SYSTEMS DEVELOPMENT MANAGEMENT

REALIZING THE ROI OF OUTSOURCED DEVELOPMENT

Vijay Sikka

INSIDE

THE CASE FOR GLOBAL SOFTWARE DEVELOPMENT AND TESTING

The world is becoming a global village. Increased communication capabilities, computer networks, and language compatibility are challenging older models of doing all software development and testing in one large in-house location. Companies are following a distributed development model with development offices around the world and development as well as testing are happening round-the-clock. For example, Microsoft, SAP, and others have development centers in the United States, Europe, Israel, and India; IBM, Hewlett Packard, and Electronic Data Systems are developing software and outsourcing support from Mexico, India, and Brazil; BearingPoint, formerly KPMG Consulting, is serving software development out of China.[1] There are numerous other examples of development centers located in Ireland, the United Kingdom, Singapore, Canada, Russia, and others.

Benefits to Global Software Development and Testing

There are several benefits to global software development and testing.

PAYOFF IDEA

Two previous articles, "Maximizing Outsourced Software Quality and ROI" (32-10-90) and "Selecting and Managing an Outsourcer" (32-10-91), in this three-part series described the software development life cycle and outsourcing software quality assurance (SQA) and testing, and provided checklists for selecting and managing outsourcers. These articles also discussed best practices and key guidelines for SQA and testing that are suitable for outsourced offshore services. This final article continues the discussion of how best to leverage a co-sourced model for maximum profitability while keeping costs in check. It also presents models to compare the return-on-investment (ROI) on in-house, outsourced, and co-sourced SQA teams.

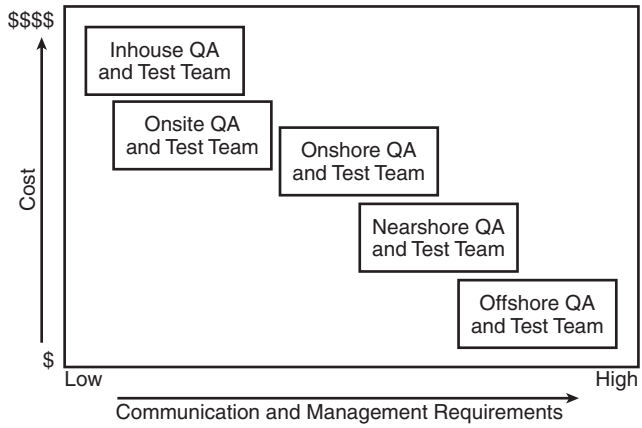
24/7 Software Development Cycle. Development continues both day and night. Through the use of effective source control, bug tracking systems, and project management tools, it is becoming possible to have geographically distributed teams cooperating in development.

Diversified Talent Pool (Nonreliance on Local Hiring Markets). Language barriers around the world are becoming irrelevant these days. English is widely spoken and technical education and training are becoming standardized. All of this has led to growing opportunities of hiring talented and qualified software developers and testers. Job market decline or glut in one country does not hurt the prospects of maintaining and adding to the workforce of the company.

Proximity to Software Consumers and Marketplaces. Several countries have growing software and technology systems demand with their growing economies. In addition to North America, the European Union, and Japan, countries in Asia and Latin America are rapidly becoming major software and IT product consumers. The need for localization and internationalization of software and systems is increasing and the development centers of major companies located in these countries are performing these tasks for local software consumption.

Cost Savings. Easy availability of talented and educated workforces in countries such as India, Brazil, China, and Mexico and reduced cost of communication infrastructure are enabling companies to enjoy reduced cost of software development. Coupled with the market slowdown and economic pressures in the United States, the European Union, and Japan, this has provided an opportunity for expansion of outsourced development centers to these countries.

EXHIBIT 1 — Cost Management Requirements Matrix



Technology Is Enabling Global Software Development and Testing

As the project management and test tools expand, it is becoming easier to manage tasks across teams in different geographic areas. Code management and bug tracking systems are increasingly global in their deployment, and maintained and updated in real-time. Software development for diverse platforms is increasingly being standardized. Software architecture is becoming increasingly component based with a rising number of object-oriented frameworks (e.g., Microsoft.net, SunOne, SourceForge, and others). Web Services are revolutionizing the way software and applications are delivered. The emergence of common standards is facilitating the use of standardized third-party components; for example, J2EE for application servers, JSP/ASP for the presentation layer, LDAP for directories, and PKI for security.

Global computer and telecommunication networks are enabling faster and cheaper communication, including video, audio, and data. Communication vehicles have become reliable, seamless, and secure over the past decade. The Internet has facilitated a cheaper medium to communicate through instant messaging, chat, and sharing.

Success Criteria for Global Software Development and Testing

Outsourced and offshore locations are no longer cheap labor suppliers. Successful companies have integrated their global offices as partners in a common development program. They have successfully added partner networks to this program and made them joint stakeholders. In addition to executing a well-defined strategy, it is important to build and manage effective partnerships, implement best-of-class communication and network infrastructures, and remain competitive with local human resource practices to retain best talent. Exhibit 1 shows cost savings related to the need for effective communication and management for in-house and outsourcing vehicles.

SOFTWARE TEST ROI

What Is ROI?

ROI (return-on-investment) is defined in the *Dictionary of Modern Economics*[2] as:

> A general concept referring to earnings from the investment of capital, where the earnings are expressed as a proportion of the outlay.

The equation for calculating the percentage of ROI is:

$$\text{ROI} = (\text{Net benefits}/\text{Net costs}) * 100$$

Net benefits can be either direct, in terms of incremental revenue generated, productivity gained, or expense saved, or indirect, from the redeployment of resources to tasks that the organization would alternatively have had to hire new and like resources to perform. *Net costs* include recruiting, salaries, benefits, software licensing, and general and administrative overheads.

The ROI for each project is compared with the net costs on a yearly basis. ROI analysis means that where expected benefits and costs are realized within the same year of implementation of the project, the project is more likely to proceed. Sometimes, ROI is not used as a percentage but as a numerical number (Net benefits/Net costs); this is referred to as the ROI ratio. This ratio is equal to 1 (or 100 percent) when net quantifiable benefits equal net quantifiable costs, which represents a break-even condition. Those projects having an ROI less than 100 percent may not be undertaken.

A 400 percent ROI over five years indicates a return of four times the original investment over a five-year period. Break-even analysis is used to indicate how many months after investment the investment is recouped (100 percent). Not all benefits or costs are easily quantifiable. However, the approach to ROI taken in this article seeks to identify and quantify all significant net benefits and costs.

The "Co-Sourced" SQA, Testing, and Verification Model

In-house, Outsource, or Co-source? We believe that SQA and testing are very suitable for *leveraged* outsourced offshore services as these provide easily measurable benefits and quantifiable ROI. Most IT and development organizations find it difficult to have adequate SQA resources and lab infrastructure, and end up shipping substandard products because of time-to-market pressures. In the current outsourced model followed by many companies, not only can these companies easily get highly trained untainted SQA resources, but they can also rent a large SQA lab infrastructure for expanded quality testing at a fraction of the cost of doing it themselves. Leveraged outsourced services include the following, in addition to the basic services of QA and testing.

- Better trained resources
- Better lab infrastructure
- Better quality results
- Faster scaling of resources for increased project load

The model that tends to work best is a "co-sourced" model in which an outsourcing SQA team is not completely replacing the in-house SQA team. Instead, the outsourced SQA team works as an extension of the in-house team with a key SQA manager on the in-house team maintaining full knowledge and control of the SQA tasks and project plan. Also, another model that has been reported to work is the "expert-contributor" model. In this model, also referred to as the mentor-student model, the in-house team maintains a key expert and the outsourced SQA team provides one or more contributors. This model provides a "round-the-clock operation" where contributors work as extensions of the expert with his or her guidance and complete most of the work in an offshore location. The keyword in successful co-sourcing model is *control*.

Whether all the SQA and testing resources are in-house, outsourced, or co-sourced using the definition provided above, a frequently asked question in the

minds of IT executives in a company is: what is the most ideal developer-to-tester ratio?

Developers-to-Tester Ratio for Co-Sourcing Model. A ratio emphasizes the wrong thing to the IT executives. Although it is most natural for IT executives to think in terms of "headcount" for planning purposes, there is no simple formula to determine: "If this project needs this many programmers, you need that many testers." The reasons to get headcount increased or get more budgets are understandable but this mindset sets up the requestor for failure.

If the project has a task backlog, adding testers will not necessarily help clear it.[3,4] The productivity, methods, processes, goals, and tasking issues cannot be addressed by adding warm bodies to testing the project. It is key to understand and flesh out the requirements and design in the beginning and then revisit and update them throughout the project duration.

For those interested in reading more on this topic, the reader is directed to Microsoft and others.[5–8]

As discussed, hard ratios should never be taken at face value. The key to success in any project, regardless of the ratio of developers to testers, is well-defined requirements, APIs, peer reviews, and unit testing early on in the project life cycle.

Cost Savings Comparison and ROI Considerations

In this section, we make certain assumptions about the ratios for illustrating the cost savings comparisons for a pure outsourcing and a co-sourcing scenario. Pure outsourcing, while it offers price advantages in offshoring to a country such as India, Russia, or China, does not work as effectively as co-sourcing, where existing SQA and testing resources are supplemented by offshore teams. As Exhibit 2 suggests, the price point marked with an (X), which is pure outsourcing, while being the cheapest is not as desirable from a successful project outcome point of view as the co-sourcing scenario marked with a (√). The section that follows illustrates the different scenarios with numbers to illustrate this point.

Tangible ROI Savings. This example involves an enterprise software application with significant amounts of user interface and E-commerce back-end work involving database access and multiple threads of business usage. The project duration is one year, with the first major release happening in six months and subsequent releases happening on a quarterly basis. For the sake of simplicity, monthly cost per developer and tester is assumed to be $10,000.00. This cost includes salaries, benefits, and other expenses associated with full-time hire in the European Union or the United States.

This example focuses on the team size of SQA and testing and assumes a fixed development team size of ten developers. A key underlying assumption is that there is a good software engineering practice being followed by development teams, including well-defined APIs, bug tracking systems, and source control systems, as well as well-defined product requirements definitions and functional requirements. Also assumed are offshore team "fully-loaded" costs of $3000 per

EXHIBIT 2 — In-House, Outsourced, and Co-Sourced Cost Comparison

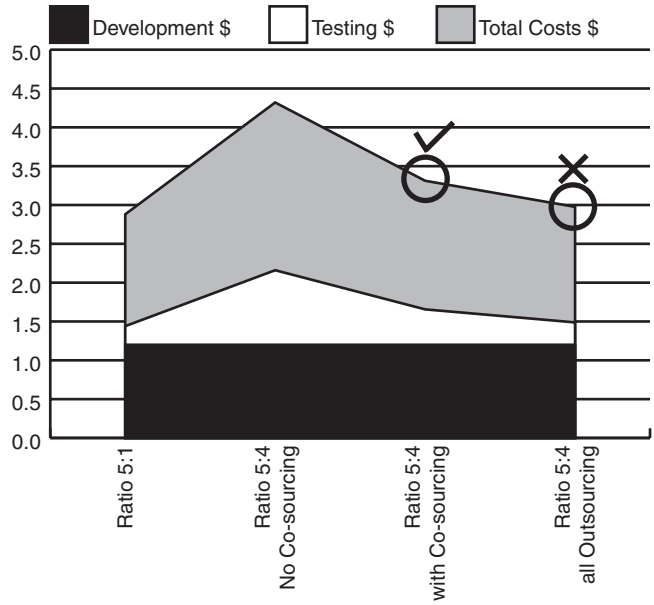


EXHIBIT 3 — Comparison Matrix (all values in $million)

	Ratio 5:1	Ratio 5:4 No Co-Sourcing	Ratio 5:4 with Co-Sourcing	Ratio 5:4 All Outsourcing
Development $	1.2	1.2	1.2	1.2
Testing $	0.24	0.96	0.456	0.288
Total costs $	1.44	2.16	1.656	1.488

All values in $millions.

tester. The fully loaded costs include extra costs such as training and communication. The outsourcer bears the cost of employee benefits and other employment-related expenses for the resource. For the one-year duration, the following numbers result (all values are in $millions):

Ratio 5:4 – No Co-Sourcing for SQA Testing

10 developers @ $10,000 per month for 12 months:	$1.2m
8 testers @ $10,000 per month for 12 months:	$0.96m

Total Cost for 1-year project	$2.16m
Ratio 5:4 – with Co-Sourcing for SQA Testing	
10 developers $10,000 per month for 12 months	$1.2m
2 testers locally @ $10,000 per month for 12 months	$0.24m
6 testers outsourced @ $3,000 per month for 12 months	$0.216m
Total Cost for 1-year project	$1.656m

ROI savings = $504,000

Exhibit 3 is a simplistic model that gives us an ROI savings of $504,000 for a one-year period.

As mentioned, we do not recommend any particular developer-to-tester ratio. In this case study, we are assuming if there are ten developers and two testers, then the testing staff will be overwhelmed. A ratio of 5:4 with no co-sourcing becomes very expensive. On the other hand, if the two testers who started with the project are supplemented by six testers from an offshore company in a co-sourced model, the chances of project success and delivering high-quality products on time are significantly increased. The outsourcing model where all testing staff is outsourced is not very effective because SQA and testing is a communication-oriented activity. With newer trends in software development, including emphasis on Web Services, SQA and testing is becoming more intertwined with early development. Consequently, it is more effective to retain some in-house SQA resources and supplement them with co-sourced offshore SQA and test resources.

Intangible ROI Savings. In addition to the direct benefits of co-sourcing, there are several intangible benefits, including:

- *Proven methodologies.* Established co-sourcing QA and test companies have streamlined operations and test practices. Documentation, test plan templates, methodologies, and processes of a good co-sourcing partner can help QA managers effectively plan, manage, and deliver results.
- *Resource strengths.* Most good co-sourcing QA and test companies have highly trained and enthusiastic engineers who are familiar with a wide gamut of tools and technologies. This provides significant resource strength and the team motivation required for project success.
- *Faster time to market.* Co-sourcing enables 24/7 test and QA cycles and reduces the time required from conceptualization to the final product. Ramping up a large pool of resources becomes possible.

Refining the Cost Savings Model

Further refinements to the cost model presented above are possible. The following outlines some of those refinements. For the purposes of simplification, we have used $10,000 per month or $120,000 per year as the "fully loaded" cost of hiring a tester in the United States or the European Union. Fully loaded includes

the taxes, benefits, insurance, and other human resource costs. Similarly, we used $3000 per month or $36,000 per year as the cost of hiring an offshore tester. These numbers are only approximations. Exhibit 4 illustrates more detailed costs

EXHIBIT 4 — Refining the Cost Model

In-House Cost Refinement	Total per Year	Offshore Costs Refinement
Senior QA engineer yearly salary	$65,000–$85,000	$36,000–$48,000
Training costs: $1,500 per session, three sessions per year	$4,500	$3000
Office space: $3.00/sq. ft./month (150 sq. ft. per person)	$5,400	
Office security cost	$5,000	
Human Resource benefits: 20 percent of salary	$13,000–$17,000	
Operating costs per employee	$17,000	$3000
Initial hiring costs	$1,500	$1,000
Hardware costs	$3,300	
Software costs: basic OS + third-party applications such as virus check, office productivity, Internet applications	$2,800	
HR and IT support	$12,000	
Total	**$129,500–$153,500**	**$43,000–$55,000**

for hiring a tester in a metropolitan center such as the San Francisco Bay Area.

Exhibit 4 compares hiring one employee locally or offshore. It is meant for representative purposes and attempts to capture the details of costing for both in-house and offshore models. Testing requirements, geographical locations, team compositions, vendors, and several other factors change costs and readers are encouraged to use these only as a guideline and analyze their own requirements carefully.

CONCLUSION

Engineering managers and IT leaders must perform QA and testing on today's demanding software systems while keeping costs low. Prevalent data indicates that outsourcing or best-shore QA and testing can help reduce costs while maintaining quality.

The first article in this series, "Maximizing Outsourced Software Quality and ROI" (32-10-90), presented the market trends toward outsourcing and strategies to successfully leverage the cost benefits while improving quality. It introduced

quality practices, the software development life cycle, product readiness, and automated test tools, and it discussed how reducing costs of testing can maximize software ROI by improving quality. This series of articles examined the software development life cycle and criteria for selecting and managing outsourcers. SQA and testing can be very suitable for leveraged outsourced offshore services because it provides easily measurable benefits and quantifiable ROI. Most IT and independent software vendor organizations find it difficult to have adequate SQA resources and lab infrastructure, and end up shipping substandard products because of time-to-market pressures. In the current outsourced model followed by many companies, not only can these companies easily acquire highly trained, untainted SQA resources, but also rent large SQA lab infrastructure for expanded quality testing at a fraction of the cost of doing it themselves.

The second article in this series, "Selecting and Managing an Outsourcer" (32-10-91), discussed strategies to evaluate and execute in companies that the user may be choosing for outsourced development, SQA, test, and verification. It shares the evaluation criteria distilled over several years of outsourcing experience. Specific experiences in dealing with communication, time zone, and cultural differences were presented. Also identified were the background requirements of people who work on SQA and testing as well as the types of testing that outsourcing lends itself best to. Discussions ranged from whether outsourcing QA and testing had been a productive experience for companies, to whether the promise of cost savings was fulfilled. Finally, this second article discussed the models, processes, and QA standards that make the outsourcing experience the most effective for business executives and information systems professionals.

This final article of the series discussed how best to leverage a co-sourced model for maximum profitability while keeping costs in check. It presented models to compare the ROI on in-house, outsourced, and co-sourced SQA teams.

References

1. "IT Migrating to Overseas Outsourcing," by Ed Frauenheim, CNET News.com, February 21, 2003.
2. Source: *The Dictionary of Modern Economics,* 4th edition, 1992, MIT Press, Cambridge, MA, David W. Pearce, Editor.
3. "Managing the Proportion of Testers to (Other) Developers." Pacific Northwest Software Quality Conference, October 2001.
4. "Estimating Tester to Developer Ratios (or Not)." SMTR3, San Jose.
5. Cusamano, Michael A. and Richard W. Selby, *Microsoft Secrets,* Simon and Schuster, New York, 1995.
6. Rothman, Johanna, "It Depends: Deciding on the Correct Ratio of Developers to Testers." 2000.
7. Software Program Manager's Network, Best Practices, 1999, on www.spmn.com.
8. McConnell, Steve, *Rapid Development,* Microsoft Press, Redmond, WA, 1996.

Vijay Sikka is currently the principal at Nirixa, Inc., a two-year old software quality and test outsourcing company in San Jose, California. Sikka has worked in software development and quality for more than 16 years. This includes eight years at Intel Corporation headquarters in Santa Clara, California, developing real-time software quality, analysis, Six Sigma, and statistical process control (SPC) systems for worldwide manufacturing labs. In 1996,

he founded IBrain Software, Inc., and served as its CEO until its acquisition by Entigen Corp in 1998 (later part of Roche). In 1999, Sikka started Bodha.com, Inc., and served on its board of directors until its acquisition by Peregrine Systems (Remedy), a Nasdaq traded company, in 2002. He built international software quality teams and engineering organizations with successful multi-year operations and is an advisor and a consultant to several San Francisco Bay Area companies. Sikka holds a Masters' degree in artificial intelligence from Syracuse University, New York, and dropped out of a Stanford Ph.D. program after four years to start his first company. He is a frequent speaker at conferences on software quality and participates in the W3C Quality Group. He is the author of the forthcoming book *Maximizing Software Quality and ROI* to be published by Auerbach Publications. He can be reached at vsikka@nirixa.com.