

# Test Managers: Chameleons of the Project World

## Authors

*James Lyndsay is an independent Test Strategist, based in London. He's spent over ten years in software testing, and has been the principal consultant at Workroom Productions since its formation in 1994. As a consultant, he's worked in a variety of businesses and project styles; from retail to telecommunications, from rapidly-evolving internet start-ups to more traditional large-scale enterprise. James is a regular and popular speaker at test conferences around the world, and has been awarded 'Best Paper' at major conferences in the US and Europe.*

*Julie Gardiner is a Test Manager with over ten years IT background. Her testing and test management experience has been gained across a broad range of industries, including Financial Services, Utilities, Retail and the Public Sector. She has contributed to many large scale projects at a number of blue chip organisations, including one of the Top 3 outsourcing businesses in the UK. Currently based in Nottingham, Julie's current role allows her to develop and apply her primary interests of risk-based testing and process improvement.*

## Abstract

*Test Managers, like Chameleons, have to adapt themselves to an ever-changing environment in order to survive. They also have to find - and dispose of - plenty of bugs. This paper puts forward two models of test management; the first sets out the ways in which test managers have to adapt their roles depending on their audience, and who they represent, while the second illustrates the responsibilities of a test manager across different phases of testing within a project.*

## Contents

Test Managers: Chameleons of the Project World.....	1
Authors.....	1
Abstract.....	1
Contents .....	2
1. Introduction .....	3
1.1. Scope .....	3
1.2. Two models and a Chameleon.....	3
1.3. Using the models.....	3
2. Chameleon Model 1 – interactions .....	4
2.1. Talking to the Test Team.....	4
2.2. Talking to technical people outside QA .....	5
2.3. Talking to Project Sponsors and their teams .....	5
3. Chameleon Model 2 – phases and responsibilities .....	6
Project initiation .....	7
Pre-delivery .....	7
Ongoing delivery .....	8
Approach to Live.....	9
Live operation.....	10
3.6. Throughout the project.....	10
4. Conclusion.....	10
5. References and further reading.....	11
6. Contacts and updates.....	11

*Copyright: This paper © Workroom Productions Limited 2003. The rights of James Lyndsay and Julie Gardiner to be identified as the authors of this work have been asserted in accordance with sections 77 and 78 of the Copyrights, Designs and Patents Act 1988.*

## 1. Introduction

Test management calls for a remarkably broad range of skills. As a test manager on a project, you can be involved at all phases, and will work with a wide variety of people from inside and outside the project. This paper aims to help both experienced and novice test managers to manage and make better sense of their roles.

### 1.1. Scope

The paper covers those aspects of the job that are specifically to do with managing test activities and teams – but does not intend to cover those aspects that are to do with management in general. Likewise, the paper covers the aspects of the role within a project context; issues that concern ongoing management between projects are not dealt with here. We have further assumed that the project in question is not a test-only project, and that the testing activity is carried out in conjunction with other development activities such as design and coding, and has further relationships with commercial pressures and tasks such as deadlines, budgeting and marketing.

We have made this paper as general as possible within these constraints. In particular, we have tried to construct a model that can be generalised across styles of project and product. The models described may be applied to waterfall or iterative approaches, and to bespoke systems as much as to shrink-wrap software and ASP-style services. Readers are left to weigh the information and advice in the light of their experiences on their own projects, and are encouraged to adapt these models to suit their circumstances.

### 1.2. Two models and a Chameleon

A Chameleon is a lizard; scaly, cold-blooded and sporting a long tail. In these characteristics, a chameleon is, of course, nothing like a test manager. However, chameleons are notable for more than simply their lizardly features. They spend much of their time seeking out bugs to eat, and they change colour to match their background and situation. These characteristics form the basis of our metaphor, and in this paper, we describe two models that parallel these ideas. We believe that by adapting to their environment, test managers maximise their effectiveness – and minimise their vulnerability.

The first model looks at the ways that a test manager adapts when representing different interests to different groups.

The second model looks at the ways that the stage of testing affects the responsibilities of a test manager. It details the activities and decisions that they might take on to make best use of the test team, the deliverables they may be responsible for, and the expectations that other project members may have.

### 1.3. Using the models

The reader may make whatever use of the models he or she sees fit. However, the models have been constructed with the following uses in mind;

- To help test managers plan their work in various phases of the project; in particular to reveal missed aspects in past phases, and to help anticipate unexpected roles and requirements in future phases.
- To act as a checklist and reminder of the breadth of the role. This will help the test manager to keep each aspect of the role in perspective, and will help the people round him or her to come to an appreciation of the complexities and set their expectations accordingly.
- By describing an overall view, to help the test manager continue to think both tactically and strategically, and avoid getting stuck in a rut of fire-fighting and reactivity.

## 2. Chameleon Model 1 – interactions

In the bug-eating world, chameleons are notable for their ability to change their appearance to suit their background. Test Managers, too, must pay attention to their environment, and change to suit.

Test Managers need contacts throughout the project in order to work effectively. Those contacts have very different expectations of testing, and the Test Manager must be able to present his or her information and ask questions in a way that is appropriate to the skills and interest of the people they are talking with. We have identified three groups: people within the test team with knowledge of how testing is done; technical people outside the test team who produce and fix code, environments or data and so are interested in faults found; and people who have little hands-on interaction with the construction of the product, but are interested in testing as a service, and the general information that it needs and can produce.

Test managers must also represent a broad range of interests. This model groups those interests into three areas; representing the interests of the test team, representing the interests of the project, and representing the interests of those who stand to be affected by the final product.

Representing Audience	Test Team	Technical/ Non QA	Project Sponsors (and teams)
Test Team	Vision Coaching Management	Bug advocacy Fix priority Testability requirements	Ask for resource Clarification of <b>product</b> requirements <b>Anticipated</b> risk/value to customer
Project	Time & Cost available	Resource negotiation and dependency	Clarification of <b>project</b> requirements Value of Testing
Product stakeholder	Test design/risk Values	Severity of bug	<b>Revealed</b> risk/value to customer

### 2.1. Talking to the Test Team

The primary interaction of a hands-on test manager with his or her team is in the role of a technical expert and decision maker. In this role, the test manager will have a vision of the process and direction of testing. This vision may be formalised in a documented test strategy, but whether the strategy is written down or not, the test manager must actively manage his or her team to proceed toward a technical goal. Without this, the test team is rudderless. It may become reactive, and be overwhelmed by unforeseen urgent problems, or it may concentrate its efforts inappropriately. The test manager, as an expert and with an eye on overall responsibility, will also take an active interest in the skills of their team, coaching the team members

Once the emphasis shifts away from this primary interaction, the test manager is behaving in a more generalised way, and is perhaps no longer behaving as simply a test manager. However, there is a strong pull from other interactions, where the test manager represents the interests of outside parties to the test team.

Taking on the interests of the project as a whole, the test manager will spend time with his team in setting and justifying constraints in terms of time and cost. In doing this, he or she may well conflict with the test team's natural focus on quality and breadth of scope – and it is part of the role of a test manager to both understand and be able to prioritise these conflicts. A good test manager will happily play Devil's Advocate to the team, knowing that such challenges are an effective way to encourage the team to innovate and realise greater value from their time.

When representing the interests of the ultimate users of the product, the test manager will want the cooperation and contribution of the test team. It is the test manager's responsibility to identify risks, and by considering value of various aspects of the product, lead the design of tests to mitigate the risks and maximise the value of the product. Some testing may be prioritised so low that it is left out, while in other areas, the test manager may advocate particular techniques to rapidly reveal common or significant faults. To do this effectively, the test manager must understand the needs of those stakeholders who are focussed on the product.

## 2.2. Talking to technical people outside QA

It has been said that testers are the ‘headlights of a project’ [1] (although this confuses the chameleons metaphor somewhat). The test manager is an important source of information to the whole project, particularly other technical teams outside the test group.

The primary interaction with this group is, once again, as a peer, representing the resource needs of test activities and negotiating support and access. This needs political savvy, and a good understanding of the effects of these resources on the overall test effort – without understanding what is being enabled, and what is being given away, the test manager can’t negotiate to maximise all-round value.

However, the test manager has important roles to play in interacting with this audience from different viewpoints. The test manager will often ‘own’ a bug until other groups grasp its importance – taking the customer’s viewpoint, he/she can help set the overall severity of the bug to accurately reflect impact on the stakeholder. Taking the test team’s viewpoint, the test manager can work to establish a priority for the fix that allows the simplest path to greater coverage. He/she can also act as an advocate to promote bugs that the test team have experienced but that are not taken as seriously by external teams, and so play a part in their prompt assignment. Finally, representing the expertise of the test team, the test manager can influence the acceptance of testability requirements with reference to simpler testing, increased knowledge and expected reduction in live bugs.

## 2.3. Talking to Project Sponsors and their teams

Few people in this audience necessarily understand software testing – or want to. Rather, they need to know how to help the teams to do a good job, and whether that ‘good job’ has produced something valuable.

This focus means that the primary interaction of a test manager with this audience is as someone who knows the product. Representing a product stakeholder, the test manager will be able to talk of confidence and the parts of the product that are of greatest value, of weaknesses and their effects of those weaknesses on the risk that each customer takes on with the product.

An earlier interaction, and one that maybe more important but is likely to be paid less immediate attention, is taking the test team’s needs to the project sponsors and their teams. This is not simply to ask the purse-holders for project-related resources such as more money, time or people, but to better understand the product that is to be built. As the product progresses toward completion, the test team becomes a valuable (indeed, often the only) source of information about the possible risks to the customer, particularly those that had not been understood at the time of design and coding. In doing this, the team may also be able to identify areas of value to the customer that may justify a timely shipment, or reduced functionality.

Finally, with this audience typically exerting some kind of budgetary control, the test manager will have to represent project interests in order to clarify project requirements of timeliness, cost or benefit. He/she needs to be able to succinctly sell the value of testing to get control of an appropriate fraction of the overall budget, or to increase the budget in response to an opportunity to add significant value.

### 3. Chameleon Model 2 – phases and responsibilities

This model shows the wide range of responsibilities of test management. For clarity, these have been grouped into four areas;

- The project as a whole will have **expectations** of the test team, and of testing. The test manager bears ultimate responsibility for understanding and meeting these expectations.
- The test manager is responsible for a range of **decisions**. The decisions described below are test management decision; project-level decisions will be made in collaboration with other members of the project team, and are treated instead as project management decisions.
- The test manager is responsible for the **activities** of the test team – and of themselves. Again, these aspects have been considered from the point of view of test management, rather than more general activities of managing or testing.
- The test manager is responsible for **delivering** certain parts of the project. These deliverables may be documents, procedures, or shared understandings, but the key is that they are used by the test team and others, and the responsibility for their production lies ultimately with the test manager.

Each of these aspects will change with the phase of the project. The following table illustrates the different responsibilities, phase by phase, and is followed by a discussion of individual phases. Note that these phases are phases of testing, not of project delivery.

Phase	Project Initiation	Pre-delivery	Ongoing delivery	Approach to live	Live	Always
<b>What</b>						
<b>Expectations</b>	Test/quality advocate Expertise – particularly cost/value	Consequences of changes to scope, timescales, budget or risk Agreed reporting and metrics	Current state of testing: - coverage - risk - metrics - faults	Assess go live Understand outstanding customer risk	Guilt for problems missed by test Support ongoing live	State of team, testing, product
<b>Decisions</b>	Methodology Test techniques	Risk and scope of testing within project limits of cost and time Implementation approach; techniques, skills, tools, environment	Prioritisation of tests	Tests to omit	Any more testing?	Next test activity
<b>Activities</b>	Feasibility studies Brainstorming Product risk assessment + mitigation	Set up infrastructure (lab, tools, skills, people) Set expectations Product design for testability and quality Planning, scheduling and designing tests	Troubleshooting Test review board Gather metrics Refinement to plan Strategic direction	Plan to wind-down team + lab Transfer knowledge Liaison with external groups	Analyse faults Propose test / product improvements Expert diagnosis of live	Coaching
<b>Deliverables</b>	Initial Test Strategy Reporting mechanism	Plan(test/ project) Progress report	Information Bug reports + diagnostics Testware	Summary report Known faults	TPI document Further test design	Enhance/main tain process

### 3.1. Project initiation

The test manager may be selected during this phase, or may be part of this team from the opening moments. If the latter, he or she is likely to be shared with other projects and must allocate time appropriately. As project initiation teams tend to be small, the test manager may have few resources to call on, and will have plenty of opportunity for technical, hands-on interaction. By the same token, he or she will be heavily involved in setting the strategic direction for the project, and defining the test team's role within it.

Project sponsors will **expect** the test manager to act as their test advocate, articulating the costs and benefits of testing, and able to negotiate on scope, quality and budget. They will expect the test manager to push for a project decision to increase or change available resources, based on likely benefit, and are likely to want him/her to understand the advantages and disadvantages of the full range of available options. The test manager may not be the only advocate of quality on the team, but others will expect him or her to be able to put both a cost and a value on quality.

With an eye on risk mitigation, the test manager will take **decisions** on the methodology and test techniques that can be used to get the most value. These decisions will be based on the technology and expected failure modes, and may be influenced by budget, timeframe and available skills.

The test manager will work closely with other project initiators, and will be involved in **activities** such as brainstorming sessions and feasibility studies. He or she will have a vital role to play in helping the project come to an understanding of the risks associated with the project and the product – and may be expected to take the lead in product risk assessments.

This understanding of risk combines with methodology decisions to make up an important part of the test strategy, which is the test manager's primary **deliverable** of this phase. Other deliverables include a regular reporting and metrics process to allow the right information to flow from the testing team to the right targets in a timely manner. This process will be started during or at the end of the project initiation phase to allow management information to flow into the overall project plan.

**Period:** from the first discussion to the point where the project has an acceptable plan and budget.

**Significant non-test activity:** Analysis of feasibility, constraints, cost/benefit, project scope. Choice of technology, architecture, project approach, key staff. Identification of sponsor, stakeholder, overall requirements.

### 3.2. Pre-delivery

During the pre-delivery period, the test team have nothing to test. The overall project, on the other hand, will be hard at work creating some part of the product. The test manager can use this period to build a team, a process and a plan to integrate testing with the rest of development.

The duration and form of this period is highly dependent on the approach of the overall project. Iterative developments, or projects with significant focus on early deliverables, will move rapidly from the start of non-speculative work to the first testable deliverable. On the other hand, if the project plan splits test teams by project phase, the User Acceptance test team may take the first deliverable late in the project.

It often proves harder than expected to arrive at the first testable deliverable – which typically leads to an extension of this phase. Flexible teams and plans are better able to cope with such delays; if recognised promptly and approached proactively, such problems can be turned into an opportunity to improve the infrastructure and support of testing to allow more efficient or more effective testing.

As testing is not directly involved in development before the first testable deliverable, it is possible for budgets and timescales to shrink and for scope and risk to inflate without considering the effect on testing. Project sponsors will **expect** the test manager to provide information to allow the true cost and likely consequence of such changes to be understood – even if peer managers on the project may not welcome the information. Project Management will expect the test manager to agree on some way of measuring the progress of testing, and the frequency and media of reports and metrics.

The Test Manager will **decide** on the scope of testing. This will generally start with a test-related risk assessment, and the Test Manager may act as a facilitator between representatives from across the business; designers, coders, external factors (ie customer, interface or regulatory representatives) and live support. The scope of testing will be defined in great part by the ways that the risks can be

**Period:** from the start of non-speculative work to the first testable deliverable into the test team.

**Significant non-test activity:** Requirements analysis; building development tools, procedures and environment; production of design documentation; production of code, data structures, interfaces. Prototyping, architecture definition and construction, formulation of algorithms.

addressed within project budget and timescales by testing activities in later phases – and the test manager will need to balance the scope with overall cost and deadlines. The Test Manager will also decide the techniques, size of environment(s), skills needed within the test team and use of test tools.

Testing is very much on the critical path in later phases. During pre-delivery, the Test Manager's **activities** will be directed toward enabling testing during those later phases to be flexible, smooth and effective. These activities are likely to include the requisition and construction of a test lab, recruitment and/or training to meet identified requirements, and implementation of test support such as test automation, data manipulation and other tools. Lead times on each of these can be long, and they should be kicked off early in the phase. A variety of stakeholders will have expectations of the team, and the manager must work to negotiate these expectations, setting goals that are both achievable and useful. The Test Manager is also likely to work towards enhancing the testability of the product, by acting as a stakeholder in change control, prior testing and quality standards. However, their greatest influence on the quality of the product may be by being directly involved in the design.

A key activity in this and later stages (*ongoing delivery* and *approach to live*) is the preparation of the test plan and schedule, and the test manager will commonly also be an active contributor to the overall project plan. The details of these plans may be highly dependent on the first delivery, and the Test Manager should plan appropriate time (close to delivery) to allow the details to be introduced as the deliverable comes into focus. The test manager will also co-ordinate and participate in the design of tests, paying particular attention to choice of test techniques and data.

The plan itself will be one of the **deliverables** of the phase, with other deliverables dependent on the activities and expectations of the project. One deliverable which is sure to be sought is a regular progress report. At this stage, the report will detail the progress of testing, as no part of the product has been available for assessment.

### 3.3. Ongoing delivery

This phase is inherently cyclic; unlike most other work on a project, the same basic activity may be repeated over and over. However, there is plenty of work that is iterative in that it builds on what has gone before, and some work may only need to be done once. Genuinely one-off work may be worth examining to see if it belongs in another phase.

**Period:** from the arrival of the first testable deliverable to the point where the software is ready to be deployed  
**Significant non-test activity:** Changes to requirements, design. Production and integration of code.

The Test Manager will be **expected** to have his or her finger on the pulse of testing at all times. Questions such as 'What have we tested?', 'What risks have we addressed?' and 'How good is the product?' all contribute to project activities that range from prioritisation of tasks to the ultimate "go-live" decision. Some test teams only gather definitive information on a weekly or monthly basis; on a more agile project, this may mean that testing does not have a voice in day-to-day project decisions.

During this phase, the test team is highly reactive, and **decisions** will focus on prioritisation of tests and test activity. Planned tests may be put off if software is absent, to be re-written, or seems to have a lower risk than expected – and may be bought forward if the software needs an urgent assessment, or seems riskier. These decisions may be made collectively, but are the responsibility of the Test Manager.

The **activities** of the Test Manager will also be reactive. The Test Manager will act to rapidly resolve unpredictable problems that can slow down testing – particularly environment, lateness/poor quality of testable deliverables, and tool issues. He or she may also spend an increasing proportion of time in meetings with representatives from other teams, discussing the prioritisation and severity of discovered product issues. Metrics need to be gathered and analysed regularly to allow immediate decisions and longer-term trend analysis. Testing activities need to be re-planned in response to events inside and outside the team, and it may be necessary to drum up political support at short notice. The Test Manager may take a proactive approach by setting up processes (such as test environment service level agreements, test review board, automated task allocation and metrics) to support these reactive activities, and should also keep a strategic eye on the direction of testing and areas of value and weakness in the product. It is also likely that he or she will continue to be involved in setting the design and approach of key test areas – particularly in response to greater understanding about the patterns of risk in the product, and increased knowledge of value of that product to its ultimate customer.

The Test Manager is responsible for the **delivery** of information. At some points in a project, incident reports can act as a greater driver of new work than the customer requirements themselves. The test

manager is responsible for these reports, and should work to ensure that they are good enough to allow accurate and efficient responses from other teams. Metrics from the set of incident reports are a key measurement in assessing the quality of the product, and are a related deliverable. A further important deliverable is the testware produced by the team. Testware includes automated tests, data, and expected results – and has a tangible value in later stages of the project. Testware may also be a required deliverable to comply with regulations, or as possible evidence in a legal challenge.

The key deliverable of any testing team is information about the product and the project. While responsible for this information, the test manager should ideally avoid being its sole conduit; automation of the gathering and distribution of such information in this reactive phase pays dividends in terms of accuracy, independence, and time saved.

### 3.4. Approach to Live

This phase is entirely absent from some projects. If the product is always releasable – or if the product is unmitigatedly awful – it may be taken live from any point in ongoing testing. However, many projects change their testing significantly as the live date approaches; requirements are stable or frozen, deliveries tend to be made mainly from fixes, tuning or configuration may throw up more problems than code. These changes are driven by outside influences; identified customers, influence of beta testing, demonstrations and reviews, training, marketing budgets and booked advertising.

**Period:** from the point where stakeholders have a believable live date to the date of deployment

**Significant non-test activity:** Marketing, pre-sales. Training and demonstration. Data construction, cleansing + migration. Roll-off of design/coding team. Construction of live environment / install / user manual. Deployment.

A key decision on any software project is the decision to go live. If the decision is not to go live, the project may oscillate between phases for a while before genuinely approaching live operation. Indeed, one of the signifying characteristics that a struggling project has at last achieved its expectations of quality and scope is that this phase is finally recognisable and distinct from any preceding confusion. Projects may go ‘live’ into a small-scale pilot or beta test many months before the business or customer feels confident enough for a full-scale rollout; a characteristic *approach to live* phase often proceeds the rollout rather than the previous ‘live’ decision.

During the *approach to live*, the Test Manager will be **expected** to participate in the go-live decision – indeed, it may be the project sponsor’s primary expectation of their role. The Test Manager is unlikely to be solely responsible for the decision, but their input is vital, and may be requested often. A related expectation is to know about customer risk – the test team is often the closest to the product in use, and the Test Manager is expected to understand the risks the customer takes on when they take on the product. The Test Manager will be expected to have this information available and up to date throughout this phase; the day before go-live is too little, and too late.

While still reactive, testing at this point is influenced more by time than by scope. The Test Manager’s **decisions** will concern what to omit, rather than what to prioritise. Configuration and integration work may reveal a need for still more tests that have never been considered, yet time pressures may force the Test Manager to omit these and many others. Such decisions are hard, and the Test Manager may need to seek support from other areas of the project to understand the risks, and gain support.

Although the test manager may still be designing some new tests and working reactively as in the previous phase, more strategic **activities** will start to make demands on his or her time. With the approach of live operation, the Test Manager needs plan to wind down the test team and decommission the test lab. This means not only organising the transfer of relevant knowledge, but also interpersonal, non-test issues related to the change of pace and the impending break-up of the team. He or she may need to liaise with external testing groups – beta testers, customer’s own user acceptance test team, regulatory testers etc. Go-live meetings will also place a further unavoidable drain on their time.

The Test Manager will probably need to **deliver** a summary report on testing when the go-live decision is taken, although a detailed report may be delayed until after the product has moved to live operation. The report will contain the cumulative progress and metrics and outline areas where risk is highest, and where value is most significant. It should also highlight those areas where knowledge of the product is too sparse to influence the decision. He or she may also need to deliver a list of known problems, or operations manual based on common problems and known workarounds.

### 3.5. Live operation

Live operation may mean a range of things depending on the product; from commercial release, to successful internal deployment. As a project phase, *live operation* may not seem to have much call for testing – but the Test Manager has a role, at least in the early days. Maintenance involves new releases, and may be considered a new and different project.

**Period:** from the moment of deployment for the remaining life of the product

**Significant non-test activity:** Analysis for enhancements, new customer potential. Live customer support / ongoing operations support (including diagnosis, workarounds, notifications).

The Test Manager may find that the project has a wealth of **expectations**, even after the system has gone live. Although such expectations should be driven by the expectation-setting activities in *pre-delivery*, two common expectations are covered here. The Test Manager may be expected to comment on problems found in live operation – particularly with regard to how the problems slipped through. While very helpful for Test Process Improvement (see below), this can lead to an ongoing cycle of blame. The Test Manager may also be expected to help support ongoing live operations, or the helpdesk, for a bedding-in period, and should look to transfer relevant knowledge rapidly and effectively to the people who bear genuine responsibility for these areas.

Many projects need the Test Manager to take a **decision** on any further testing that should be carried on in live operation. This may be a mopping up exercise, where untested areas are tested before customers get to them, or it may be part of a process of ongoing assessment of live operation. The first tends to be led by functional testing, while the second is led by non-functional testing. Configurable systems, or software that attracts diverse user bases, may need further testing as use of the software changes.

Test Managers are likely to spend time on post-partum **activities** and recommendations for overall improvements. The project (or its surrounding organisation) will want to learn from its experiences, and the Test Manager has vital information for such a learning process. The Test manager may be involved in a review of faults found – particularly comparing them with faults found outside, testing or after live. On a reactive level, he or she may be asked for expert diagnosis of live problems.

**Deliverables** from these activities may include recommendations for Test Process Improvement, a formal fault analysis of the product.

### 3.6. Throughout the project

Some aspects of test management do not fit the phased model above. This paper does not deal with aspects that are to do with being an expert tester, nor with aspects that are to do with management, but this section covers those aspects of test management that are not dependent on phase.

It is reasonable to **expect** the Test Manager to always know the state of the team and of testing. The state of the product may not be so easily known, but it is reasonable to expect the him or her to be able to communicate how much is known about the product. The test manager always needs to hold responsibility for **deciding** what the team will do next, and will be involved in coaching **activities** throughout the life of the team. Finally, the Test Manager is responsible for **delivering** enhancements and improvements to the test processes where appropriate, and maintaining them where necessary.

## 4. Conclusion

In “*The Mythical Man-Month*”[2], Frederick Brooks says “*the techniques of communication and organisation demand from the manager as much thought and as much experienced competence as the software technology itself.*”

The role of a Test Manager is vitally different from the role of an expert tester. By considering the models of interactions and phases put forward in this paper, we believe that new and experienced test managers will gain a novel perspective on their roles.

## 5. References and further reading

- [1] Kaner, Bach, Pettichord 2002. *Lessons Learned in Software Testing*. Wiley
- [2] Brooks 1995. *The Mythical Man-Month, Anniversary Edition*. Addison-Wesley
- [3] Rosenau 1998. *Successful Project Management, Third Edition*. Wiley

## 6. Contacts and updates

We would like to encourage readers to contact us to discuss the paper – particularly with regard to ways in which you have found this paper useful, or have found that it does not fit your own circumstances.

James can be contacted at [jdl@workroom-productions.com](mailto:jdl@workroom-productions.com)

This paper will be updated with new information and ideas, and will gain an addendum illustrating application on the model to project scenarios. For further information and updates to this paper, see the Workroom Productions website at <http://www.workroom-productions.com>