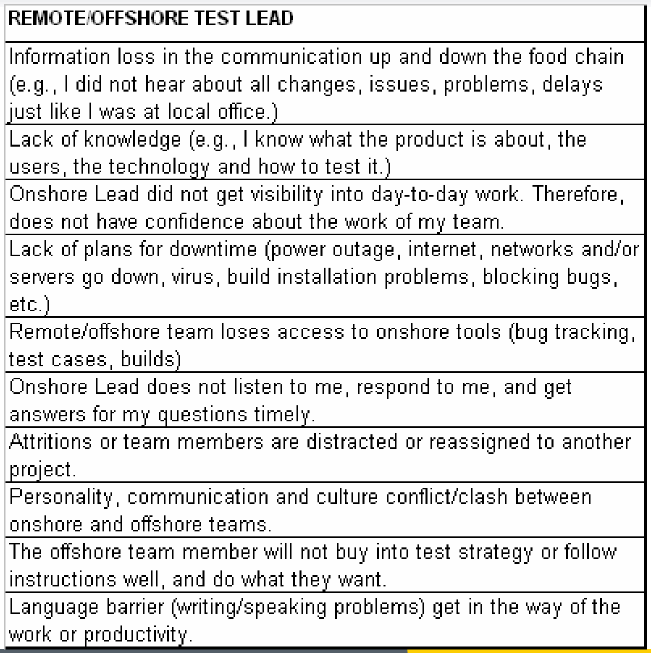
US TEST LEAD
No or missing requirements & requirement/code creep.
Delays of the first testable builds and the software builds get worse over time, cut into testing time.
Poor testing and QA strategies.
Missed bugs due to poor coverage and/or quality of test cases.
Poor communication across functions, up and down the food chain.
Can't keep up with the work load due to lack of resource.
Personality conflicts among team members.
Team members get pulled into another project, side-tracked with production issues or hot fixes.
Forced to use outsourced resources but is not ready.
Automation program is ineffective: Time consuming but add little value, and maintainable and scalable.

Top 10 Risk to Manage



US TEST LEAD w/ OFFSHORE MANAGEMENT RESPONSIBILITY
Offshore work is not measurable or quantifiable so I have confidence that it's working.
Lack of visibility into day-to-day work.
Missing a competent lead/point-of-contact responsible to keep people on task and resolving on daily issues.
Lack of plans for downtime (power outage, internet, networks and/or servers go down, virus, build installation problems, blocking bugs, etc.)
Remote/offshore team loses access to onshore tools/resources (bug tracking, test cases, build server, test servers, etc.)
Offshore team did not tell me the truth--surprise!
Attritions or team members are distracted or reassigned to another project.
Personality, communication and culture conflict/clash between onshore and offshore teams.
Offshore team did not listen vs. Onshore team did not listen.
Language barrier (writing/speaking problems) get in the way of the work or productivity.

Test Lead Training—Top 10 Risk



REMOTE/OFFSHORE TEST LEAD
Information loss in the communication up and down the food chain (e.g., I did not hear about all changes, issues, problems, delays just like I was at local office.)
Lack of knowledge (e.g., I know what the product is about, the users, the technology and how to test it.)
Onshore Lead did not get visibility into day-to-day work. Therefore, does not have confidence about the work of my team.
Lack of plans for downtime (power outage, internet, networks and/or servers go down, virus, build installation problems, blocking bugs, etc.)
Remote/offshore team loses access to onshore tools (bug tracking, test cases, builds)
Onshore Lead does not listen to me, respond to me, and get answers for my questions timely.
Attritions or team members are distracted or reassigned to another project.
Personality, communication and culture conflict/clash between onshore and offshore teams.
The offshore team member will not buy into test strategy or follow instructions well, and do what they want.
Language barrier (writing/speaking problems) get in the way of the work or productivity.

7. Measure, Analyze & Optimize

- To improve you first need to have data
- To collect data you need measurement
- To know which measurement to take, you need to know
  - what you want to improve
  - the relationship between the measurement and the improvement indicator
  - what affects your measurement
- To make improvement decisions, you need to do root-cause analysis on your collected metrics

---

7. Measure, Analyze & Optimize

- If you are not the ‘key’ decision maker, you need to know who is the stakeholder. Who would care?
- You need to establish what number would be used as a success indicator because you will need to compare your data against referenced data.
- Metrics must be actionable
- Often time, the place to start is collecting metrics to find out where you are, so you can establish tangible goals (after you qualify the metric validity).

7. Measure, Analyze & Optimize

- What do you want to optimize?
  - QA/Test resource capability
    - Testing throughput or productivity?
    - Testing service quality?
  - The effect of the team capability on the released software (Strategy)?
    - The breadth of the test coverage?
    - The depth of the test coverage?

---

7. Measure, Analyze & Optimize

- What do you want to optimize?
  - Production throughputs?
    - Requirements?
    - Architectural and/or application design?
    - Specification?
    - Code?
  - The production quality (deliverables)?
    - Requirements?
    - Specification?
    - Code?
  - The effect of production throughputs and quality (Strategy)?
    - Deliverable quality?
    - Timeliness?
    - Cost efficient?

7. Measure, Analyze & Optimize

- What do you want to optimize?
  - Project management visibility and/or predictability?
  - Customer satisfaction?
    - Functionality?
    - Reliability?
    - Usability?
    - Performance?
    - Compatibility?
    - Security?

What Metric to Use and Why?

- What value do we want to associate with being “successful?”

  e.g., miss a fewer number of bugs, or execute a larger number of tests
- Why *this* metric is relevant to the “success” value?
- What are we comparing against?

  E.g., Wrote and executed 2,064 in 26 cycles. Is it a lot or two few comparing to what?
- What is value that we strive to meet?
- What about data integrity and dependency?

Some Examples

Project Management—getting the product out. These are typically measures of software stability and activity

- How many test cases done
- Defect counts
- Hours tested against a build
- Code churn
- Requirements stability

Process Improvement

- Defect aging
- Requirement stability
- Valid defects found vs. test method
- Bugs by severity post release

Lessons Learned

- Quality engineering and testing strategy starts from the top.
- Your management must budget and adequately fund testing and QA as a separate line item.
- You need to create visibility into the testing and test automation process.
- Metrics for visibility are not rocket science, but they need to be established quantitatively and qualitatively.
- Recognize that manual testing is unavoidable, but that you need to encourage automation when and wherever possible while applying the Automation 5% Rules.
- Although automation solves the speed problem, it's not a silver bullet.
- Global software test automation can save money and time as well as provide around the clock productivity.
- Plan first, execute second — the Global Test Automation strategy should come first. The strategy then integrates automated testing programs and global testing resources.
- Be critical on staffing — don't settle for second-class quality staff, and don't treat them as second class.

In Summary

- You want to view your QA/testing and test automation activities as a complete sub-lifecycle.
- You want to develop a strategy that optimize your overall *team* throughputs in terms of quality of service, speed and cost.
- Use the Seven-step process to develop your strategy: (1) Assess, (2) Align your test process, (3) Leverage automation, (4) Minimize costs and risks of global resources, (5) Select the right tools, (6) Secure/Develop competency, and (7) Measure, set goals and optimize.
- You want to turn you strategy into a roadmap for your global test automation strategy. Changes will take time. So long that you don't get lost, you will reach your destination.

About LogiGear Corporation

LogiGear provides global solutions for software testing, focusing on test automation. For over a decade, we've worked with hundreds of companies, from Fortune 500 to startups, delivering unique testing solutions that meet their unique needs. We double their test coverage, cut test time in half, improve quality and reduce cost.

LogiGear has built a reputation for offering the widest range of services in the software testing industry. Be it turn-key test automation, consulting, training, outsourced testing, or products, we partner with software organizations to create approaches that precisely meet their demands.

Enjoy CAST 2006!

www.LOGIGEAR.com