

DEDICATED TO SHOWCASING NEW TECHNOLOGY AND WORLD LEADERS IN SOFTWARE TESTING

LogiGear MAGAZINE

THE OFFSHORING & OUTSOURCING ISSUE

Spotlight Interview :
Guido Schoonheim,
Distributed Agile

FEATURE ARTICLE

*Important Lessons Learned
in 10 Years of Offshoring*
By Michael Hackett

RELATED ARTICLE

Building Babel
By Christopher Williams,
Ph.D., MBCS

BOOK EXCERPT

*Global Software Test
Automation: A Discussion of
Software Testing for Executives*

VISTA CON™ 2011

viet Nam International Software Testing & Automation Conference



**December 6 & 7
Ho Chi Minh City, Viet Nam**

ADVANCING THE PRACTICE OF SOFTWARE TESTING AND TEST AUTOMATION

Topics:

Test Automation

Test Strategy, Approach, Methods and Techniques

Testing in Agile Development

Cross-cultural Business and Outsourcing

Participating Companies:

Microsoft

Oracle

EA Games

And more!



**Early Bird
Register before
October 15**

Up to 30% OFF

Contact us

+84.8.39954072

Option 3

www.vistacon.vn

EDITOR'S LETTER

EDITORIAL STAFF

Editor in Chief

Michael Hackett

Managing Editor

Lolita Guevarra

Contributing Editor

Thi Doan

Webmaster

Dang Truong

Worldwide Offices

United States Headquarters
2015 Pioneer Ct., Suite B
San Mateo, CA 94403
Tel +01 650 572 1400
Fax +01 650 572 2822

Viet Nam Headquarters
1A Phan Xich Long, Ward 2
Phu Nhuan District
Ho Chi Minh City
Tel +84 8 3995 4072
Fax +84 8 3995 4076

Viet Nam, Da Nang
7th Floor, Dana Book building
76-78 Bach Dang
Hai Chau District
Tel +84 511 3655 33
Fax +84 511 3655 336

www.logigear.com
www.logigearmagazine.com

Copyright 2011
LogiGear
All rights reserved.
Reproduction without permission is prohibited.

Submission guidelines are located at
<http://www.logigear.com/logigear-magazine/submission-guidelines.html>



I worked for a company in the late 80s that outsourced and offshored some software development to Bulgaria. Another company I worked for in the early 90s contracted its documentation development overseas to the Bahamas.

There is nothing new concerning the idea of offshoring or outsourcing. What has changed are a few key aspects: the company type distributing their work, the amount of work outsourced and offshored, and countries of choice. Outsourcing and offshoring are no longer only for very large corporations but also for small and startup companies. This standard has shifted significantly in the last decade. The pool of offshored services has also expanded from India, China, and Russia to now include countries such as El Salvador, Romania, and Viet Nam. Lastly, companies that once asked themselves, "Should we?" now focus on "How do we do it better?".

For the last dozen years I have done consulting and training. When I began this work, it was centered around Silicon Valley. Since 2000, I have conducted the same work but in sixteen countries. Today, there are *Silicon Valleys* in dozens of countries. In a world where global software engineering is the norm, often lost in the discussions regarding distributed software development are the human issues: communication, cooperation, respect, honesty, and trust. Each idea is a two way street. Trust, for example, has to be gained by all sides in a work distribution. It is not granted by one participant. Failure at any one of these will undermine or destroy a project. Too often soft skill training—whether it is Doing Business with Americans or Cross-Cultural Communication—is not given the time and attention it needs. Software development is about people. You can't get satisfied customers from an unhappy staff. People are the key asset of any knowledge-based company. This is especially true about outsourcing and offshoring: it's about people and building communication and trust—not the SDLC process or management tool.

In this issue we cover a variety of issues concerning outsource and offshore services. The feature story focuses on 10 lessons learned from 10 years of offshoring; an article by Christopher Williams discusses his research conducted in India on the importance of explicit and tacit knowledge between offshore vendors and clients; Spotlight Interview presents a one-on-one with Guido Schoonheim fielding questions from offshore testers; and Blogger of the Month Vipul Gupta explains how outsource testing vendors have developed their strategies to meet their clients' needs. We've also included information on LogiGear's new training course "Working with Offshore Test Teams"; 2010 Global Survey on offshoring; and a book excerpt from *Global Software Test Automation*.

We've provided a jam-packed magazine addressing many areas of interest for those of you who have been outsourcing and offshoring for years, or are about to outsource your first project.

Michael Hackett
Senior Vice President
Editor in Chief

IN THIS ISSUE



Guido Schoonheim
Spotlight Interview
Page 14

Departments

5 IN BRIEF

Working with Offshore Test Teams
New LogiGear Training Course



International Conference on Global Software Engineering, ICGSE 2011

2011 Conference Schedules

7 BLOGGER OF THE MONTH

Vipul Gupta
Software Testing to Software Engineering

24 2010 GLOBAL TESTING SURVEYS

Offshore Survey

In two parts, Michael Hackett analyzes the results of the home team and the distributed team in the sixth installment of the series.

28 VIET NAM SCOPE

Viet Nam's Next Generation

A young country progressing quickly makes way for its new leaders. Hong Mai Nguyen leads a movement of women in the technology industry.

8 FEATURE ARTICLE

Important Lessons Learned in 10 Years of Offshoring

Michael Hackett, Senior VP LogiGear

From advice on training to building a communication infrastructure, Michael lists the fine points of what makes a successful relationship with your outsource/offshore partner.

12 RELATED ARTICLE

Building Babel

Christopher Williams, Ph.D., MBCS

Christopher examines how explicit and tacit knowledge strengthens the relationship between client and vendor.

14 SPOTLIGHT INTERVIEW

Guido Schoonheim

As a pledged Scrum fanatic, Guido answers field questions from testers on how to address offshoring challenges dealing with Agile practices.

18 BOOK EXCERPT

Global Software Test Automation: A Discussion of Software Testing for Executives

Hung Q. Nguyen, Michael Hackett, and Brent K. Whitlock.

Addressing fundamental issues in software testing, *Global Software Test Automation* focuses on the high-level elements necessary to better execute software automation and outsourcing initiatives.

Working with Offshore Test Teams

New LogiGear Training Course

This two-day course focuses on developing a strategic approach to distributed test project management, effective communication, bug-database management and metrics, resource evaluation, and successful test execution, as well as many of the "soft" skills needed to lead and manage offshore teams. Implementation and use of test management tools and documentation will also be covered. Application of these concepts will be demonstrated in class examples, discussions and exercises. In this class, you learn how to best prepare yourself for leading and managing offshore teams to maximize test productivity while minimizing quality risks and stress.

For more information, visit <http://www.logigear.com/services/qa-training/working-with-offshore-test-teams.html>

LogiGear Channel



"What is the most important issue to resolve in Global Software Engineering?"

From the 6th IEEE International Conference on Global Software Engineering, participants answered the question sharing their thoughts. To view video, please click [here](#) or visit <http://www.youtube.com/watch?v=k3NmUIQdV-g> or <http://www.logigear.com/resources/videos.html>



International Conference on Global Software Engineering, ICGSE 2011



A conference on Global Software Engineering: University researchers and Industry experts get together to compare findings and work toward solutions.

Michael Hackett

The 6th IEEE International Conference on Global Software Engineering of 2011 (<http://icgse.org/>), is organized by the world's largest association for the advancement of technology. The conference brought together researchers and practitioners interested in exploring how globally distributed teams work and how challenges can be met.

It also provides opportunities for researchers from across the globe to exchange and discuss scientific and engineering ideas at various stages of development. Conference attendees got an opportunity to explore current best practices as well as industry trends that will shape the future.

ICGSE 2011 took place August 15- 18, 2011 in Helsinki, Finland. The theme this year was Global Software Engineering Challenges for the Next Decade. There were approximately 100 attendees from 24 countries. The topics this year were: GSE Governance, Global Agile and Lean, Knowledge and People Management, Testing and Quality, and Communication and Education.

There were three keynote addresses, including Walt Scacchi from the Institute for Software Research at University of California Irvine, speaking on Free/Open Source Software Development as an Approach to Global Software Engineering; Christian Engblom from Ericsson Finland R & D, speak-

ing on Waterfall to Agile in Global Software Development at Ericsson; and Kaj Arnö from SkySQL Ab presenting a case study: MySQL AB: Doing Open Source Business with a Global Community of Developers.

I gave an industry-based technical session called “An Industry-based Examination of Distributing and Outsourcing Testing in Agile Projects” this research paper examined the realities of implementing Agile practices combined with offshore outsourcing.

Most of the participants were from Europe. They have very different flavors of offshoring and outsourcing than typical North and South Americans and Asians are used to. In Europe, you have high cost and low cost labor markets and availability of well trained staff on the same continent. It is most common for partnerships between, for example, Sweden and Hungary or Germany and Romania. There are sometimes similar situation in Asia, for example, between Japan and Viet Nam or the US and Brazil but not nearly as common as in Europe.

The overlapping work time has a great impact on teamwork and available varieties of work time communication leading to few late-night phone calls so common to North American to Asian partnerships. Yet it is still common to have language, visibility and culture issues. Time difference (propinquity), culture, language, visibility and support tools were often referenced in the research presented.

The next ICGSE conference, August 27 - 30th 2012, will be held in Porto Alegre Brazil.

<http://icgse2011.soberit.hut.fi/icgse2012/>

Gartner Symposium IT^{XPO} 2011

Gartner Symposium/ITxpo 2011 is the industry's largest and most important annual gathering of CIOs and senior IT leaders. It delivers independent and objective content with the authority and weight of the world's leading IT research and advisory organization.

When: October 16 - 20, 2011

Where: Orlando, Florida

LogiGear will be attending the conference as an exhibitor and will be able to discuss your testing and training needs . Stop by and say “hello”!

For more information and to register, please visit
[http://www.gartner.com/technology/symposium/
orlando/index.jsp](http://www.gartner.com/technology/symposium/orlando/index.jsp)

29th Annual Pacific Northwest Software Quality Conference

Announcing the 29th Annual Pacific Northwest Software Quality Conference, October 10-12, 2011 at the World Trade Center Portland in Portland, Oregon.

For more information, please visit [http://
www.pnpsc.org/2011-conference](http://www.pnpsc.org/2011-conference)



BLOGGER OF THE MONTH

Vipul Gupta



According to analysts, the growing dependence on software testing to improve product quality is resulting in greater IT budgets getting allocated to the testing activity. In fact, it is being suggested that in the years ahead, software testing will make up nearly 20-25 percent of the software budget. Forrester Research

estimates that the demand for outsourced software testing will account for 28% of IT budgets in Europe and the U.S. within two years.

Recognizing the fact that customers are looking at effectively and efficiently developing quality products, offshore software testing service providers are focusing on improving the analysis and design phases of their offerings and ensuring that they have the capabilities to test earlier in the product life cycle. Their aim is to align their software testing services with the business needs of companies.

In order to do so, they are also changing their testing approaches. The service providers are working in proximity with customers and involving their business teams to drive software product quality. They are establishing strong connectivity and traceability between needs, the product features and the testing that must be undertaken.

All in all, offshore software testing service providers are delivering more value to customers using mixed methodologies of exploratory testing and automation testing. Specialized testing services like performance testing and security testing, act as the extra topping on the pie.

As software testing becomes more complex, based on increasing refinement of the applications and technologies, outsourcers are designing solutions to meet business objectives of the applications being tested and the technologies utilized by applications. They are also building test infrastructure required to test such applications

and selecting the right test methodologies to provide adequate test coverage.

With test automation becoming a key element of software test strategy, these companies are also using more sophisticated, mature testing tools that better integrate with other tools to support collaboration. Many of these tools are based on open source standards and built on common infrastructure that makes them easier to integrate with other life cycle tools. The goal is to use tools that provide a seamless testing experience and enable more holistic, realistic and robust testing. Outsourced software testing sector is thankful to such tool sets.

Interestingly, offshore software testing organizations are using prevention practices as well, like using static analyzers, etc., for early bug detection. At the same time, they are also using commonly available simulation tools to easily create artificial computing scenarios so as to test for exceptions and error paths earlier in the development process. Such tools often provide the functionality to generate tests on-the-go. The [offshore software testing](#) service providers are replacing misleading metrics such as bug counts and test case counts in testing projects by useful metrics such as specifications coverage, model coverage and code coverage.

Truly, today the outsourced software testing has become need of the hour to sustain competition to serve customers efficiently. ■

Vipul Gupta leads the Test Engineering R&D at [Impe-tus Technologies](#). As a resident testing expert at Impe-tus, Vipul works on innovating test engineering methodologies. He is currently focusing on innovating cost effective software testing solutions for desktop, web and mobile applications. He is a cofounder of [NCR Testers community](#) and is contributing to several online testing forums. Vipul has extensive understanding of testing activities as it applies to all phases of the testing life cycle, including specialized testing. He has been involved in designing RUP and Agile based testing frameworks for various organizations including Fortune 500 clients.

FEATURE ARTICLE

Important Lessons Learned in 10 Years of Offshoring



Senior Vice President of LogiGear, Michael Hackett, in a very personal approach highlights his views on 10 key lessons learned in the past decade. From advice on training to building a communication infrastructure, Michael lists the fine points of what makes a successful relationship with your out-source/offshore partner.

In 2000, offshoring was not a new concept as many large companies had already been offshoring and outsourcing for over a decade. What did change, in the past 10 years was how many companies began outsourcing and how many development tasks were being distributed.

It was not only Texas Instrument and Microsoft sized companies distributing maintenance projects, but it quickly became medium and small sized companies outsourcing or opening their own offshore facilities for varying development tasks, maintenance, new product development, and testing.

This pace of outsourcing has accelerated exponentially over the last decade. The necessity of outsourcing/offshoring has also increased. Whereas most people think of distributing software development offshore as only a cost cutting measure, it is more than that.

For example, there are not as many new U.S. software engineering graduates today. This is causing a shortage of skilled developers and test engineers. There is also the factor of quick ramping up and ramping down of project teams influencing more contracts and distributing resources.

However, the downside of more distributed development is increased management oversight. Visibility into work at remote sites is often a common problem as well as frustration with communication.

In the past 10 years, there have been notable events profoundly impacting global distribution. For instance, the emergence of Agile and Scrum caused many groups to re-think *how much and what* is distributed. Most Agile teams are co-located for immediate and easy communication, especially verbal discussion on newly developed functionality.

This immediate and easy communication is supposed to lead to leaner product development, e.g., less documentation. This is often difficult for distributed teams. On the other hand, there is also the impact of communication tools. With the introduction of communication platforms such as Skype, Cisco's UMI telepresence and the full range of instant messaging tools, common communication has become much easier.

In my capacity as a test lead then a test manager at various U.S. companies, I have worked on both sides of outsourcing and offshoring situations. In the late 80s, I worked for a company outsourcing and offshoring software development to Bulgaria, and at another company sending offshore documentation development to the Bahamas. More recently, I am the cofounder of LogiGear where we provide outsourced engineering services mostly based in Viet Nam.

From my experiences—the successes and failures—I will explain what I draw from them. In the following section, I list 10 important lessons learned over the past decade on how to make offshoring a greater success and higher value for you and your organization. For the sake of convenience, I will refer to the various models as distributed or global development.

1. Build a Relationship

- Rome was not built in a day! Neither are good working relationships. Take time and be patient. Think of local/home team issues, problems, misunderstandings—work to avoid these same issues with the offshore teams.
- Governance
 - ◊ Make sure you manage your vendor with metrics against an SLA
 - ◊ Make sure you manage the relationship effectively and setup good management in the beginning
 - ◊ Set up a separate team to manage the outsource vendor
- Remember to focus on the value the distributed team is providing. Communicate and focus on getting the business value you expect; your attention to goals will transfer to the distributed teams.

2. Invest in Your Partner

- Invest in Training
 - ◊ Deliver training in the test and development process, technologies, domain, tools, communication, reporting, working across cultures, etc.

- Build trust
 - ◊ Don't think once the process and tools are setup, you're done. You are only at the beginning! There is much more to offshoring than just setting up the operation.
 - ◊ Make time to get to know the new staff. Have the new staff get to know the onshore staff. Take photos. Make a board with everyone's pictures, names, and role.

3. Focus on Communication Infrastructure

- Build a communication infrastructure and tracking system. Don't rely on instant messaging.
- Make a communication plan for methods, days of the week—especially Sunday and Friday—preferred times of communication to make reference easier, boost task tracking, reinforce agenda, task lists, action items, etc.
- Build project wikis and document sharing formats.
- Conduct training in communication skills.

4. Visit Your Team

- Build rapport and trust. Visitations make you and your team *human* and *real* instead of merely a headcount and a boss. Visits have profound impacts on working relationships and building bridges of understanding.

5. ALM- Implement an Application Lifecycle Management Tool

- Implement an application lifecycle management tool or set of tools, including user story/requirements management, change control, test case management, test automation tool interface, issue tracking, source control, ftp, and/or code delivery. This will centralize processes, tools, focus and reduce a lot of uncertainty. It can be a single source location for all answers.

6. Its human!

- Do not underestimate the power of human connection.
 - ◊ It is easy for projects to get off track. If it was all fun and games, software development would be teaming with people. It's not.

◊ There are inevitable problems to be resolved. When I have local team members in different locations that I never see, talk to, have lunch with—the interactions may be rough. Expect the same with offshore teams until bonds are built.

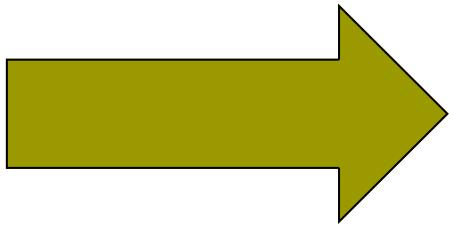
- Understand the culture and its differences: “Cross-cultural communication,” “Working with Americans” or “Working with Whomever” training will have a profound impact on teams.
- Soft skill training and work are always the most important.
- Honesty and trust are more important than test case metrics.
- Human issues run both ways. U.S. teams having late night Sunday phone calls to begin the work week in Asia will lead to job dissatisfaction, retention and political problems. Cultural differences, time zone issues, communication roadblocks need to be talked about, not ignored.

7. Start Small, Do a Pilot

- Choose a small project to begin the relationship, refine expectation, clarify deadlines, build a regular schedule, and focus on good communication.
 - ◊ Choose your project to distribute carefully
 - ⇒ You don't want to send the key to your company's revenue growth as your first assignment.
 - ⇒ Start with low hanging fruit that might help setup the vendor for success.
- Make sure you understand what it takes to do the work before you send it offshore.

8. Home Team: pay attention to how this is going to be for the employees at home

- The home team may have an altered career path and need different skill training.
- Training for the home team is often neglected, consequently they will need communication and cross-cultural training as well.
- Be careful what you ask your domestic employees to take on. You don't want them resentful.
 - ◊ Do you have employees who want to work into the night with offshore teams?



- ◊ Can you stagger their hours so that they can work effectively?
- ◊ What about quality of life impact?
- ◊ Are the tasks distributed fairly?

9. Be Realistic

- Expecting perfection or expecting a miracle can be dangerous.
- Expect glitches at the beginning; it's how the teams grow and adapt.
- Have patience and work through problems, just as you would with building a team in your home office.

10. Build Subject Matter Expertise

- Home teams are sometimes staffed with subject matter experts or people who will easily relate to and model your users. This is most often not the case for your offshore teams. Make an effort to introduce teams to your target users—their needs, work habits and patterns. Show your competitor's or similar products. Describe the varieties of users you expect to use the product and what they want. Your life will be easier, your goals will be accomplished, and quality will be higher when you transfer business expertise to your distributed teams.

In the end, it's the relationships and people that matter the most. In all matters, remember the Golden Rule: treat others as you want to be treated. You can't fail! ■

Key Words

Outsourcing is taking work that used to be done by internal teams and sending it to people outside the company. It has been going on since the beginning of business. Every company used to have its own payroll department. Now, most companies outsource their payroll to ADP or Intuit or their bank.

Offshoring is taking work that used to be done in a home office and sending it to an office of your company's in another country, most typically moving from a High Cost Center (HCC) to a low cost center (LCC). These offices are sometimes called captive offices if they belong to your company.

There is a newer situation called **nearshoring**. After 2000, offshoring from the U.S. mainly went to India. That's about as far offshore as American companies can get with a twelve hour time difference.

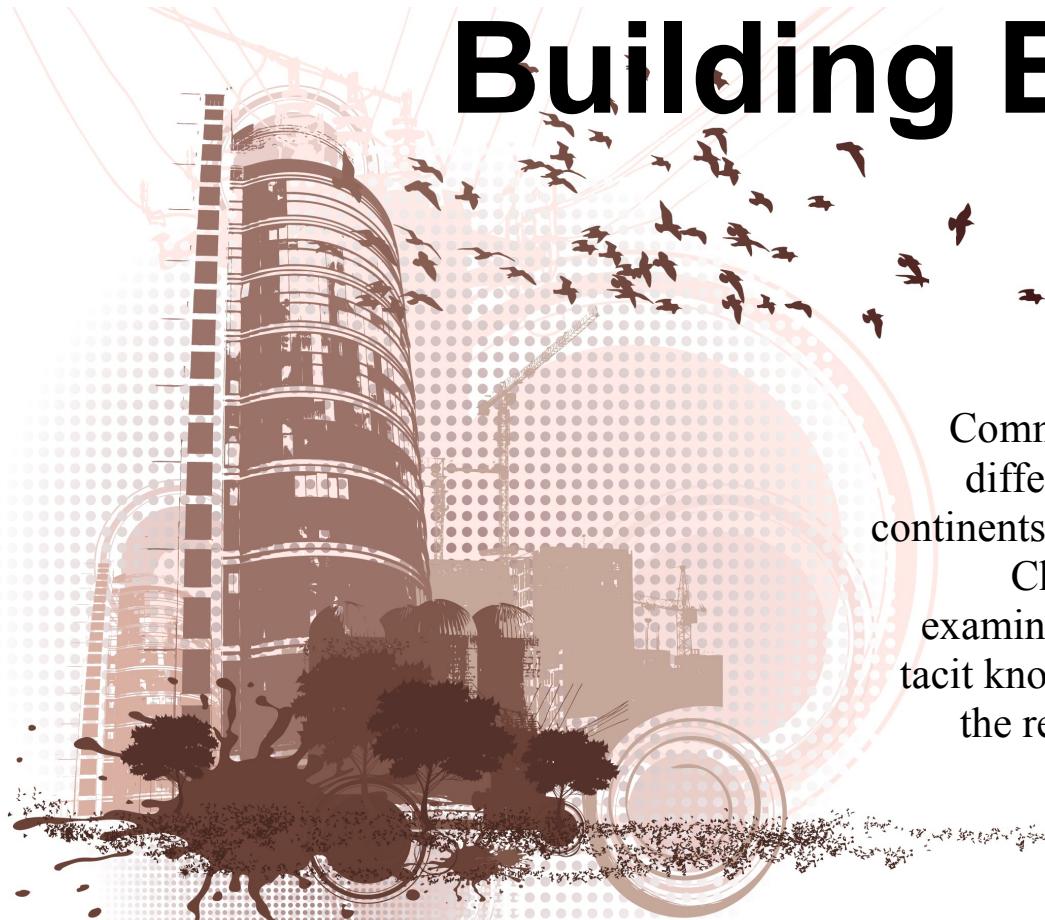
An example of **nearshoring** is an American company sending work to Canada, Mexico, Central America, Brazil or another South American location that is closer in time zone allowing an overlap in normal working hours. These countries may not be as low cost as some Asian or African countries, but the time overlap is seen by some companies for higher productivity and less stressful work.

Outsourced offshoring is the most common situation we read about today. It is a company with a home team taking some of their work and sending it both outside the company and to another country—whether its companies the size of Citibank or a Web2.0 startup with twenty employees—sending their software development to a small boutique outsourcing company in Nicaragua.

Distributed software engineering and **global development** are commonly used phrases signifying any variety or combination of distribution arrangements: multi-sourcing, combinations of offshoring, outsourcing and multi-site development.

RELATED ARTICLE

Building Babel



Communication between differing cultures across continents creates challenges.

Christopher Williams examines how explicit and tacit knowledge strengthens the relationship between client and vendor.



How can IT offshore outsourcing vendors in countries such as India gain knowledge of their western clients such that they become able to work more effectively for the benefit of their clients?

This question lies at the heart of offshore outsourcing, but it is one that is often misunderstood.

I recently conducted a study looking at this issue from the perspective of software development engineers located offshore within Indian vendor firms.

This study is published in *Information Systems Journal* (Williams, 2010). I approached this issue from a knowledge-based perspec-

tive. This line of reasoning suggests there are two types of knowledge that need to be accounted for: explicit knowledge and tacit knowledge.

Explicit knowledge can be articulated and codified, and can be transferred from client to vendor in the form of documentation, training and the like. This is essentially a formal knowledge transfer. Tacit knowledge is gained through experience and an individual's repeated interactions with others in the task environment. In offshore outsourcing, this type of knowledge is most likely to be transferred by embedding the offshore engineers tightly within the client organization (e.g., through experience and frequent contact with the client).

In my study, I included a third element that has the potential to explain the effectiveness of knowledge transfer from client to vendor. This related to the extent to which employees in the vendor organization relied on informal discussions in the offshore

location in order to share understanding of the client. I argued that when such informal discussions about the client take place in an inappropriate way, offshore engineers may actually end up confused or unclear about client needs and organizational dynamics.

An additional aspect to the knowledge-based approach to this issue is to recognize that knowledge is not fully transferred simply by transmitting it. The knowledge must be internalized by those receiving it and then actually utilized. It is only when the new knowledge gained from a client firm is utilized by a vendor firm that we can claim the knowledge has been transferred.

My study was based on a survey of 140 software engineers located offshore in India. The respondents worked for a range of vendor companies, including the biggest names in IT offshore outsourcing. They all were assigned to software development projects for western clients. Some of the respondents in the sample had already had onshore placement experience, and some did not. I was therefore able to test the effect of onshore experience gained by the engineer.

I found formal knowledge transfer through training to have a strong impact on the perception that the engineer understands their client. This was significant both for engineers with and without onshore experience, although it was slightly stronger for those with onshore client experience.

In terms of embedding the offshore engineer with the client (measured through past experience working with the client, as well as frequency of communication with the client), I found this embedment to impact the offshore engineer's understanding of the client only for engineers who had already had prior onshore placement with the client.

In terms of the vendor engineer's reliance on informal discussions with offshore colleagues to understand the client, I did not find a significant impact (although coefficient signs were, as expected, negative). Interestingly, for engineers who had not had any prior onshore experience, both formal knowledge transfer and client embedment acted to reduce reliance on informal discussions amongst engineers in the offshore location.

What do these findings mean for managers in IT offshore outsourcing arrangements? Firstly, gaining an understanding of the client by individual engineers within the vendor firm is essential to allow the vendor firm to apply knowledge effectively and in the interests of the client.

Even young, newly recruited offshore engineers need to have exposure to client knowledge outside of the constraints of an IT system or project. They need to understand the wider organizational structure and dynamics of the client organization, as well

as the industry pressures facing the client. These elements of knowledge help engineers in interpreting priorities set by the client and seeing where their work fits into the bigger picture.

Secondly, client firm managers continually need to work on transforming complex knowledge within the client organization into codified forms that can be used in training vendor engineers offshore. Keeping offshore engineers up-to-date is critical, especially if the client's own environment is changing.

Thirdly, opportunities for offshore engineers to gain an embedment with their onshore client should be pursued. This can be achieved from an offshore location over time and with frequent communications with the client. It should be clear to clients that this is in their own best interests.

In terms of onshore placements, it is unrealistic on most IT offshore outsourcing contracts for all engineers to receive this exposure. Building client embedment from an offshore location is possible and should be encouraged, especially for those with no opportunity for an actual placement onshore.

In summary, the transfer of client knowledge to individuals within a vendor firm is essential to the success of IT offshore outsourcing arrangements. The results of my study suggest that managers on both sides of the equation have to work at formal and informal modes of knowledge transfer, as well as appreciating the differences in utilizing client knowledge that arise between engineers that have had experience with the client onshore, compared to those that have not. ■

Reference

Williams, C. 'Client-vendor knowledge transfer in IS offshore outsourcing: insights from a survey of Indian software engineers', *Information Systems Journal* (forthcoming, DOI: 10.1111/j.1365-2575.2010.00354.x).

This article was first published by BCS, The Chartered Institute of IT.

Christopher Williams, Ph.D., MBCS, has worked in the field of software development in technical, management and consulting roles since the mid-1980s and is currently Assistant Professor of International Business at the Richard Ivey School of Business in Canada, where his research interests include the management of offshoring transitions, knowledge creation and transfer and entrepreneurial initiatives within multinational enterprises.

SPOTLIGHT INTERVIEW

Guido Schoonheim



As CTO of Xebia and highly experienced in offshore testing in India, Guido articulates his methods in addressing common challenges faced by the in-house and offshore teams. He weighs heavily on strategic tactics as well as key cultural aspects to execute efficient and effective Agile methods.

1) I work at a US-based company and we have been offshoring for a few years. We have a good working relationship with the offshore team; the work and communication is good. We are now implementing Scrum and want to continue having these offshore folks on our team. We have been distributing mainly testing tasks to them. What would you suggest is the best work to distribute to them?

The goal of Scrum is to deliver the right fully tested production ready software at the end of each sprint. That means you can no longer hand off testing as a separate phase. Testing now needs to be incorporated as a continuous team activity into your sprints. This means that your offshore testers should become part of your Scrum teams and work with you on getting tasks from “to do” to “ready for testing” to “done.” Creating automated test cases should start in parallel with actual development of features.

2) We have people we used to call testers integrated into *The Team*, fully tasked during sprints. They can't keep up with building and maintaining our automated regression suite during sprints. The regression contains longer business scenarios, workflow, and interoperability tests. We want to outsource the regression suite. What do you think?

In Scrum, there is only [one] role responsible for delivering a fully tested increment and that is *The Team*. That means that the division between testers, regression testers and developers is an artificial one, secondary to team goals. As a team member you will have specialties that you use, but at the end of the day you need to maintain your automated tests as a team. That means that you should never write more code than your team can test. Instead developers need to help with the test automation and testing where necessary.

Otherwise you increase technical debt and put the project at risk. And that is irresponsible, even if you mean well! Test automation is the single most important thing for a team to keep a high velocity. Never ever outsource anything this crucial outside your team because then you can not commit to delivering fully tested software anymore, when part of that is [out] of your hands

3) The people who used to test in our old style development projects are now integrated into our Scrum teams. They have great user knowledge, are extremely helpful for details on user stories, run our continuous integration/build validation process and conduct user story validation tests. These members do not write code. What other things can former testers do on a Scrum team?

Building good software is a matter of providing instant rigorous feedback on your quality. Automated functional regression testing is one of the most important aspects to getting this right. Focus on getting this in place. After creating the initial setup as a team, it is usually the (former) testers who fill the tooling with cases and maintain the test cases during the sprints to resolve any broken tests. When you have full regression and keep up with it, the team generally needs two testers for every four developers.

If you have this in place and have extra capacity then I suggest work that is client facing, such as developing user stories, help with backlog grooming and other tasks that require a mindset from a client perspective. Or you can of course reduce your team size.

4) I work at an outsource development company. We use Scrum with many clients. Some are successful, others are not. The projects that do not run so well are all, as Jeff Sutherland calls them, Scrumbutts. They pick and choose what practices to implement. In all those companies, it is not standard to invite the outsource team to the Sprint Retrospective if one is in place. Do you have any suggestions how I can help the team improve without going to the retrospectives?

No team is working to their full potential when they form up as a Scrum team, and that is not the point. The point is to generate quick feedback and heighten learning so that you become better rapidly, thus achieving hyper-productivity. That means team retrospectives are crucial.

Before giving you more options let me say that there should be no excuse for a Scrumbutt implementation. Your first priority should be to create a proper Scrum. When working with offshoring or outsourcing, that means acknowledging that you are One Team and should be treated as such. That means involving everyone in the full Scrum ceremony. This is worth fighting for and getting in trouble over, as you are all equal in a Scrum team.

If your client refuses a full retrospective with business + onsite team members + offsite team, then have your own smaller retrospective previous to the full one with the onsite + offsite team members. That way you can discuss your points and the points of the onsite team members and they can take the relevant points to their local retrospective and share the results with you afterwards. Stacking retrospectives this way might be helpful in creating a safe environment to share issues, however, it is also added inefficiency that you should try to avoid.

Should you be unable to organize this then share your unasked feedback to the onsite team members and send them a questionnaire to fill out in return.

5) Are there any situations where you find Scrum does not work, e.g., safety-critical, regulated software? Large cross-team integrated software systems? Device development where the software will have one big release and no iterations?

Managing complexity by up front design has proven not to work. We need to divide large problems into small chunks and learn rapidly about the domain and the problem at hand. Probably we also need to fail 2 or 3 times when doing something novel, as a part of that learning process. Scrum enables this learning, and combined with proper risk management enables us to fail fast in order to get it right at the lowest cost.

This means that the more complex the system, the bigger the need for an incremental approach!

There are things that you need to add both as deliverables and to your definition of done to get your quality up to 99.999% and to ensure compliancy. After implementing Scrum in mission critical projects—finance and automotive device development—I can tell you that these are in no way exclusive to doing Scrum. The tricky art is in finding out how to divide your work up in the right order, which is more an engineering and analysis problem than a methodology problem.

6) I have a situation where management (the chickens), particularly the sales team wants to keep control of functionality delivery and schedule. But they say, we have to be more agile, stop writing so much documentation and be faster. Do you have any suggestions?

First of all, our main goal is to provide maximum value to our client and /or company. That is why the product owner determines what goes into each sprint. Value however is measured not just by the output of today's sprint, but over the lifecycle of your product.

Documentation should always have a clear identified audience and clear criteria so that you know what “just enough” is. Since creating documentation is not a favorite activity for many people I assume you have a good reason. Try to put that reason into velocity or risk numbers and make clear how much you save by doing so.

On the other hand your management might have a good reason for wanting to speed up, and the discussion about documentation could just be a symptom. Find out why they want to speed up and brainstorm with the team on ways to help reach that real goal, along with other ways to go faster (like getting managements help to finally solve some really nasty impediments that you have institutionalized).

7) I work in an offshore outsourced testing company, some clients have become seemingly very agile but demand a lot of test documentation. Test cases, bug reports, and often, test plans. This isn't lean and I think it points to a trust problem. What can we do when the team cuts back most of their documentation, wants me to be more agile and rapid but asks for a lot of documentation?

It is quite natural for clients to ask for extensive testing documentation. The thing to question is why they are requesting this. This comes from an understandable underlying need to validate your work. So the best way to lighten your documentation load is to be more transparent about your work in another way.

For my clients this means somehow showing a lightweight logical test plan and actual physical test cases along with test results.

This is where test automation can help you (again). By creating your functional test cases in automated tooling right away they can be witnessed later at no extra effort. Using a tool such as Fitnesse (<http://fitnesse.org/>) will allow you to specify both logical and physical cases in a simple wiki markup per story or scenario. To integrate GUI testing into this, have a look at Xebium (<http://xebia.github.com/Xebium/>).

Being smart about your tooling will save you from large additional reporting effort. The cases that you create in your tooling can be your documentation. This also lightens the regression workload considerably, and it can make for a very impressive sprint demo if you use it right.

8) I do most of the non-unit tests on my team. My team does not always value my estimates. Estimating is often difficult. I sometimes have to think about all the possible data and platforms and paths and uses—the majority of which are not captured in user stories, my story-point estimates are usually pretty different than the rest of the team. It's an education process for them and sometimes I am way off. Do you have any suggestions about estimates including testing?

That is a tricky one and teams tend to come up with different approaches. Generally, I see two approaches being used the most:

1. Give separate estimates for each story for development and test. Then see if the amount of work matches the availability of test and development capacity in the sprint.
2. Only estimate development effort during team planning poker, and have the tester(s) remove stories from the sprint until they too can commit to the result. This is often done when there is only one tester, since planning poker across disciplines can be frustrating and does not always make sense.

My preference is the first approach, to involve the rest of the team as much as possible. If you need a sparring partner then ask one of the developers to become tester for one or two sprints. Afterwards they should be able to think with you a lot more. Alternatively, what I have seen work is to have a tester from another team sit in on your planning session and vice versa.

9) What characteristics should we look for when selecting people best suited for Scrum teams?

The most important part is to create a team that works well together. This is just as important as technical competence. I have seen a team of only top notch engineers being a total disaster, because of ego's colliding over every decision. So look at personality types and the ability to communicate and collaborate as well as technical competences. There is no room for prima-donna developers in a team, no matter how good they are.

This comes together in team composition, so think about what makes a balanced team. For instance get a scrum master who is a good communicator and likes organizing, one or two deep thinkers for tough problems, two solid hard workers, a savvy tester and a more junior tester.

Also have a look at Tuckmans stages of Team Development if you want more structured guidance in turning a group of people into a great Scrum Team (http://en.wikipedia.org/wiki/Tuckman's_stages_of_group_development).

10) Do you see any cultural issues implementing Scrum for teams not in the U.S. or Europe; such as self-directed teams in cultures that are more comfortable with follow-the-leader?

Scrum is about liberating the potential of all team members and for different cultures, this works in different ways. Something that helps me understand the differences is Geert Hofstede's cultural dimensions theory (http://en.wikipedia.org/wiki/Hofstede%27s_cultural_dimensions_theory). Scrum seems to have the quickest initial adoption in countries that have a lower Power Distance Index (PDI), meaning less orientation towards hierarchy, such as the Netherlands and Scandinavian countries.

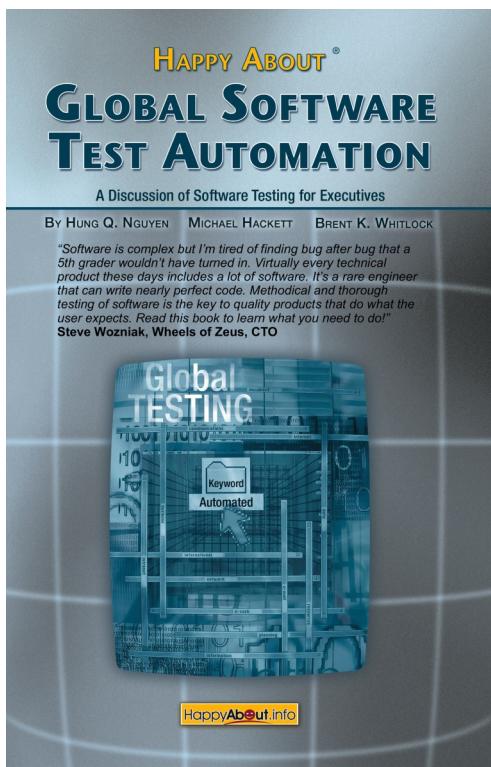
From personal experience I can say that in countries with a strong hierarchical culture, it is crucial to create a strong company and team culture of equality and openness, in order to be able to have a proper Scrum. You need an open environment without strong blame or power influences to get all valuable ideas and feedback from all team members. This is hard when team members worry "what impression will people get of me when I ask this question."

How to most effectively enable your team members works different in every culture and therefore so does implementing a really good Scrum. Just take those specifics into account and you will be fine. ■

Guido Schoonheim is an Agile fanatic with a specific focus on Scrum, organizational patterns and distributed development. In the past, Guido has worked as project manager, agile adoption coach, architect, scrum master, product owner, and of course, as JEE developer.

As CTO of Xebia, he developed the Xebia model for Fully Distributed Scrum model. Fascinated by India with her strong contrasts and infinite possibilities, he believes very strongly in the combination of Agile and Offshoring to get the best of both without compromising on either. With focus on people and their interactions using strong guiding principles the cultural difference, time zones and distance are of no issue at all.

BOOK EXCERPT



Authors: Hung Q. Nguyen, Michael Hackett, Brent K. Whitlock

Paperback: 164 pages

Publisher: Happy About (August 1, 2006)

Language: English

Product Dimensions: 8.4 x 5.1 x 0.5 inches

"Software is complex but I'm tired of finding bug after bug that a 5th grader would have turned in. Virtually every technical product these days includes a lot of software. It's rare that an engineer can write nearly perfect code. Methodical and thorough testing of software is the key to quality products that do what the user expects. Read this book to learn what you need to do!"

Steve Wozniak, Wheels of Zeus, CTO

Global Software Test Automation is the first book to offer software testing strategies and tactics for executives. Written by executives and endorsed by executives, it is also the first to offer a practical business case for effective test automation, as part of the innovative new approach to software testing: Global Test Automation — a proven solution, backed by case studies, that leverages both test automation and offshoring to meet your organization's quality goals.

The following is a review from Scott Barber, Chief Technologist at PerfTestPlus.

"Happy About Global Software Test Automation: A Discussion of Software Testing for Exec-

utives is an absolute must read for any executive in a company that develops, customizes or implements software. For years, software testing has been notoriously under valued and misunderstood by corporate executives. While leading software testers have been trying to get their message to executives from the bottom up, they have been largely unsuccessful. This book has the potential to change that.

With this book, all it takes is one business trip and you'll be able to engage in risk and ROI based planning to minimize many of the challenges and expenses your company faces related to software through the efficient and effective application and management of software testing." ■

To obtain a free PDF copy of the book, please email logigearmagazine@logigear.com.

Chapter 6: Strategies and Tactics for Global Test Automation

In this chapter, you will learn the following:

- The benefits of Global Test Automation
- The seven-step process of developing a Global Test Automation strategy and roadmap

Introduction

In the previous chapters, we have discussed software testing and a number of pitfalls associated with software testing. In particular, we have discussed manual software testing, test automation, and outsourcing/offshoring of software testing. We have also presented a number of suggestions to improve the results in each of these areas, responding to the pitfalls you may experience. In this chapter, we present a comprehensive methodology to address the pitfalls and create a successful test effort. This methodology entails an array of powerful strategies and tactics for Global Test Automation that creates successful outcomes by intelligently combining manual software testing, test automation, and outsourcing/offshoring of software testing.

What is Global Test Automation (GTA)?

We can all agree that software testing is necessary. We need to test software to be sure that it performs the functions it is designed to perform, under the conditions in which it will be deployed, and in a responsive and user-satisfying manner. We also know that manual software testing, software test automation, and outsourcing/offshoring all inter-relate yet have distinct characteristics with unique issues that need to be addressed. By understanding their pitfalls and suggestions for improvement in these areas, you will gain a fuller understanding of how Global Test Automation can create a holistic solution for your organization's testing needs.

Software testing takes time and costs money. As an executive, you want to have a strategy that will provide the needed results while saving both time and money. The 2 by 2 chart in Figure 8 shows strategies for saving time and saving money. But how can you save both time and money? That is where the Global Test Automation strategy comes in. It saves time by speeding up the test process, saves money, and provides the needed results.

An Exercise for the Reader

The first step in establishing a test strategy and methodology is to assess where your organization is currently in its test strategy. To help you internalize the material in this chapter and apply it to your organization, we have provided this exercise for you to begin to evaluate your organization's current test strategy. Please consider the following questions and answer them for yourself in regards to your organization.

1. How much, in terms of percentage to revenue and/or development dollars respectively, do you budget for software testing?
2. What is your percentage of automated tests versus manual tests?

3. What are the three things that you want to change in your testing strategies to optimize the quality of your re-released product?

4. What are the three things that you want to change in your testing strategies to optimize the ROI on your test spending?

An Illustration of the Issues

After working on this exercise, you see how important visibility is in making management decisions regarding testing. Visibility gives you the power to make the right choices for the strategic direction of your company. You need visibility into the test process to set the best strategic directions for testing, as well. The right quantitative measurements, test metrics, can give you that visibility. Automation alone won't necessarily provide you with that visibility, but it can help. Automation isn't a silver bullet, but it's a part of the solution.

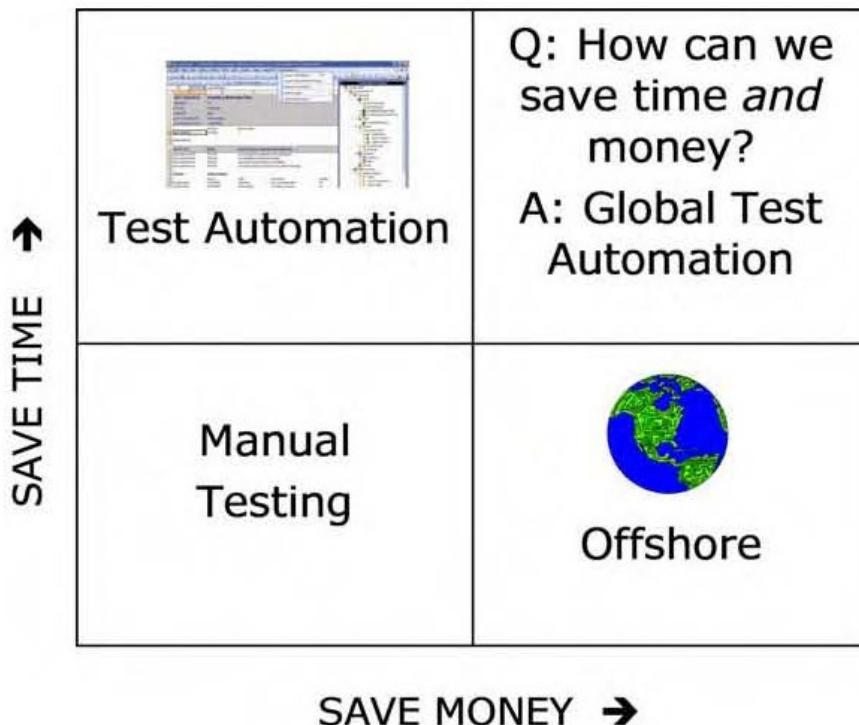


Figure 8: The Global Test Automation 2 by 2 matrix.

Global Test Automation 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.

That is a mouthful, so let us break it down into the critical aspects and discuss each one independently.

Global Test Automation is the integrated solution for:

- Software test automation
- Outsource/offshore software testing
- Global team management

The main problems with manual testing are that it is too slow, too expensive, and does not scale. Software test automation can address these issues, if strategically and skillfully applied. However, so long as applications are meant for human end users, test automation will never entirely replace the need for human testers. No matter how sophisticated test automation tools become, they will never be as good as human testers at finding bugs in an application. Human testers will instantly notice subtle bugs that are almost never detected by test automation, particularly usability bugs. Automated test tools cannot “follow their instincts” to uncover bugs using exploratory and ad-hoc testing techniques. By freeing manual testers from having to execute repetitive, mundane tests, properly deployed test automation enables them to focus on using their creativity, knowledge, and instincts to discover more important bugs.

Strategy Formulation

The steps in creating an effective test automation strategy are to assess your testing capability, define a good methodology, select the proper tools to implement

this methodology, and put people in place with the proper skills and training to successfully implement the defined test methodology using these tools. Common problems in test automation include its potentially high cost and inability to obtain the desired ROI due to a lack of high productivity and anticipated savings. Scalability, reusability, visibility, and maintainability can be problematic.

The Global Test Automation strategy addresses these issues in the four phases of test automation: deployment, production, execution, and maintenance. By providing visibility, the GTA strategy utilizing the Action-Based Testing (ABT) methodology greatly improves manageability, and consequently improves the test coverage and test quality. It also addresses scalability and reusability. These four benefits of GTA (scalability, reusability, visibility, and maintainability) combine to effect high productivity (see Figure 7 in Chapter 4).

The main problems with outsourcing and offshoring software testing include communications problems due to cultural issues and time zone differences and incorrect skill sets. The GTA strategy provides a structured approach that addresses these problems, including a combination of clear, repeatable and manageable processes, appropriate training, powerful tools, and effective management procedures.

The strategy of Global Test Automation is central to its success. The strategy provides a bridge between the problems of outdated manual testing, attempts to address the speed problems with test automation, and attempts to address the cost problems with outsourcing and offshoring of software testing, with the desired end result being an integrated Global Test Automation strategy that achieves both time and cost savings with the desired testing benefits. Global Test Automation makes use of a combination of powerful test automation technology for distributed teams for speed, world-wide resources for cost control, and best practices in management of software testing.

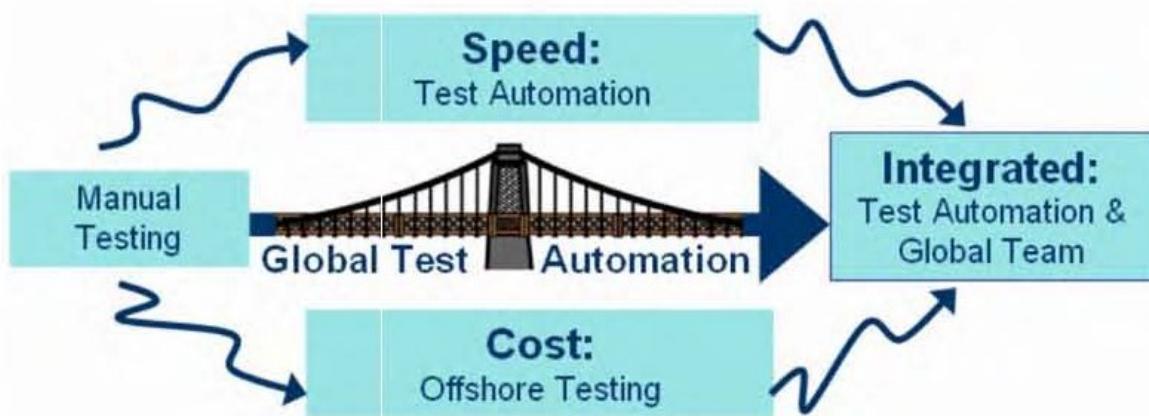


Figure 9: Global Test Automation bridges manual testing with test automation and a global team to lower costs and speed up testing.

There are seven steps to establishing a successful Global Test Automation strategy in your organization. The steps are identified below:

1. Assess your testing needs.
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.
7. Measure, set goals, and optimize.

We will describe each of these steps in the following sections.

We call the strategy development methodology for Global Test Automation “SP3™”, which is named after the first initials of each of the critical elements in the strategy development process. Figure 10 graphically illustrates this concept:

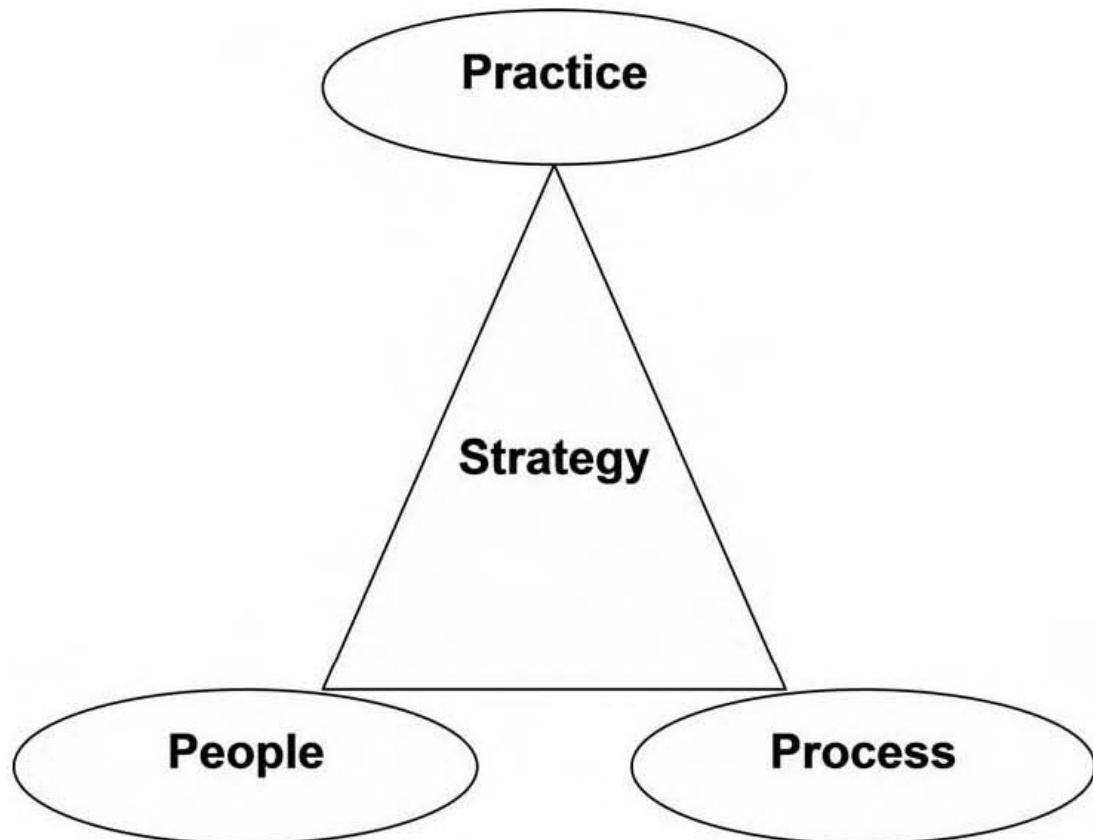


Figure 10: The SP3™ Strategy Development Methodology for Global Test Automation.

A strategy to integrate people, practice, and process for success—the graphic describes that test strategy consists of inter-relationships between people, process, and practice. *Process* incorporates the lifecycle of testing. *People* incorporates the combination of skill sets, communication, and morale. *Practice* involves methodologies and tools.

2010 GLOBAL TESTING SURVEY RESULTS

Offshore



From left to right; Patuxay in Laos, Grand Palace and Wat Arun in Thailand, People's Committee Building and Thin Mu

Data was compiled and analyzed by Michael Hackett, LogiGear Senior Vice President. This is the sixth analysis of the 2010 Global Testing Survey Series. More survey results will be included in subsequent magazine issues. To read past surveys, visit <http://www.logigear.com/survey-response-overview.html> .

Part 1- The Home Team

HT1. Do you outsource testing (outside your company)?

	Response percent	Response count
Yes	87.5%	7
No	12.5%	1

Analysis: You can see from the varied results in this section, many people and organizations are conflicted about distributing work. At the same time, for most respondents, the outsourcing/offshoring is effective, the teams are competent but not trusted, and they would not do it if they had the choice. The results are an indication that we have work to do!

HT2. Is your outsourcing/offshoring (any variety) successful/effective?

Yes	87.5%	7
No	12.5%	1

Analysis: It is very good that so many of these organizations see their outsourcing/offshoring as successful. There is a lingering notion that some teams are forced into unsuccessful distributed teams based on business necessities. This is not the case.

Survey



Pagoda in Viet Nam, Angkor Wat in Cambodia, Twin Towers and Kuala Lumpur Tower in Malaysia, and Merlion in Singapore.

HT3. What is biggest impact of outsourcing/offshoring of testing

Faster product release	12.5%	1
More test time	25%	2
More effective testing	0%	0
More technical testing	12.5%	1
More automation	12.5%	1
Slower releases	0%	0
Less effective testing	12.5%	1
More management oversight	25%	2
No difference in test effort	0%	0
Successful projects	0%	0
Failed projects	0%	0
Better project team morale	0%	0
Worse project team morale	0%	0

Analysis: The truth about outsourcing and offshoring is that it leads to more management supervision. This has been found many times in surveys of all levels and varieties of outsourcing. The good news is you will get better at leading and managing.

The bad news is the increased time and effort needed to get the same work done. The range of other answers is positive, except for teams getting less effective testing from distributing work.

HT4. Is the outsourced/offshore team respected and trusted to the same level as the internal team?

	Response percent	Response count
Yes	37.5%	3
No	62.5%	5

Analysis: The “No” answer being so high is problematic and common, yet gets to the heart of all other problems with outsourcing and offshoring; the remote team is often not respected or trusted like the home team.

The reasons for this are many and spring from shortcomings on both sides; ranging from unrealistic expectations by the home team of immediate ramp-up and smooth sailings to incompetent teams. Regardless of why, the problem of mistrust must be resolved or the problem is guaranteed to get worse. Angry teams and high staff turnover can be the next step in unresolved situations.

HT5. Do you view the outsource/offshore test team as competent?

Yes	75%	6
No	25%	2

HT6. How much time and effort is spent training the out-sourced/offshore team?

None	12.5%	1
Little	25%	2
Enough	37.5%	3
A lot	25%	2

Analysis: Training is the key to any successful work distribution. More important than any process or tool, training builds trust as well as skill. That many organizations do not train the distributed teams enough is a problem.

HT7. If you had the choice to outsource/offshore or not would you?

Yes	37.5%	3
No	62.5%	5

Analysis: I am surprised with this answer in that although distributing work needs more management oversight time—a negative—for most other responses, these organizations seemed happy with their work distribution arrangements.

HT8. Please share a success or failure story about offshoring/outsourcing that would be interesting and informative for other test teams.

1. “Offshore team members have to spend time onsite to better understand domain knowledge and better understand team structure, roles and responsibilities. Leads need to have daily communication with offshore, to communicate sense of urgency, which is not perceived the same way by remote locations. When

team gets large offshore, think about sending onsite folks on long assignments offshore.”

2. “I have built 5 QA ODC (offshore development center). The keys to success is standard process, good resources, effective knowledge transfer and ongoing engagement. On and offshore need to follow the same process, this enables resources to ramp up quickly. Good resources, we screen all our offshore candidates when we build out initially and then allow the vendor to chose junior level shadow resources who are brought up to speed on the vendors time.

Effective knowledge transfer, knowledge transfer is bi-directional and continuous. Some of our best process improvements such as “video taping of complex defects” has come from offshore. Offshore resources are professionals, treat them as such. Ongoing engagement, an engaged resource is a productive resource. Rotate and cross train to keep people interested and have backups.”

Part 2 The Distributed Team

DT1. Is your team respected and trusted to the same level as internal te

Yes	50%	8
No	50%	8

Analysis: With half the outsourced/offshored teams feeling no respect or trust, this high percentage lines up with the home team’s similar response. It is a serious problem that so many teams feel they are not trusted.

DT2. Do you do an effective testing job?

	Response percent	Response count
Yes	86.7%	13
No	13.3%	2

DT3. Do you view the home/main corporate test team as competent?

Yes	64.3%	9
No	35.7%	5

Analysis: This is a high and interesting number of groups that do not view the home team as competent. It is a direct comment on the relationship between distributed teams.

DT4. How much time and effort is spent training your team?

None	7.1%	1
Little	50%	7
Enough	42.9%	6
A lot	0%	0

Analysis: As I said above, training is the key to every successful distributed project. More than half the teams that responded felt they are not adequately prepared for their work.

DT5. Do you have tools to support effective communication and quick access to information?

Yes	86.7%	13
No	13.3%	2

DT6. If you could fix one thing about the home office test team you what would it be?

1. "Stop looking at offshore team as your competitor
 - Be ready to move up the value chain when you want to introduce offshoring (actually, demarcate key contribution areas from home & offshore teams).
 - Understand the difference between managed offshore resources and unmanaged onsite consultants.
 - Use offshore teams to complement onsite teams and get the best out of both worlds."
2. "Make them more cooperative."
3. "Provide more training to offshore."
4. "Provide better documentation of tool/features being tested
 - Test plan details
 - Focus areas
 - Provide overall objectives of the team, so we don't lose sight of the forest while going for the trees."
5. "More clear requirements"

DT7. Please share a success or failure stories about offshoring/outsourcing that would be interesting and informative for other test teams.

1. "What I would like to quote here is my experience as a manager of an offshore team working for U.S. based financial services client. The most difficult part for me was to make the client 'QA Manager' understand that the offshore team is a team of managed resources. They always thought of it like bodies being shopped to them and they have no management support or that they have to manage them individually.

I had to work for over 6 months without being recognized as a manager by the client. I prepared the ground for 6 months, created a lot of data with respect to team members and projects, metrics, etc identified improvement areas, training needs, etc and once I visited the client and presented this to them, they were then able to appreciate that the team is managed and they don't have to micro-manage."

2. "We get the job done with very little help from the outsourcing location."
3. "The 'rotational' model does not always work due to incompatibility between the rotated resources in client environment; requires higher degrees of management than originally anticipated." ■

**2010 Global Testing Survey
Results Available Online
([http://www.logigear.com/
survey-response-
overview.html](http://www.logigear.com/survey-response-overview.html)):**

Overview

Agile

Automation

Test Process & SDLC

Methods

Tools

Metrics

Viet Nam's Next Generation

A young country emerging quickly makes way for its new leaders.

She stands tall, svelte in figure with long black hair pulled to a pony-tail, a physique typical of Vietnamese women. She is poised and speaks English fluently with the exception of her native accent. At 24, Hong Mai Nguyen is the oldest of two daughters and one of six women who graduated from Huflit University in Ho Chi Minh City with a degree in information technology in 2009.

Hong Mai has always been interested in computers since the beginning. She recalls having written her first script and thought it was amazing. "Writing a script and having it run successfully is a good feeling," says Hong Mai.

Hong Mai's parents were high school teachers but have since retired. It was certain that their two only daughters must continue on to higher education, the elder in IT and the younger in economics. Hong Mai currently lives with her family, almost an hour drive away from work and chooses to spend her weekends with her family indulging at the house or eating out at local restaurants. As she speaks in perfect English, she exposes her unfaltering determination to succeed in life.

However, her tone softens when asked about what she recently did with her family. "I took my mother for the first time to watch a movie in 3D, it was *Priest*," she explains smiling with her head gently tilted down in humility. "She was not scared, but my sister and I were. She knew it was all fake. She wished she had seen *Avatar* in 3D."

The young and old generations are experiencing Viet Nam's rapid progression on the international scene. Since opening its doors to the world in 1992, the explosion of limitless opportunities burst within the country's borders. Hong Mai is part of the 65% of the Vietnamese population under the age of 30. Her ambitions and her interests reflect the growing number of young professionals in Viet Nam who are taking advantage of their country's political and economic climate.

Upon becoming a member of the World Trade Organization in 2006, Viet Nam provides an open-door policy to foreign investors along with a strong support for a global community with numerous English language centers found in Ho Chi Minh City

and Ha Noi. With labor costs over 30% less expensive than India, Viet Nam steadily makes a presence in the top 20 of Global Services Media's Top 100 Outsourcing Cities 2010 Poll.

The country continues to emerge as a highly sought after off-shore/outsource destination who's literacy rate according to UNICEF is in the top 96%. Vietnamese culture mirrors its Chinese counterparts with their hardworking attitude and strong family ties. The unity shared for a nation's progression as a world player encourages the next generation to prove themselves in the international scene through all channels available.

Hong Mai could never have guessed that posting her resume online a few weeks before graduating college would land her an interview with LogiGear and put her developer skills to good use in software testing. But when opportunity knocks, it's always a good idea to open the door. ■





TestArchitect™

YOUR
HIGH
VOLUME
TEST
AUTOMATION
SOLUTION

TestArchitect™ features:

- All-In-One Solution: Test Management, Test Development and Test Automation
- Action Based Testing™ Methodology
- Built-In Customizable Automation
- Remote Test Execution
- Customizable Dashboard



Software Testing

LOGIGEAR MAGAZINE
SEPTEMBER 2011 | VOL V | ISSUE 6

United States
2015 Pioneer Ct., Suite B
San Mateo, CA 94403
Tel +1 650 572 1400
Fax +1 650 572 2822

Viet Nam, Da Nang
7th floor, Dana Book Building
76-78 Bach Dang
Hai Chau District
Tel: +84 511 3655 333
Fax: +84 511 3655 336

Viet Nam, Ho Chi Minh City
1A Phan Xich Long, Ward 2
Phu Nhuan District
Tel +84 8 3995 4072
Fax +84 8 3995 4076