

te testing experience

The Magazine for Professional Testers

Outsourcing



什么文化，费用和成果？
What about the culture, costs and results?

How to Build and Maintain a Successful Outsourced, Offshored Testing Partnership

by Dr Mike Bartley

Somebody recently asked me to review a course on “Building a Great Team” - where is the section on outsourcing? I asked. In my opinion, outsourcing should always be a consideration when trying to maximise the performance of a team. In this article I will consider this in the context of software *testing*. I will mainly consider software testing that is both offshored and outsourced¹, although it should also be applicable to software *development* and to companies that just use outsourcing or offshoring. The article is based the successful application of outsourcing at ClearSpeed Technologies plc (which I’ve described elsewhere -see [1]) and in advising other companies on how to outsource their software development and testing.

The objectives for offshoring are evolving. Initially companies looked for cost savings. Then companies looked externally for extra resource (for example during the Y2K crisis). Companies are now also looking for strategic benefits and better defined business impacts from their offshoring relationships. My aim in this article is to demonstrate how such benefits and impacts can be achieved with minimum risk.

To attain the strategic benefits you seek, you must first identify them. Let’s consider the benefits claimed for offshoring by managers of technology SMEs (Small/Medium Enterprises) in [2]:

To accelerate the maturation of new products: “*The internal skills of rapidly developing products that meet market needs is augmented by offshored teams who have the patience, precision and perseverance to evolve young products into mature ones*” (Jajiv Dholakia, Cenzie Inc.)

¹ Note offshoring does not imply outsourcing - a company can establish an overseas function.

Agility and cost savings: “*on average we are able to deliver projects at a much faster rate and save at least 30% cost to our clients*” (Jing Liu, CEO, Enter Suite)

Better software through additional resources: “*The additional talent and resource that we can carry allows us to develop better software than our competition can afford in the same period*” (Scott Allan, Symbol technologies Inc.)

Concentrate on core competence: “*Businesses should focus on clients and IP which satisfies their need. Everything else should be outsourced to those whose core competency is that business*” (Bill Widmer, President and CEO, g8solutions)

Ability to undertake one-off projects: “*we were entering a virtually untested market with a new product. The US team focused on the current roadmap while I was able to develop the new product at a rapid speed and at a low cost*” (Gopan Madathil, Founder, TechCoire)

The strategic goals must also be agreed by a suitable forum. When I started on the offshoring route at ClearSpeed, I (as the Quality Manager) agreed the following prioritised objectives with the CFO and Engineering Manager.

- To improve company focus.
- To improve company resource flexibility and by doing so, reduce time to market.
- Quality improvement.
- Cost variablization².
- Cost reduction.

² This means converting fixed costs to variable costs.

The above goals were shared with the offshore supplier. This gave us a common understanding which helped the day-to-day management of the projects. We also derived metrics to measure our progress against the goals. For example, we used Defect Detection Percentage (DDP) to measure the improvement in our testing (see [3] for an explanation of DDP). The time to market and the cost of testing relative to development were also tracked.

The next major decision regards on your road to offshoring is to decide what type or stage of testing to offshore. There are a number of considerations to take into account.

- **Amount of documentation:** In order to test your software your offshore supplier will need to understand it. This will ultimately require good documentation.
- **Domain knowledge:** How much domain knowledge will the tester need and how easy will it be to find or train testers with that knowledge?
- **Complexity of the environment:** How complex is the test environment and how easy will it be for the offshore test organisation to re-create it?
- **Development process:** Testing within the V model lends itself well to offshoring. Offshoring of agile development less so as it is based on nimble teams with good communication and rapid change.
- **Stability:** How stable is the software that you are asking the supplier to test.
- **Level of interaction:** To what extent will the offshored test team need to interact with the development team.
- **Level of independence:** NASA uses an independent V&V team to improve the

quality of their software (see [7]). An offshore test team allows you independence along three dimensions: managerial, financial and technical.

- **Taking a customer perspective:** An offshore team can be used as a “friendly” customer, helping to identify all the problems that your real customers would otherwise find.
- **IP protection:** This is always a major concern in outsourcing and the concern seems to deepen with offshoring.
- **Sign-off criteria and incoming acceptance:** You will need to sign-off the work performed by your supplier through some sort of incoming Quality Assurance (yes – testing the tester!).

Now consider how the above affect your decision on what to outsource:

- Unit testing:
 - This will probably require you to release source code which may lead to IP protection issues.
 - Unit code may not initially be stable and may not be well documented.
 - The offshore organisation can act as an independent V&V team.
 - The test environment for unit testing is not usually over-complex and the domain knowledge requirements tend to be lower.
 - Sign-off could be defined through structural coverage goals.

As an example, I successfully offshored the unit testing of a new library of memory management functions. The functions were developed using agile techniques in close collaboration with marketing and two lead customers

and, once the specification was stable, we offshored the unit testing. We had an existing, trusted relationship with an offshore partner who first produced a test specification and then tests against that specification. Finally, they topped up their tests to ensure our structural coverage goals were met. Our incoming signoff consisted of signing off the test specification, reviewing a selection of actual tests against that specification and re-running the tests to check the coverage.

- System testing:
 - In my experience by the time you reach system testing the software is more stable and the chances of some documentation are improved.
 - If you are following the V-model of development then the appropriate documentation may have been available for a while so that the offshore organisation could have already written test specifications reviewed by you.
 - The test environment and domain knowledge required can be complex.
 - The level of interaction should be reduced compared to earlier stages of testing.
 - The offshore team can perform the testing independently and can take a customer perspective in their testing.
 - IP protection issues can be avoided by releasing the binaries rather than source.
 - Testing the tester is harder as an objective measure of system test quality is more difficult - but it is not im-

possible. A system test specification and reviews of the test scripts for example. The test results should be also made available for audit.

Some of my most successful offshored testing has been in system testing at ClearSpeed. We had an automated test environment which gave very high unit and integration test coverage, and some system testing. We then offshored all of the system testing that was impossible or not cost-effective to automate. The offshore team was encouraged to act as a customer with no prior assumptions and not to use any work-a-rounds offered by “helpful” members of the development team!

There are numerous other types and phases of testing where the above considerations can be applied but space does not allow for here. For example: **specialist testing** (such as security testing) where the outsourcing arguments are very strong but expertise rather than price will be most important; **compliance testing** (where you are required to demonstrate conformance to a particular standard) where there is a strong argument for outsourcing to specialists in that standard; **release or acceptance testing** has similar arguments to system testing discussed above. And the list goes on...

You also need to identify your preferred outsourcing model. [5] describes three types of service deal related to the scope and complexity of the offshoring contract and your strategic objectives.

Efficiency: These deals tend to focus on cost efficiency, improved resource agility or greater organisational focus on core skills. The contract terms concentrate on service levels. So, for example, you may be outsourcing unit testing currently performed by developers to free them up for additional projects and the service level

Commercial model	Description and discussion of risk	Recipient Risk	Provider Risk
Cost-plus pricing	The recipient pays the providers costs plus a percentage. The provider is guaranteed cost recovery. The recipient is not guaranteed service levels. Appropriate for efficiency deals.	Medium	Low
Fee-for-service pricing	Price is based on the amount and/or quality of the work actually delivered. The provider's costs are not covered unless service levels are met. The recipient costs are not entirely predictable and service levels are not guaranteed. Appropriate for efficiency and enhancement deals.	Medium	Medium
Fixed price	Defines a specific service level and a price for achieving it. This is higher risk for the provider. The recipient has lower risk as prices are capped. Appropriate for efficiency deals.	Low ³	High
Shared-risk/reward pricing	This involves a flat rate with additional payments for achieving specified outcomes. Appropriate for enhancement and transformation deals.	High	High
Business outcome achievement	The provider only receives payment for achievement of specified business outcomes. The receiver has no guarantee that expected outcomes will be achieved. Appropriate for transformation deals.	Medium	High

Table 1: Commercial models, service deals and risk

³ I have been involved in too many fixed-price deals to agree with [5] that this is low risk for the recipient. There is always the risk of changes in requirements which can lead to significant cost increases.

might be around test specification, structural coverage and time to completion.

Enhancement: These deals are more to do with process improvement. For example, you may outsource to a specialist test company to improve your security testing.

Transformation: These deals aim to improve your competitiveness through innovation in the relationship. So, for example your goal may be to reduce time to market or improve the quality of your software (through reduced DDP for example). Your outsource supplier would need to act in a consultancy role as well as a service supplier.

Obviously the above three imply an increasingly complex relationship with increasing levels of trust. [5] goes on to consider the appropriate commercial models for the deal and this is summarised in Table 1.

You also need to consider your outsource location: on-shore, near-shore or offshore. This decision should be made in the context of your chosen service deal. I would also recommend considering multiple suppliers. In my experience this brings a number of advantages.

- Using suppliers with different types of skills.
- Using one supplier for an efficiency deal and another for an enhancement or transformation deal.
- Allowing the suppliers to competitively tender for new work (rather than becoming increasingly dependent on a single supplier).

Now that you understand what you want to offshore and your preferred commercial model, the next consideration is your readiness for offshoring. A number of attempts at offshoring fail because an organisation has had existing internal development practices: *offshoring your chaos will just generate additional chaos!* So, how do you measure your readiness? There are formal measurements such as CMMi (see [4]). Table 2 is taken from [8] and shows the national differences in CMMi take-up. It is also interesting to note from the table the differences in maturity from the countries more likely to be supplier to those more likely to be provider. In my experience providers tend to go for CMMi for a mixture of technical, management and marketing purposes.

Most organisations will not want to gate their offshoring on such formal appraisals. In my experience, the minimum set of processes that your organisation will require to outsource software testing is:

- Requirements management
- Change tracking and management
- Project planning, monitoring and control
- Configuration management
- Build, test and release automation
- Validation and verification processes

You now need to select your partner. There is no magic in the selection process:

1. Identify potential suppliers.
2. Define your selection criteria.
3. Create a short-list.
4. Perform an in-depth appraisal of your short-list (including a visit if possible)

5. Contract negotiation.

6. Perform a pilot project with your selected supplier (or suppliers)

The major issues in the above list are identifying appropriate selection criteria and exchanging contracts. Below are some example selection criteria

- **People:** Skills profile; Academic background of staff; Staff training statistics.

- **Retention:** Staff attrition rates.

- **Process:** Process maturity; CMMi and other process maturity measures attained. You should consider how well matched the relative process maturity between your two organisations

are. For example, do the outsource organisation processes require deliverables and maturity on your side that you cannot deliver on?

- **Quality:** Quality processes employed; how well do they match your quality processes?
- **Project and risk management:** What management processes do they follow? How well do they fit with your preferred management methodology?
- **Company profile:** Financial stability; historic financial data strategic goals for the company.
- **Technology:** Domain knowledge/experience of the technologies employed by your company.
- **Cost:** What are the typical pricing models? How are changes managed from a cost point-of-view? Recent cost change history.

If you are offshoring then your selection criteria will need to reflect circumstances in the providers' country. For example, if the destination is India then staff turnover or retention rate should be a major consideration.

Regarding contractual negotiations, I have found it helpful to have two documents: a Master Services Agreement (MSA) and a Service Level Agreement (SLA). The MSA covers the legal, financial and contractual issues (such as confidentiality, ownership of deliverables, assignment of rights, indemnification, warranties, liability, changes in personnel, subcontracting, termination, penalties for not meeting agreed service and delivery targets, etc.). The Service Level Agreement (SLA) should cover the service and support goals and is strongly affected by the type of service deal that you have (efficiency, enhancement or transformation). The MSA and SLA are complex legal documents which I will not attempt to cover in detail in this article.

You are finally ready to execute your offshored software testing project and this is where it starts to become difficult! If you have followed the process I have described above then your chances of your success are significantly improved. However, any offshored development is inherently risky. First there are the problems of managing a distributed team (communication issues, knowledge management, change management, etc). Then, according to Ralph Kliem in [7], in offshoring there are also "*the unique challenges posed by geographical, cultural, and other differences*". There is not enough space in this article to go into details of how to manage offshored projects but I will highlight the main items to consider. Firstly, you need to distinguish management and governance. Management is about responsibility for specific decisions; Governance covers decision-making processes: the who, the how and the when. It is important to define the governance rules to allow your offshore team to know when they are empowered to make decisions and how to make good decisions.

Country	Total Appraisals	Maturity Level Reported				
		1	2	3	4	5
Argentina	47		31	10	2	3
Australia	29	1	8	4	2	4
Brazil	79		37	31	1	8
Canada	43		10	18	5	3
China	465	1	103	293	18	34
Egypt	27		12	11	2	2
France	112	4	67	34	1	2
Germany	51	7	27	1	1	1
India	323	1	11	127	22	151
Japan	220	16	64	88	13	15
Korea, Republic of	107	1	31	48	11	7
Malaysia	42		15	24		3
Mexico	39	1	18	13	3	4
Spain	75	1	49	21	1	3
Taiwan	88		60	25		2
UK	71	3	36	24	1	2
US	1034	25	365	347	21	114

Table 2: Number of Appraisals and Maturity Levels Reported to the SEI by Country (where total appraisals is above 20)

Good governance involves at least the following

- Established standard decision-making procedures.
- Well-understood processes for handling exceptions.
- Decisions made quickly at the correct level against well-defined criteria.
- Decisions get recorded.

And good governance is helped by the following

- Having agreed strategic objectives.
- Formal communication methods which record project status and decisions.
- Controls & records to ensure governance procedures are followed.
- Regular meeting of the stakeholder group.

Good management means the usual good software project management practices plus

- Communication
 - Who can communicate with whom? And how often?
 - Formal vs. informal communication.
 - How does the communication get recorded?
 - What happens on a miscommunication?
- Knowledge management
 - Getting up the initial learning curve and staying there through staffing changes
 - On-going knowledge management
 - What is the process for asking a question?
 - What is the process for recording an answer?
- Audit trails
 - What trails do you need? Decision making, Execution history, Timesheets?
 - You need to save this data quickly and cheaply but what about access time?
 - How strict does your audit need to be? Are there any legal requirements?
- Well-defined incoming QA procedures

In summary, there are many benefits to offshoring your software testing, especially if you aim to build a competence based rather than transactional outsourcing relationship. However, the risks are also high and in the above I have laid out a process to improve your chances of success.

- Agree the strategic goals in an appropriate forum and share them with your provider.
- Decide what type or stage of testing to offshore.
- Identify your preferred outsourcing model and location.
- Assess your readiness for outsourcing.
- Select your partner or partners.

References

1. "How to Boost your Productivity through Outsourcing", Mike Bartley, Software and Systems Quality Conference, London, Sept 2008
2. "Happy About™ Outsourcing", Mitchell Levy, January 2005
3. "How to measure test effectiveness using DDP (Defect Detection Percentage)", by Dorothy Graham et al, Grove Consultants (available from http://www.dorothygraham.co.uk/downloads/generalPdfs/DDP_Tutorial.pdf)
4. Capability Maturity Model Integration, <http://www.sei.cmu.edu/cmmi>
5. "Multisourcing: Moving Beyond Outsourcing to Achieve Growth and Agility", Linda Cohen and Allie Young, Gartner, Inc.



Biography

The author, Dr Mike Bartley, gained a PhD in Mathematics at Bristol University and an MSc in SW engineering and MBA with the Open University. More recently he has added ISEB Practitioner Testing Qualifications in both Test Management and Test Analysis. Mike has been involved in both software and hardware development for the past 20 years, including outsourcing, and regularly writes articles and delivers conference papers. He has recently established his own consultancy to help companies in product QA and offshoring.

Interview Dr Mike Bartley

Dr Mike Bartley,
PhD in Mathematics at Bristol University and
José Díaz, Editor



1. When did your firm decide to go for an outsourcing vendor?

In May 2006, ahead of my recruitment. It was decided to recruit a Manager with specific responsibility for software testing (and pre-Silicon hardware verification) and to ask that Manager to have as much of the team offshore as possible.

2. What were the business criteria that were considered before coming up with the decision?

There were many, but the main ones were to improve company focus; to improve company resource flexibility and in doing so, reduce time to market; software quality improvement; and finally cost variablization and cost reduction.

3. What were the IT processes you planned to outsource and why?

We were looking to outsource our manual software testing. We had built up an extensive internal automated build and test process suite which allowed us to perform daily build and test. But there were a number of tests that were either too hard or not cost-effective to automate.

4. How did you go about the vendor selection process, and what were the basic criteria for evaluating the vendors?

Well, that is a long story which I cover in my paper elsewhere in the magazine. We were not just looking for a transaction-based outsourcing relationship, we were looking to enhance our software testing and bring real business level benefits. So we looked for “partners” rather than “vendors”, people that we could build a long-term relationship with. But as I said, you need to read my paper to get the details.

5. What were the key expectations from the outsourcing activity?

We mainly wanted to improve time-to-market and software quality – things that would have a real impact on the business. It annoys me when people view software testing purely as a cost centre. Testing can add real value to a business when it is done well – so aim high!

6. Who were the vendors short-listed (nature of vendors/ size)?

There was a lot of variety in the short-list. It included large companies (over 10,000 employees) and small (those with less than 100 staff); specialist testing companies and large general-purpose IT service providers.

7. Why were Development and Testing split? What were the reasons for going for independent testing? How was it beneficial?

When I first arrived, they weren’t split. The main reasons for splitting them were to free up the internal developer team (who were doing the testing) and to improve quality. There is also very strong evidence that separating the development and testing along management, technical and financial boundaries is a major contributor to software quality improvement.

8. Would you recommend that others outsource their software development and/or testing?

I now spend my time helping companies to do exactly that. The potential business benefits are huge and can be had by both large and small, young and old companies. But the risks are high, too. My paper in the magazine describes how to maximize your chances of success.