



Achieving Agility in Testing through Outsourcing

Mike Bartley
Test and Verification Solutions
Bristol, England
mike@tandvsolns.co.uk

Abstract:

This paper looks into how a company can achieve agility in its testing approach through the use of outsourcing. Specifically, it considers how outsourcing can help with the following:

- Agility in resources: outsourcing allows a company to adopt a flexible resource profile during the development lifecycle and across different software development projects.
- Agility in skills: a company can gain access to a range of the specialist skills required for full testing of their software release.
- Agility in quality: giving a company the ability to adapt the amount and depth of the testing it performs prior to release.
- Agility in cost/risk: a company can adapt where they perform their testing and vary the cost of that testing and the level of perceived risk in that outsourcing.
- Agility in schedule: agility in the above give a company agility in schedule. That is, they are better able to adapt the schedule to meet market demands rather than engineering limitations.

The paper will make recommendations about how a company can arrange its outsourced testing services to maximise its testing agility, and thus achieve a competitive advantage.

1.0 Introduction

The theme of Test2008 is “Agility in Testing” and this paper looks at this theme from an outsourcing perspective. The conference theme is intended to allow delegates to explore and expand on the need for testing teams to adapt to the ever changing demands of release and quality. So, this introduction first explores some of those changing demands. We then turn our attention to the theme of outsourcing and agility. It first considers how to establish an effective outsourcing relationship and then how that outsourcing gives agility in various ways. The paper will pay particular attention to the impact of outsourcing on small-medium enterprises (SMEs), so the introduction gives an overview of SMEs. Finally the introduction gives an overview of the contents and the structure of the paper.

PureConferences

www.pureconferences.com

Pure Conference Solution Pvt Ltd., A-108B, Sector 58, NOIDA 201301, India; Tel: +91 120 4621080

Page 1 of 23

1.1 The demands of release

A delayed release can cost you from a number of perspectives: delayed revenue; reduced revenue (as you miss the market window); lost market leadership; brand image.

On the positive side, early to market can lead to significant advantage. [[5]] notes that the general consensus in the economics and strategy literature is that firms that move first and are able to establish a standard have the opportunity to economically benefit from their initiatives. [[6]] finds that firms first to market can develop advantages that can last for decades.

Thus, in general, there is demand for reducing development times and releasing sooner to meet the increased pressure caused by shrinking market windows.

1.2 The demands of quality

Quality is a hard term to define and it is very hard to reach a consensus on its meaning. However, we are considering the changes in demand for quality. So, whatever your organizational definition of quality is, how are the demands on it changing? In my experience, what are required to answer this question are measures of quality so that we can consider the changing demands on those metrics. However, again since quality is such a hard term to define, our measurements are usually indirect. That is, we measure things which are indicative of our definition of quality.

From a software testing perspective, I measure quality through a number of test completion metrics when trying to make a release decision. Some examples are given below.

Coverage: Either quantitative (e.g. structural coverage) or qualitative. The latter is based on the test coverage of the software functionality or non-functional features such as performance as measured by how much of your test cases have been run.

- **The bug rate:** Has the number of defects found dropped? This needs to be assessed together with coverage. For example, it is not much use to have no new defects found for 2 weeks if you only have 10% coverage.
- **Code stability:** Has the code churn rate dropped? That is, is the code stable?
- **Static analysis:** Using free such tools as “lint” or commercial tools such as CoVerity. Have all the issues identified been resolved adequately?
- **Known defects:** Are the known defects acceptable?

Please note that this is mainly a testing perspective on quality. There are other activities and non-testing related measures.

As an industry, poor “quality” is often quoted as a reason for delayed release. The above measures should be used to measure the level of risk being taken in a software release. This allows informed commercially driven risk-benefit release decisions to be taken rather than engineering driven decisions. For example:

- It may be better to release an alpha or beta product knowing that certain parts are of lower quality if it is felt that the benefit of the feedback from the users

outweighs the risk. It usually makes good business sense to then carefully select the alpha or beta customers who equally understand and agree with your risk-benefit analysis.

- It might be determined that overall the software is of sufficient quality except for a particular area which can be “switched off” for the release. For example, I have seen performance optimisations disabled due to quality concerns.

In this paper we will consider quality from just a testing perspective. Then, during a release, what is required is a measure of the confidence in the level of testing for a piece of software to help make a release decision. The changing demand of “quality” is thus firstly to make the software testing more measurable in order to allow a company to make better release decisions. Other quality demands can then be translated through those metrics.

1.3 The relationship between quality and release

Quality and release are not orthogonal.

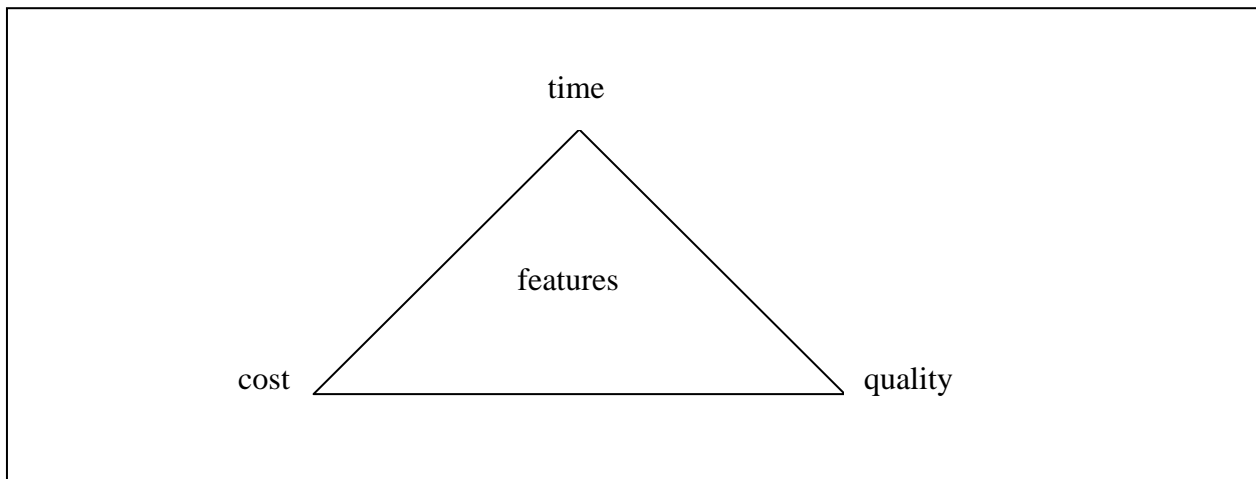


Figure 1: Resources triangle

We can see here that as the feature set changes, we can adapt the time, cost and quality of the software to implement those features.

1.4 The meaning of agility

The meaning of agile in software engineering has possibly changed since the concept of “agile development”. Agile development has its own manifesto, a set of guiding principles (for example “satisfy the customer through early and continuous delivery of valuable software”) and advocates a number of new methods such as extreme programming (see “www.agilemanifesto.org” for more details).

According to the Oxford English Dictionary, the definition of agile is “*having the facility of quick motion*” which I interpret as mainly as meaning the ability to respond quickly. In this paper we will look at agility more from the OED perspective.

1.5 Looking at Small to Medium Enterprises

This paper will pay special attention (but not exclusive attention) to SMEs (Small-Medium Enterprises) where I believe there is potential for outsourcing to have a large beneficial effect. There is no generally accepted definition of an SME. The most widely used definitions tend to be based on size using the number of employees rather than financial or other measures. However, there is no generally accepted definition in terms of the number of employees either.

The EU has attempted to standardize the concept. According to [[7]], the EU categorizes companies with fewer than 50 employees as "small", and those with fewer than 250 as "medium". The same reference also highlights the importance of SMEs to the economy. In the EU, SMEs comprise approximately 99% of all firms and employ between them about 65 million people. In many sectors, SMEs are also responsible for driving innovation and competition. These EU numbers are reflected globally with SMEs accounting for 99% of business numbers and 40% to 50% of GDP.

1.6 An overview of this paper

As the title suggest, this paper looks at how a company can become more agile in its software testing through the use of outsourcing. However, the paper also considers how best to arrange that outsourcing to maximise its impact on a companies agility. The paper will make recommendations about how a company can arrange its outsourced testing services to maximise its testing agility, and thus achieve a competitive advantage.

The paper starts with a look at outsourcing in general and outsourcing of testing services in particular. The paper then goes on to look at how outsourcing can help a company achieve agility in a number of areas: resources; skills; quality; cost/risk; schedule. We then look at how we actually go about realizing that potential agility when we enter into an outsourcing relationship. Finally, the paper makes recommendations about how a company can improve its agility through outsourcing.

2 Outsourcing as part of an overall test strategy

Your software development and testing strategy should drive your outsourcing strategy rather than vice-versa. In this section we will consider two typical but very different development models and associated test strategies and how a software test outsourcing might fit within those.

2.1 The V development model

“Figure 2: Software development V-model” shows a typical software development and testing model.

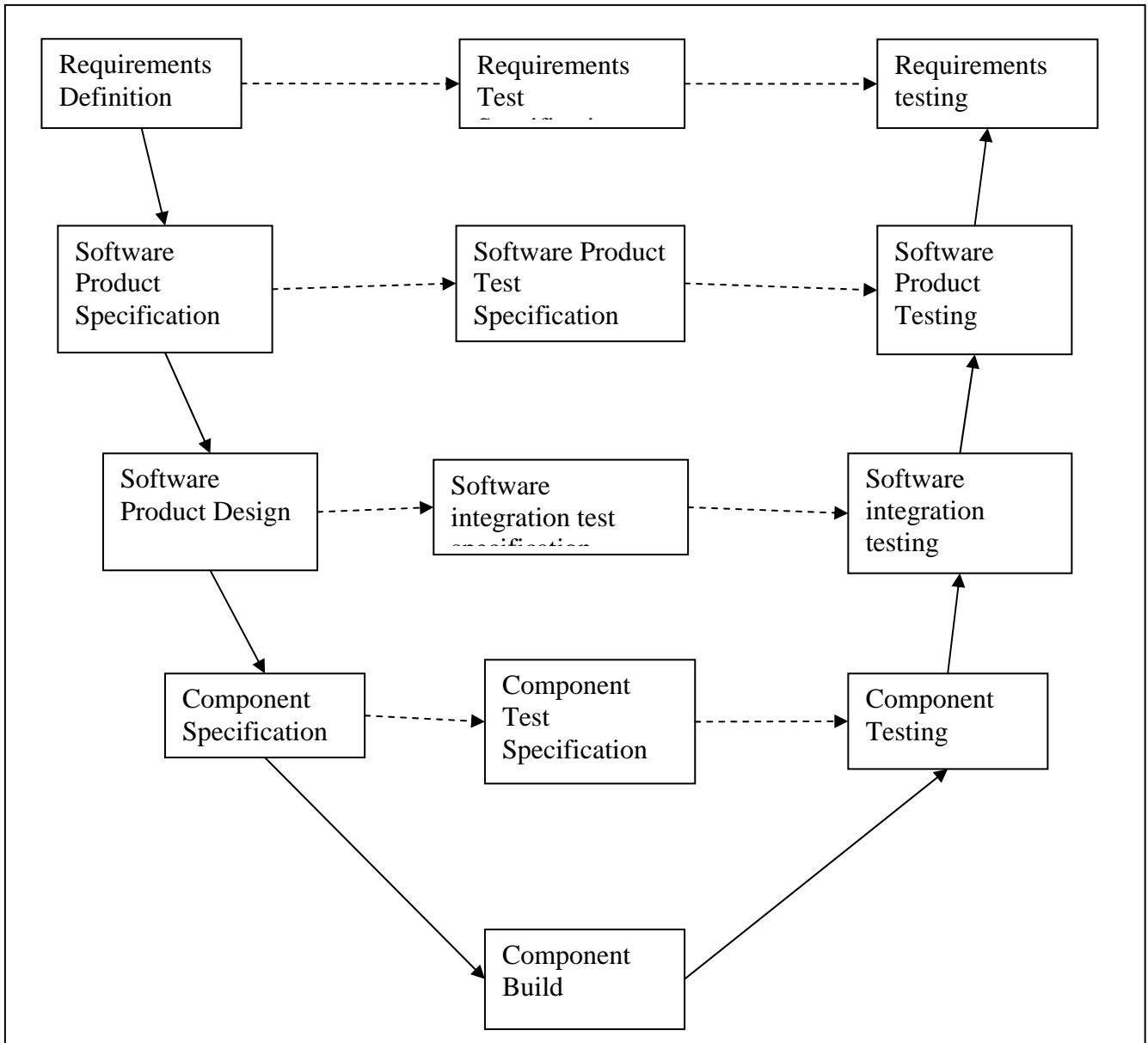


Figure 2: Software development V-model

Looking at the V-model is not the complete story on your test strategy but does give a very good framework for considering your outsource test strategy. For example:

Component level testing requires close interaction with the developers but bugs found at this level are cheaper to fix than if they left to later in testing.

Components are often less well specified and undergo more change than higher levels of the V.

In the experience of the author

- The higher levels of the V tend to be better specified and thus better suited to outsourcing.
- The higher levels of testing are harder to automate and thus require higher manual effort and are therefore more appropriate for outsourcing.
- An outsource organisation are better positioned to imitate a customer at the higher levels of testing.

The traditional V-model approach to software development is formalised and attempts to control the development process in a defined and rigid manner. For example, it requires a defined set of requirements up-front, good documentation throughout the process and a well defined change control process. Many offshore technology companies have turned to ISO and CMM certification as a way to increase confidence that they can handle such a formalised approach.

However, the formalised approach of the V-model development model is not always considered the best or most appropriate to develop software. Organisations or projects move to more flexible development models to overcome these issues – “agile” being the umbrella term for such development models.

2.2 Agile development methods

There is much debate as to whether agile software development can be outsourced. The quality of communication both within the team and between the team and outside stakeholders is very important factor in agile projects. [[10]] assesses the quality of communication in terms the following measures:

- Bandwidth
- Latency
- Reliability
- Availability

Of these measures, we want latency to be low and the others to be high. The paper goes onto argue that the measures are good for the XP team but bad for an outsourced team that's located in another continent.

Another issue with agile development methods is the contractual issues. Fixed price projects suit prescriptive methods such as the V-model but are probably unsuitable for agile development methods. Having a team is probably more suitable and [[11]] examines this issue in detail. Responsiveness to changes encourages using pricing models like time and materials, which gives flexibility both to client and provider and is a better fit for agile development. Large organisations can use economies of scale to establish their own offshore development centres but this doesn't make so much sense to a small-to-medium enterprise and their best route is often to work through an offshore service provider. [[11]] also identifies communication as a major issue but believes these can be overcome. ABN AMRO adapted agile to offshoring by using a two week development

iteration. The reason given in [[12]] is that daily communication added too much complexity and was not adding value to the development efforts. [[12]] says that ABN AMRO only outsource non-core projects but this frees up key in-house developers for the most important or sensitive projects.

Test driven development can also be used with outsourcing. In [[13]] IBM describe how they used test driven development for unit test development but still retained an independent “functional verification test” (FVT) team who developed black-box tests based on a requirements document and conversations with developers. I believe that it is possible to outsource the FVT. Indeed I have used a similar approach and this is described in the case study in section ???.

3 Effective outsourcing

In this section we consider the elements that need to be considered in order to outsource effectively.

- Define the strategic goals behind your decision to outsource.
- Decide what to outsource.
- Review your readiness to outsource.
- Select your preferred outsource model
- Select your outsource partner.
- Agree the Master Services Agreement and Service level Agreement.
- Outsource execution: Management and Governance.

3.1 Defined strategic goals

It is important that the strategic goals for the outsourcing are both defined in advance and agreed by the appropriate stakeholders. Some typical strategic goals might be

- Getting access to specialist skills.
- Improve your company focus.
- Improve company resource flexibility and by doing so, reduce time to market.
- Quality improvement.
- Getting access to higher quality staff.
- Cost variablization.
- Cost reduction.

These goals need to be agreed by the appropriate stakeholders. For example: the software manager; the testing manager; the engineering manager; the chief financial officer; the outsourcing manager. This stakeholder group should continue to meet after the strategic goals have been defined to ensure the goals are being met.

It is recognized that outsourcing could deliver benefits above and beyond those defined in your strategic goals. However, the important points to note here are:



- The strategic goals are agreed in advance.
- By all of the appropriate stakeholders.

3.2 What to outsource?

The next consideration is to decide which activities should be outsourced. This decision is driven by the strategic goals but there are a number of issues to consider.

- Intellectual property issues: How much information are you prepared to divulge? Do you need to deliver source code or just the executables?
- Proximity to product development: Unit level testing requires closer proximity to the development than does system or acceptance testing.
- Test development vs. test execution: Outsourcing execution requires less knowledge about the product than test development.

3.3 Readiness for outsourcing

Outsourcing a chaotic internal process only breeds further chaos. So the next step in your outsourcing strategy is to audit the activities you wish to outsource to determine how ready and how suitable they are to make the change. There are some formal ways to measure this.

The Capability Maturity Model: Originally developed during the late 1980s at Carnegie Mellon University in Pittsburgh, the Capability Maturity Model (CMM) was designed as a tool to use in assessing whether government contractors had the right processes to execute a contracted software project. Subsequently, it has been deployed more widely as a means for measuring the process capability maturity of organizations in diverse areas (more information on its use is available online at www.sei.cmu.edu/cmml).

If, after running the relevant CMM-framed assessment, you find you are at Level 1 (Initial - “Processes are usually not documented and change based on the user or event”), then you are unlikely to succeed at outsourcing. Simply sharing your chaos with other organisations is unlikely to improve the situation. CMM Level 2 is “Repeatable” - “The software development processes are *sometimes* repeatable, possibly with consistent results and may not repeat for all the projects in the organization”. Level 3 is “Defined” - your processes can be considered repeatable.

The Test Maturity Model Integration:

The Test Maturity Model Integration (TMMi) has been developed to compliment the CMMI framework, providing a structured presentation of maturity levels, specifically for test process measurement method (more information at www.tmmifoundation.org).

You may regard these models as too formal. However, you should certainly consider how mature the following processes are regarding measuring your readiness for outsourcing: requirements management; change tracking and management; project planning,

monitoring and control; configuration management; build, test and release automation; validation and verification processes.

3.4 Outsource model selection

The commercial model employed obviously impacts the balance between cost and risk. [[1]] identifies five basic pricing frameworks each with a different risk profile for the service recipient and provider.

Cost-plus pricing: The recipient pays for the services at the suppliers cost plus a set percentage. This is medium risk unless you can negotiate guaranteed service levels.

Fee-for-service pricing (or T&M): The recipient pays based on the amount or quality of the service provided. This is medium risk as costs are not entirely predictable and service levels are not guaranteed.

Fixed price: This defines a specific deliverable or service level at a fixed price. **[Error! Reference source not found.]** describes this as low risk as costs are capped. However, there may be little flexibility unless you can negotiate this into the contract. Also, any efficiency benefits achieved by the supplier are not passed on.

Shared-risk/reward pricing: This involves a flat rate with additional payments based on achieving specified outcomes. This is medium risk as costs are not entirely predictable and service levels are not guaranteed.

Business outcome achievement: Under this framework the recipient only pays if specified business outcomes are achieved. This is medium risk as the desired outcomes may not be achieved.

The prices changed in the above should reflect the risk to the provider. We can see therefore that there is a spectrum of cost and risk and the recipient has the flexibility to adopt an appropriate balance which is not usually open to them without outsourcing.

You also need to consider your preferred location model.

- Onsite.
- Near-shore.
- Off-shore.

At a very basic level you are trying to get the appropriate balance of cost and risk. You also need to decide which country to outsource to. Your strategic goals should guide you through the model selection.

3.5 Outsource partner selection

The first step is to define the partner selection criteria. Some example criteria might be:

- **People:** Skills profile; Academic background of staff; Staff training statistics.
- **Retention:** Staff attrition rates.
- **Process:** Process maturity; CMM and other process maturity measures attained. You should consider how well matched the relative process maturity between

- your two organisations are. For example, does 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 and 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.

Your particular choice of criteria should be driven by your strategic goals. Armed with your selection criteria, your next step is to identify and assess your potential suppliers. You should now be able to follow a straight forward selection process. For example,

- Identify potential partners.
- Assess potential partners against selection criteria to generate a short-list.
- Visit short-listed potential partners to undertake a more detailed analysis.
- Have potential partners undertake a pilot project.

3.6 Master Service Agreement and Service level agreements

[[14]] found that 83 percent of companies surveyed had achieved an ROI of more than 25 percent on their outsourcing projects. However, 49 percent of the executives surveyed said they would have defined service levels that aligned better with their companies' business goals if they could re-start their outsourcing projects.

The Master Services Agreement (MSA) should cover the legal, financial and contractual issues. There are a number of these to be agreed including 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. It will often cover access to documents and people and the handling of change requests as these often have financial implications. The MSA is a complex legal document which we will not attempt to cover in this paper.

The Service Level Agreement (SLA) should cover the service and support goals. For example, the acceptance criteria for both responsiveness and the quality of deliverables, providing performance targets for those objectives and how to handle exceptions. The SLA should define the metrics by which achievement of the targets will be measured and the remedies the targets be missed. Such targets are easier to define for call centres or hosting services than they are for software development. For software development and

testing, the content of the SLA will be different dependent on the pricing model you employ.

- **Fixed-price projects:** The author has found it easier to cover the penalty clauses within an overall MSA and then define a statement of work (SOW) for each fixed-price project. The SOW defines the deliverables from both side, the acceptance criteria for the deliverables and the delivery dates.
- **Time-and-material projects:** As stated above, such projects are usually entered into because they offer flexibility. However, that flexibility makes it harder to define the targets and metrics required to measure the success or otherwise of the project.

When working with this type of relationship, the author found it easiest to maintain a planned list of work and deliverables which is regularly updated through mutual agreement. A record of delivery against the targets was also maintained which gave a historical record of achievement and served as a good basis for reviewing. The main purpose of those reviews was to identify causes for concern and suitable remedies.

A number of research papers also point to the benefits of share success. Setting up and MSA and SLA in this way is complex and the targets need to be very carefully defined to ensure that you are really encouraging the behaviour that you require. Also, it can be difficult to separate the contribution of the outsource organisation from your own organisation. For example, your target might be to release your software on time but this may not motivate your external supplier if the past reasons for missing the target are problems on your side. A better target may be to reward timely completion of their targets or to reward the ultimate quality of the software through measurement of the bugs found in the field. Again the latter may not give a sufficient level of independence if the defects discovered by customers should have been found by the internal tests rather than the outsourced testing.

The author also recommends setting behavioral expectations as well as delivery expectations. For example, in testing it is very important for the test team to retain a level of independence from the development team. Thus, when the outsource team discovers what they consider to be ambiguous documentation then they should not use prior knowledge or clarification from the development team to resolve the ambiguity. It can be a difficult balance to ensure that the outsource team understands the product and has a good working relationship with the development team, yet retains a suitable level of independence and does submit defect reports for all the issues they identify.

3.7 Execution: Management and Governance

The management of offshore projects requires all of the standard multi-site project management practices. These were considered in section 3.3 “Readiness for outsourcing”: requirements management; change tracking and management; project planning, monitoring and control; configuration management; build, test and release processes; validation and verification processes.

However, executing an outsource project and on-going relationship requires additional considerations which [[1]] puts under the heading of “governance”. Whereas management is about responsibility for making and implementing specific decisions, governance is about how and where decisions get made, and who gets to make them. Good governance involves at least the following:

- Established standard decision-making procedures and well-understood processes for handling exceptions within the outsourcing relationships. This allows decisions to be made quickly and at the correct level against well-defined criteria. These decisions need to be recorded.
- The author found that having agreed strategic objectives for the outsourcing and regular reviews with the stakeholders helpful here because it meant that decision making responsibility could be delegated to the level where it made most sense. It meant that guidance and approval was not continuously required or sought. However, the stakeholders had a regular chance to review such decisions.
- Formal communication methods which not only record project status, but also record decisions.
- A system of controls and records which ensure the governance procedures are followed.
- Regular meeting of the stakeholder group to ensure that the governance procedures are being followed and to review progress against the strategic goals. The availability of metrics to track progress against those strategic goals will help.

3.8 Looking at Small to Medium Enterprises and Multi-sourcing

Outsourcing offers particular advantages to SMEs. These are nicely summarised in [[8]]: SMEs that utilize offshore outsourcing find that they can complete projects that are not possible locally due to prohibitive cost or unavailable resources. This offshore leverage can help SMEs go to market faster with new products and services. Additionally, flexible labor pools offshore allow SMEs to complete one-time projects without having to ramp-up internally. Shipping these one-time projects offshore enables SMEs to focus more on their core business units.

Finally, turning our attention to multi-sourcing, according to [[1]], multi-sourcing is defined as

The disciplined provisioning and blending of business and IT services from the optimal set of internal and external providers in the pursuit of business goals.

In the experience of the author, it is multi-sourcing which gives an organisation its greatest agility. Large outsource suppliers may have expertise in all the areas required. However, not all companies want to outsource to a large organisation for a number of reasons. The author sees the following as the main reasons for turning to multi-sourcing:

- **Cost:** Having multiple suppliers for your outsourcing needs adds competition to the tendering process which usually results in lower prices.

- **Expertise:** Smaller niche organisations offering expertise in a particular area often have greater expertise than a large organisation.

The theme of multi-sourcing will be considered throughout this paper.

4 Agility in testing

In this section we consider a number of aspects of testing where agility is helpful. For each we look at why agility is useful or required, and how outsourcing helps to achieve that agility.

4.1 Agility in resources

4.1.1 The resourcing challenge

If we consider the traditional V-model development then the typical resource profile required on a software development project is far from flat and different types of resources are required at different stages. For testing, purposes the skill and resource profile may look as follows:

- Initially a test expert is required to define infrastructure considerations: the test strategy, the test plan, the test environment etc.
- As the requirements definition and software specification become available, the acceptance and system testing plans can be written by a test designer. It could also enable the development of the test environments required for those activities to begin.
- As the software design and module specifications are created, lower level test design can occur. The lower level test environment can also be developed.
- As the code appears, unit level and integration tests can be written against the test specifications.
- We then move to implement the higher levels of system and acceptance levels tests.
- Finally we move into a signoff stage.

As we can see, the skills and resource profile changes over the project. A company with multiple projects may be able to smooth out the resource humps but that often adds undesirable dependency risk between projects. An SME is unlikely to have this option and consequently resources often perform multiple disciplinary roles in order to cover the range of resource required across the project. Also, this type of planning is driven by resource availability rather than being driven by the market or customer needs.

The resource profiles we have considered above are the typical resource profiles for typical projects. However there are a number of unexpected events that disrupt the best resource plans. For example

- Unexpected feature request from a customer.
- Bugs in the field requiring a re-release of the software.
- Requests to support for new platforms.

These disrupt the best planned projects and require resource agility to solve them.

4.1.2 The outsourcing solution

Agility in resources is probably the most obvious consequence of outsourcing. By definition outsourcing is providing access to additional external resource. However, that simplistic argument misses the point that the money spent on buying external resource could have been spent on hiring additional internal resource. So, we need to understand if outsourcing provides additional agility not offered by using additional internal resource. I look at this from the following perspectives:

- How quickly can I turn the resource on and off?

Note that this question does also address the problem of reducing the resource once a task is complete. Treating human resource in this way has much wider implications such as: motivation; continuity; employment law. Turning off resource with outsource suppliers depends on your contract but should certainly be easier and less expensive than turning off internal resource.

Outsource suppliers quite often have techniques in order to allow their customers to have access to resource as-and-when it is required. For example, the use of buffer resources (additional uncharged staff) on a project to allow them to:

Learn about the project.

Stand in for charged staff during vacation/illness (this benefits both sides – the customer gets their expected resource profile, the supplier gets their expected revenue profile)

Become a charged resource if the customer requests additional resource.

The use of buffer resources is usually a commercial cost-benefit issue for the supplier and often is only applied in lower cost offshore destinations. It does have the benefit of providing both a smoother revenue stream to the supplier and a faster route to increase revenue should the customer require it.

- How good am I at keeping that resource?

This refers to the issue of attrition. One of the major issues associated with offshoring is attrition. Different locations have different attrition rates and these should be considered in your outsourcing strategy.

How much resource can I get for my money?

Whilst cost is obviously important, the level of importance depends on your strategic goals for outsourcing. If resource agility is your primary goal then this question should be secondary. We will investigate this balance further in section 5. However, it should be noted that whether outsourcing provides you with more resource for your money depends on:

Your internal resource cost compared with your outsource resource cost:



This will be affected by your outsource location. You may also have an offshore development centre located in a lower cost region.

Your outsource commercial model:

For example, is it time-and-materials or fixed price.

I have been involved in outsourcing through offshoring with different companies over a number of different projects. In my experience when just the software testing is outsourced rather than the whole software development project the companies are much more comfortable outsourcing the latter stages of the testing (system and acceptance testing) or the specialist testing activities. The anecdotal evidence suggests the following rationale for the first of these

Companies often feel that unit testing requires close collaboration with the developer.

Companies feel that they want a tight feedback loop at the unit test stage.

The outsourcer acts as a “tame customer” who are able to interact with the software as a real customer might. I find that this is particularly useful for general product developments rather than specific bespoke developments.

For the earlier stages of software testing companies much prefer to outsource onsite.

For the specialist testing, the rationale is access to specialist skills that you cannot justify as having in-house.

Multi-sourcing can help here in that it may give you the option for resource flexibility with more than one supplier. Those suppliers will tend to compete to provide that resource. Indeed, the more proactive suppliers might have already taken action to make it easier for them to supply additional resource (e.g. using buffer resources as described above). Multi-sourcing also increases the chance that one of your suppliers will have the type of resource you actually require.

In my experience outsourcing has provided me with greater resource flexibility. However, this does not come about by chance or through what [[1]] calls “compulsive outsourcing”. One must take a strategic view to outsourcing in order to truly achieve resource flexibility. Here are some of the strategic decisions

Internal infrastructure decisions must consider the general availability or popularity of the tools selected. For example, on one project I managed we were able to quickly outsource some work at short notice because we were using standard test bench technology.

Outsource supplier selection must consider how well matched the supplier skill base matches the types of skills you are likely to require and the depth of that skill base.

Add the flexibility you require into your contract. For example, how much notice you need to give to turn off resource.

Setup the resource accordingly. For example, a few guidelines I usually try to follow are:

Team-based resource is better for performing regular product release testing or regression testing. A good service level agreement needs to be established so that both parties are in agreement on their expectations.

Testing new well-defined features being added to an existing software project is best undertaken using fixed price projects.

Time and materials (T&M) resourcing should be used very sparingly. It may be appropriate for small testing projects where the requirements are not well understood and a lot of change is expected. However, it is usually best to try to break such projects down into distinct phases:

A short investigation phase under T&M which attempts to better define the second phase

A longer second fixed price phase.

4.2 Agility in skills

To see the variety of testing skills required, let's list just some of the types of testing that are typically required on a software development project: *unit testing; integration testing; system testing; acceptance testing; functional testing; regression testing; load testing; stress testing; performance testing; interface testing; usability testing; install/uninstall testing; recovery testing; security testing; compatibility testing; conformance testing; alpha testing; beta testing; GA testing.*

Some of these types of testing overlap in their skills requirements, but some require specialist knowledge and skills. It is extremely hard for a company to have skills in all of these areas. Large companies may have specialist departments for some of the skills (e.g. a GUI testing department with specialist people and tools, or a security testing department). Again, we come back to the project dependency risk mentioned before.

Typically, a company does not require all of these skills all of the time. In the same way that the resource profile changes over a testing project, so also do the required skills. So, recruiting people with the appropriate skills is not very agile. There is also a question of how that resource keeps their skills up-to-date. An outsource supplier can arbitrage across a number of companies. They can also position themselves as a niche supplier in a particular skill set and thus assign resources specifically to staying up-to-date in that specialism or in developing specialist tools and techniques.

However, it is not just skills in testing that are required when testing software. Often specific domain knowledge and skills are also important. Consider the following real examples:

Company Z has built a basic silicon chip and SDK¹. They want to be able to build demonstrator applications for that basic product such as a software implementation of Ethernet for example. They have considered outsourcing both the development and testing of that application as Ethernet was not part of their core competence. Given that development requires domain knowledge of their tool the outsourcing of development is more of a strategic decision. However, outsourcing of the testing is a much easier decision:

The testing can be performed as a signoff against the standard.

Independence in that testing is a distinct advantage.

¹ Software Development Kit

There is specific domain knowledge about protocol testing in general and Ethernet protocol testing in particular that Company Z does not believe they need to understand.

Company X has expertise in accelerating math computing on a server. They see an opportunity to accelerate an application with a high math content. They implement a thin layer to connect the application to their acceleration technology but look to outsource the testing to a company with domain expertise in that application to ensure the application still works correctly and to measure the performance increase customers can expect.

Multi-sourcing becomes a big win when considering agility in skills. Outsource suppliers will typically have areas of specialist knowledge and so one needs to blend those skills and services into an optimal delivery set. I work with a number of companies who specifically want to multi-source specifically to access a range wide of specialist skills from different suppliers.

4.3 Agility in cost/risk

One of the major differences between internal and outsourced resource is the way it appears on the balance sheet. Outsourcing allows for the conversion of "Fixed" costs to "Variable" costs. [[3]] says “that variabilisation is one of the main motives to use external services. Many firms will externalize in particular those activities that meet a fluctuating internal demand. The cost advantage then flows from scale and time advantage”. In other words, cost savings can be achieved by outsourcing those activities where internal demand is variable as it allows resource (and hence cost) to be varied accordingly. Cost can also be tailored to business drivers other than internal demand such as revenue for example. Obviously tailoring your software testing effort according to revenue may not make complete engineering sense. However, having your costs completely driven by internal engineering demand does not make business sense. The answer, as always, is a compromise between the two. If there is some level of outsourcing then this gives some degree of flexibility in striking that balance.

Outsourcing also allows a company to adapt where they perform their testing and thus vary the cost of that testing against the level of perceived risk in that outsourcing. For example:

On-site is generally regarded as lowest risk but highest cost

Offshoring to a low cost economy is generally regarded as a higher risk but lower cost.

And there are obviously other options between these two extremes.

We saw earlier how [[1]] identified five basic pricing frameworks each with a different risk profile for the service recipient and provider: Cost-plus pricing; Fee-for-service pricing (or T&M); Fixed price; Shared-risk/reward pricing; Business outcome achievement. The prices changed in the above should reflect the risk to the provider. We can see therefore that there is a spectrum of cost and risk and the recipient has the flexibility to adopt an appropriate balance which is not usually open to them without outsourcing.

4.4 Agility in quality

As discussed earlier, quality has a number of definitions and means different things to different people. However, we do not need to get hung up on the definition of quality here. Agility in quality means “having the facility of quick motion” to adapt the level of quality according to your definition. Earlier in the paper we discussed that quality can be defined indirectly through metrics (such as coverage; bug rate; code stability; static analysis; known defects) which can be used to judge confidence in the quality of software when trying to make a release decision.

Outsourcing will not directly affect those metrics but it can give us agility in how we undertake the activities that affect those metrics. So let us look at just a few of the activities associated with software quality.

Specifications: There is strong evidence that writing and reviewing specifications has a huge impact on the quality of the final product. In my experience (although my evidence is only anecdotal) having a customer for your specification firstly improves the chances of it being written and secondly improves the chances of it being adequately reviewed and used.

Bugs which are found earlier are much cheaper to fix. Hence the need to write specifications and the fact that they are thoroughly reviewed has both a large impact on quality and cost. This can also affect the shape of the bug rate improving that quality metric.

Independent Verification and Validation (IV&V): This is the verification and validation performed by an organization that is technically, managerially, and financially independent. According to [[1]] IV&V plays a role in the overall NASA software risk mitigation strategy applied throughout the life cycle, to improve the safety and quality of software. IV&V thus plays a vital role in software quality. By outsourcing the software testing to a group different to the development group then you automatically get the independence that NASA requires.

Code inspections: Code inspections can have a huge impact on quality but rarely occur. The biggest reasons appear to be the lack of time, resource and knowledge/skills. Technology now allows for remote code inspections and there are outsource suppliers that offer code inspections as a service.

Testing: it is often said that you cannot test quality into a product. Testing can only demonstrate the absence of quality. In other words, testing is a quality control activity rather than a quality assurance activity. However, it does make a very important contribution to our confidence at our release decision. It can also

The above list of activities is by no means exhaustive and is not intended to be so. As stated above, the definition of software quality is still debated and so is the means to achieving it. What is important to this paper is the agility an organisation has in performing those activities required to achieve their perception of software quality.

A lot of what is discussed above is not directly influenced by outsourcing. For example, just having a separate software testing team on a project would achieve much of what is

discussed above. However, the agility in resources and skills achievable through outsourcing (as discussed above) does impact agility to perform the tasks associated with improved quality. For example, outsourcing will give a company the ability to adapt the amount and depth of the testing it performs prior to release.

4.5 Agility in schedule

By agility in schedule I include both the agility to speed and slow a schedule. Although the latter is not often required it can sometimes be useful. For example, when working on a new product, if the market window is later than initially expected then it may be advantageous to slow the schedule. Of course, schedule delay can be achieved with internal resource by deploying the resource elsewhere. However, the main purpose of delaying is usually to would be delay the cost which is more easily achieved by delaying outsourced resources. In both cases, minimizing ramp-up is a concern when re-engaging the resource.

The agility in resources, skills, costs and risks discussed above should lead to agility in schedule. That is, through outsourcing a testing manager should be more able to adapt the schedule than they would be if they were limited by the size and cost of an internal team and skill set. Some agility can be achieved if the size and skill-set of that internal team is variable but this may not always be possible and is often more likely in large companies through internal specialist groups or by use of large testing teams across multiple projects.

5 Agility in balancing the 5 dimensions

Project and programme managers often look for flexibility in their plans. The “PACE” product development process described in [[4]] recognises three different variables that can be adapted during a product development. These are scope, schedule and resource. Table 1 “A typical PACE flexibility matrix” shows a how we consider the flexibility that can be applied to the three variables over the product development.

| | Most Flexible | Moderately Flexible | Least Flexible (Fixed) |
|-------|---------------|---------------------|------------------------|
| Scope | ✓ | | |

| | | | |
|-----------|--|---|---|
| Schedule | | | ✓ |
| Resources | | ✓ | |

Table 1: A typical PACE flexibility matrix

Table 1 can be interpreted as follows:

Schedule is non-negotiable. The product must be released on time.

Scope is the most flexible. It might be that features can be deferred to a subsequent release.

Resources can be re-negotiated once the minimum feature set has been reached or when the extra cost justifies the market value of the feature being saved.

I have often used a fourth variable – quality. That is we can also flex the desired quality of the product by performing more or less QA (Quality Assurance) and QC (Quality Control) activities.

Agility in scope does not require outsourcing as it is an internal decisions (albeit driven by market demands) The preceding sections have analysed the additional agility that outsourcing can provide in all of the other variables. I would put agility in skills under resources and agility in cost/risk as the agility to balance between resource and quality. And this is the important point – ultimately we want the agility to strike the appropriate balance between the 4 variables. Let us consider some case studies.

Company X must make a new SW release for a conference that demonstrates some new features to allow them to launch into a new market. The schedule is fixed (you cannot move the conference date). Scope is moderately flexible as the new features are prioritised and so some could be dropped although this is not desirable. Resource is more flexible although this is an SME and does not have additional resource or deep pockets. Quality is most flexible as they can smooth over glitches at the demo stand and reach the required release level quality post-conference.

The SME can use outsourcing to add the additional resource they require for the project. They can then the location of that outsourcing resource to adjust the level of cost/risk they are comfortable with.

Obviously, they cannot move immediately to outsourcing as setting up an outsourcing relationship can take time. Using an outsourcing consultant can accelerate this process.

Company Y is developing bespoke software for a company where quality is of primary concern. It is a fixed price project but the bid has sufficient margin that costs can rise by 20% and Y will still make sufficient profit. Schedule is important but below quality. Scope is fixed.

Obviously resource is the most variable control here. Bringing in resource to ensure sufficient quality could be performed using on-site staff. But again, if the company has an external outsourcing relationship then this could be used to add additional resource at lower cost thus preserving more of the margin.



6 Conclusions and recommendations

The theme of Test2008 is “Agility in Testing” and this paper has looked at this theme from an outsourcing perspective. The conference theme is intended to allow delegates to explore and expand on the need for testing teams to adapt to the ever changing demands of release and quality. The author sees the most significant changes in demand as being

The demands for release increase continue to increase due to shortening market windows and in order to capture first mover advantages.

The demand for quality is to make it more measurable and controlled. As a software testing manager I am continuously trying to measure the confidence in a software release so that the business can make a quantified cost/risk analysis at release time.

The paper then considered outsourcing. In order to establish an efficient and effective outsourcing ability the paper argues that you first need to establish agreed strategic goals. This gives you a solid foundation on which to build good governance and enable agile management of your outsourcing service. We discussed four key models which impact the agility in your outsourcing relationship: location, commercial model, management model and service level agreements. Many of the choices for these models are about balancing cost and risk. However, we also considered how these affect the agility of your outsourcing relationship.

Finally, we considered five key aspects of testing where agility is important: resources; skills; quality; cost/risk; schedule. Agility in these areas allows commercial advantage by allowing planning to align with market and customer requirements rather engineering resource.

So, returning to the conference theme. How does outsourcing help? I would argue that increased agility in resources, skills, quality, cost/risk and schedule firstly gives the flexibility to adapt to shortening market windows. And secondly, it gives the agility to adapt our testing effort to meet the particular quality requirements for a project. In other words, a well-managed outsourcing capability will deliver agility in testing giving significant commercial advantages for the organisation.

References

- [1] Linda Cohen and Allie Young, Gartner, Inc. “Multisourcing: Moving Beyond Outsourcing to Achieve Growth and Agility”
- [2] http://www.hq.nasa.gov/office/codeq/software/umbrella_defs.htm
- [3] Pieter P. Tordoir. “The Professional Knowledge Economy: The Management and Integration of Professional Services in Business Organizations” KLUWER ACADEMIC PUBLISHERS. ISBN:0792336682
- [4] Setting the PACE in Product Development, A Guide to Product and Cycle-time Excellence (Paperback) by Michael E. McGrath (Author)
- [5] The Economic Impacts of Inadequate Infrastructure for Software Testing. Prepared by RTI for the National Institute of Standards and Technology. May 2002
- [6] First-Mover Advantages from Pioneering New Markets: A Survey of Empirical Evidence WILLIAM T. ROBINSON School of Business Administration, University of Michigan, Ann Arbor, MI 48109, U.S.A. GURUMURTHY KALYANARAM School of Management, 2601 North Floyd, University of Texas" at Dallas, Richardson, TX 75083, U.S.A. and GLEN L. URBAN Stoa School of Management, MIT, 50 Memorial Drive, Cambridge, MA 02139, U.S.A. Review of Industrial Organization 9: 1-23, 1994 © 1994 Kluwer Academic Publishers. Printed in the Netherlands.
- [7] European Commission (2003-05-06), Recommendation 2003/361/EC: SME Definition,
<http://europa.eu.int/comm/enterprise/enterprise_policy/sme_definition/index_en.htm>. Retrieved on 24 May 2007
- [8] “Small to Medium Enterprise Outsourcing Guide”, Trestle Group Research.
- [9] Deloitte Consulting 2008 Outsourcing Report, “Why Settle For Less?”, Deloitte Consulting Outsourcing Advisory Services,
- [10] Romilly Cocking, “Why agile development and outsourcing don't mix”, <http://www.cocking.co.uk/blog/2004/11/why-agile-development-and-outsourcing.html>
- [11] “Adopting and Benefiting from Agile Processes in Offshore Software Development” by Andrew Filev. The Architecture Journal, Journal 8, June 2006 (see <http://msdn.microsoft.com/en-us/library/bb245671.aspx>)
- [12] “Everything You Know About Offshore Outsourcing is Wrong” by Linda G. Hayes, Datamation Magazine (see <http://itmanagement.earthweb.com/columns/article.php/1856621>)
- [13] “On the Sustained Use of a Test-Driven Development Practice at IBM”, by Julio Cesar Sanchez, Laurie Williams and E. Michael Maximilien, IEEE Agile 2007.
- [14] Deloitte Consulting 2008 Outsourcing Report, “Why Settle For Less?”, Deloitte Consulting Outsourcing Advisory Services.

Biography



Dr. Mike Bartley
Founder & CEO, Test & Verification Solutions, UK

Mike graduated from Bristol University with a PhD in Mathematical Logic. Since then he has studied with the Open University obtaining an MSc in Software Engineering and an MBA. He has been involved in both software testing and hardware verification for about 20 years. He started in software testing specialising in formal methods, before moving to verify hardware at ST Microelectronics and Infineon. Most recently, as Test and Verification Manager at Elixent (now Panasonic) and ClearSpeed in Bristol, he has been responsible for the sign off of complex hardware/software products. Mike has had numerous papers published, presented at a number of conferences, and has written on software testing for the Open University. He has been involved in a number of successful outsourcing relationships.