

IE TESTING CONSULTANCY LTD



Software Testing
Consultancy - Training - Management - Practise
Associated with ISABEL EVANS - *Common Sense Quality*

Restdale House, 32/33 Foregate Street, Worcester WR1 1EE UK

THE TROUBLED PROJECT Best Practice *from theory to reality...?*

Isabel Evans - IE Testing Consultancy Ltd

Abstract:

Standards and best practice are great in theory but when a troubled project is midway through it can be difficult to introduce them without slowing progress.

- ⊙ Is the start of user acceptance testing too late to say, "you should have used the V model"?
- ⊙ What really helps introduce good practice to a troubled project?

Deming said, "People and organisations only change when the pain of staying the same is too great."

For the mid-term troubled project introducing an improvement programme might be kill or cure. This presentation uses practical real-life projects to show what improvements can make a difference. It includes test design, strategy, documentation and measurements & reporting.

Key Points:

- ⊙ Do standards slow down progress?
- ⊙ What helps a mid term project in trouble?
- ⊙ Are standards just a tool-set rather than a rulebook?

Isabel Evans - IE Testing Consultancy Ltd. - Biography

Isabel is the principle consultant at IE Testing Consultancy Ltd. She has nearly 20 years experience in the IT industry, mainly in quality management, testing, training and documentation. She has helped organisations in development of procedures, standards and methods to aid testing of software during development and maintenance projects. She has managed test groups, and performed testing design and development for acceptance and system testing of packages and bespoke systems. She has also provided Test Management, Implementation Planning, Quality Assurance Support, Release Management, and Customer Support for IT organisations. In the last five years her projects include training, consultancy and process improvement projects as well as "hands on" test and implementation management in major projects.

Isabel is the author and an accredited tutor for an ISEB accredited Foundation Certificate in Software Testing course. She has presented at many conferences. She is an active member of the BCS Special Interest Group in Software Testing Standards Working Party, especially working on the Usability, Installability and Conversion Testing sections of the Non Functional testing standard.

Why look at troubled projects?

W Edwards Deming said, "People and organisations only change when the pain of staying the same is too great." This can mean that when a project is in trouble its owners start to look at radical changes to the way of work. It is often the time when management calls on consultants and outside experts to assist the project and find fast solutions to problems. As practitioner, consultant and trainer I have observed that these efforts to effect change during crisis are sometimes successful, but sometimes fail. Why is this? In the last few years, I have observed a number of behaviours that help or hinder the process of change. Some initiatives help in the present crisis, but other can only be applied as part of lessons learned. In this talk and paper I want to share with you some of those observations. This includes looking at when it was appropriate to use consultancy, and what types of training to use both in the immediate troubled project and as an aid to future change and improvement.

Telephone: +44 (0) 1905 726727
Email: isabel@ietesting.co.uk



Fax: +44 (0) 1905 726766
Web Site: <http://www.ietesting.co.uk>

In summary, if you are suffering from a troubled project...

- ⊙ You and the project / organisation are in pain
- ⊙ Something needs to be done - and fast
- ⊙ You are about to make a quick decision... which may have unforeseen consequences
- ⊙ What will help and what will hinder you?
- ⊙ Long and short term solutions to problems

The troubled project revealed...

Here are some of the typical problems and some thoughts and quotes from projects in a variety of organisations over the last 10 years. Not all the projects exhibited all the problems, but a troubled project will often have a collection of problems that conflict with each other; the solution for one problem would exacerbate another problem.

Characteristic perceived problems for a troubled project:

- Time** - The project end or intermediate dates and deadlines have been set - perhaps over ambitiously - and are approaching or have been over-run. The project end date is going to be missed, or end processes are going to be cut. Time seems to move faster each day.
"We are on a roller-coaster and it's the last slope"
- Money** - The project budget or testing budget was set some time ago and now seems too small. It is about to be or is being exceeded. Some activity or resource has to be cut.
"Testing is very expensive - can we do less of it?"
- Expertise** - Expertise and experienced resource limited - the organisation has never done anything like this before, or expertise resides with a small number of experts who are in constant demand.
"Only three people understand this part of the system - J is on maternity leave, K is on holiday and L is seconded full time to another project..."
- Resource** - Perception that bigger team would reduce the timescales, more people being requested than is available, team working increasingly longer hours to complete the work, especially experts. Conflict for scarce resources (environments, tools, expertise).
"No progress this week, as all my staff have been moved to project X"
- Motivation** - Team is fed-up, so are the team leaders. They are tired, ill, bored, over-stressed, feel unappreciated. They do not believe what managers are saying.
Team leader to IE: "the sooner I am off this project the better"
Team member to IE: "you are the first person to be enthusiastic about this project"
- Quality** - The quality of deliverables is not known / has been compromised. Quality decisions not linked to risk. Deliverables marked as complete on the project plan even when not complete. Delivered software fails. Reputation of the project is falling.
"that team... I expect we'll be up all night with it..."

Underlying those immediate, visible problems are generally several causes or reasons for the visible problems. Again here are some of the typical reasons from projects in a variety of organisations over the last 10 years.

Characteristic underlying reasons for the troubled project's problems:

- Unclear goals** - Goals for the project are not clear or conflict. Different parts of the project or organisation have different goals, hidden agendas, political and power plays. Goals not associated with risks. Testing and other quality activities not related to goals or to risks.
"... why is this important?"
- Communication** - Communication poor within the project and across project boundaries. Key people not on speaking terms (!) Progress messages communicated upwards but not downwards or sideways. Only negative messages given to the team. Or, only positive messages given to the team.
"... nobody tells me anything..."

Poor knowledge - No clear understanding of current position. No measurement taken. No method of charting progress. Information available but too difficult to get.

"It takes me 2 days to get together the test progress statistics for the management meeting"

Scope slippage - Informal changes made to the scope of deliverables with no changes agreed to time or budget. Additional deliverable/functionality not working or agreed functionality compromised.

Having added additional functionality that did not work, removing the functionality had to be at a high level as the customer expectation had been built - reversing that expectation back to the agreed specification was damaging.

What helps a mid term project in trouble?

When problems arise during a project we need to look at both *long term (strategic)* and *short term (tactical)* solutions. When we are in the midst of the troubled project, it becomes difficult to separate long-term strategy and short term tactics.

Proposed solutions to the problems tend to be addressed to the perceived problems rather than the underlying reasons. The underlying reasons for the problems also need to be addressed.

Each solution has its merits and its problems. I will show below for each solution the short and long term possibilities. Some ideas will affect the project immediately, others are best applied as lessons learned to the next project. However, once the pain of the current project ceases so not forget it - it is still your motivation for change.

The desire for change may be motivated by the pain of the current project, or by pain in the wider organisation. Characteristic motivations or reasons for improvement projects include:

- ⊙ Fire fighting
- ⊙ Marketing / image
- ⊙ Market forces / legal requirements
- ⊙ Improve status / grade of product
- ⊙ Reduce faults / increase quality
- ⊙ Reduce costs
- ⊙ Reduce manpower
- ⊙ Organisation change
- ⊙ Selling the company
- ⊙ Merger pressures
- ⊙ Desire to impress
- ⊙ Need to impress a new client...
- ⊙ Market forces
- ⊙ Genuine desire to improve quality

The motivation or reason for the improvement project will affect what is done, the types of change selected and the outcome. You need to understand the reasons for change and the reasons for training, improvement initiatives, or adoption of standards. If there are conflicts of motivation between different groups, for example the sales director, development director, project managers, and teams of developers & testers, then the changes will not succeed. Make sure that the changes - whether for the benefit of the troubled project or for the longer term benefit of the organisation - have clear goals, that the risks to be addressed are clearly understood, and that achievable, measurable targets and indicators are used.

Overall, remember that you have to drive the change with conviction:

- ⊙ be clear whether you are looking for first aid or a long term cure for the problem.
- ⊙ be enthusiastic honest, open minded, and use people's experience
- ⊙ what doesn't work is not telling people the reason things are happening

In the rest of this paper I will go through the possible solutions and assess them:

- ⊙ Do they work?
- ⊙ What can you try for the immediate project?
- ⊙ What can you try for the long term?

The table below shows some, but not all, of the ways that problems can be addressed.

Problem	Possible solution to perceived problem	Underlying problem that needs addressing
Time	Work harder Add resource De scope	Clarify goals and risks - is the date the biggest risk? Address scope slip
Money	Remove resource De scope	Clarify goals and risks - is the budget the biggest risk? Address scope slip
Expertise	Training Bring in outside consultants	Clarify goals and risks - what expertise do you need? Improve metrics / gain knowledge
Resource	Add resource Existing resource asked to do more De scope	Address scope slip Improve communication
Motivation	Training	Clarify goals and risks Improve communication
Quality	Set improvement targets Introduce standards, methodologies Training Bringing outside consultants	Clarify goals and risks improve metrics / gain knowledge Address scope slip

Does working harder help?

Yes: look for efficiency and effectiveness

No: KAROSHI (death by stress of work) also increased mistakes, de-motivation, illness

Possibly try: focus days, deputies for experts, display progress openly, remove admin tasks, de-scope, complete later, understand whether time is the real driver

For the future: assign deputies and apprentices earlier, display progress in terms the team understand, Mythical Man Month, manage Board expectations, estimate honestly!

The typical problems that we attempt to resolve by working harder are:

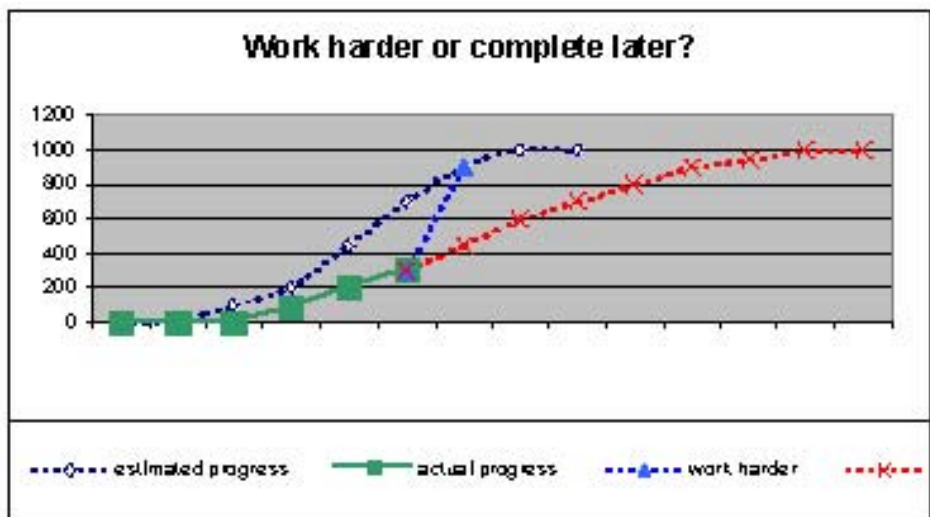
- Time** - **The project end or intermediate dates and deadlines have been set - perhaps over ambitiously - and are approaching or have been over-run. The project end date is going to be missed, or end processes are going to be cut. Time seems to move faster each day.**
"We are on a roller-coaster and it's the last slope"
- Resource** - **Perception that bigger team would reduce the timescales, more people being requested than is available, team working increasingly longer hours to complete the work, especially experts. Conflict for scarce resources (environments, tools, expertise).**
"No progress this week, as all my staff have been moved to project X"

It is important to consider the effects of this tactic carefully. Working harder to address the projects problems may work if you are looking at an unusual peak in work, but it is not a general long term solution. Address the efficiency and effectiveness of the team, rather than just hours worked. Simple time management techniques that may be obvious to you as an experienced person may not be obvious to the team.

- ⊙ I have spent time with individuals looking at the deadlines on the plan, that they affect. I then ask them *"What do you need to do this week in order to meet that deadline in 2 months?"* From that first question we build plans for *"today"* *"this week"* and *"this fortnight"*. They then report on a rolling week of fortnight. Like all the best ideas - I learnt this one from one of my project managers!

Remember too that increasing hours work and pressure on the team may cause illness and even death. The Japanese have a word for this - Karoshi "Death by work related stress" and press reports are showing this to be an increasing problem (ref: Times August 28 2001 "Are we working ourselves to death?"). It is strange to think that after the Industrial Revolution in Britain laws were passed (the Factory Acts 1819 to 1850) restricting the hours worked by factory workers and less than 200 years later many workers in IT exceed those hours, as well as current EC directives on working hours.

In order to understand how late you are going to be, or how much harder you need to work, graphical representations of progress can help. In the graph below, we can see by comparing estimated and actual progress that we are behind. We have a choice - to increase our work rate in the next month, or to continue at the same rate and deliver late.



Which is the preferable option? Discussion of the risks of being late compared with the possibility of working harder will give the answer.

- ⊙ One team used focus days to reduce the backlog of testing. On 3 days of each week, no one carried out any activity or answered any emails that were not directly focused on the testing. The team leader protected the team from administrative and other work on those days. At the end of the day the results were displayed on a whiteboard so that the team knew how much ground they had covered against target.

Adding resource may not help the problem. In Brookes' book "The Mythical Man Month" he describes how adding additional people to his overstretched team on a late project made the project even later and analyses the reasons why adding people makes it late:

- ⊙ The tasks are not partitionable - "The bearing of a child takes nine months however many women are assigned"
- ⊙ Training is needed - the new people have to be inducted onto the project
- ⊙ Communication increases with more people so more time is spent on communicating

Additionally, poor hand-over (for example poor specification of the work) may slow down progress, and an increase in team size increases communication requirements (see Brookes, Mythical Man Month for an excellent description of how adding resource to a late project makes it even later). In the long term, better specifications of work will aid handover to new people on the team.

In the long term estimating must be honest and reflect the work required. Brookes (Mythical Man Month again) talks about "Gutless estimating" - we estimate to please the listener not to reflect the work required.

Does setting targets for improvement help?

Yes: focus the targets and the areas of improvement - whole team understands the focus

No: if the targets are impossible or if no measurement of current position

Possibly try: (Crude) measure of current position, independent audit/review/spot check to assess best area for improvement, set focused target in one or two areas, look where you can give ground (tactics)

For the future: set improvement targets for the organisation as a whole and for individual projects, consider whether you want a world record or not, always ask "Why?"

In any project it is vital to set goals with targets and indicators which reflect the real purpose and the risks of the project. Without these it is easy for projects to get off track. Within the troubled project, when an improvement drive is being used to resolve problems, the same rules apply - if it is not clear what improvement goal is set, people will each try to apply their own improvement. To help you set the targets, use the Weaver triangle below, with the Gilb arrow (see I Evans EuroSTAR99).

The triangle asks to document your reasons (aims) for the improvement - the real difference you want to make. The objectives are the things you will do in order to make the difference. Indicators are how you measure that a difference has been made. Targets show when, how many and so on - the measurable evidence you have done something. They must be clear, agreed, measurable, and achievable.

- ⊙ One on-going maintenance project set its targets as:
 - "Reduce cost by 10% in the next 18 months"
 - "Reduce infield faults by 5% in the next 6 months"
 - "Reduce rework by 15% in the next 2 years"

An example of an unachievable target was set by the managing director who said:

"I want the whole organisation at CMM level 5 in 6 weeks"
and
"in field faults must be reduced - at once!"



Unfortunately, not only was the organisation at level 1, with no processes in place, there were no measures of failures in the field or in software testing, so there was no basis for setting an improvement target for in-field faults. The first tasks in improvement were

1. a holding exercise on some critical areas with focus on testing those areas
2. a measurement exercise (crude and simple at first) to understand the current problems.

We developed some aims, objectives and targets for one of the on-going maintenance projects to help in immediate improvement, for example:

What is your Aim? Why are you doing this project? What is your reason?	<i>"To improve the delivery of software to customers, making the delivery more effective and efficient"</i>
What are your Specific Aims? Specifically why are you doing it? List the reasons. List the things that will make a difference.	<i>"... to increase the knowledge of our test groups in best practice" "... to have a more efficient delivery process" "... to improve our understanding and management of the risks associated with the ABC sector ..." "... to maximise profit on our XYZ range of products... " "... to increase our customer share in the ABC sector... "</i>
What are your Objectives? What will you do to meet the aim? What actions will you take? Every specific aim needs an objective, every objective must have a specific aim.	<i>Objective O1: running a risk assessment programme covering ABC and comparing it with the risks of the DEF sector. Objective O2: presenting awareness seminars to all testing staff Objective O3: training all testing staff in the standard review technique</i>
What are your Targets? - to help you measure progress of set targets against each objective.	<i>Initial benchmark measure complete by 1/4/99 All testing staff completed a foundation level training course by 31/8/99 All developers received training based on BS7925-2 techniques by 31/12/99</i>
What are your Indicators? How will you know the project has succeeded? Each aim should have an indicator - something that will show you whether your project has made a difference. The indicators may or may not have targets set against them.	<i>increased satisfaction of customer increase in compliments to customer support increase in profit during the next 2 years developer/ tester relationships improved reduced number of problems reported by customers reduced on-site support time.</i>

Does introducing standards/methodologies help?

Yes: speed it up, increase quality, better methods, less rework, one way of working, re-use, clear communication
No: slow it down, cost more - learning curve, degree of ceremony, counting angels on the head of a pin, rigidity

Possibly try: test strategy, test design methods, key headings in EQM for guidance, key decision points, entry and exit criteria

For the future: internal/external standards, use as guideline or rulebook, overall process tailor to projects by risk / size, changing needs with standards

Note: I will use “standards” as a short hand for standards, methodology, industry best practice, process. Standards might include those written outside the organisation (ISO, IEEE, etc.) and those designed and applied within the organisation.

Standards and best practice are great in theory but when a troubled project is midway through it can be difficult to introduce them without slowing progress. It is important to focus on quick wins. This might include picking out the essence or a key point of the standard, for example:

- ⊙ one of the things BS7925-2 helps you to do is *understand test coverage*
- ⊙ one of the things ISO9000 focuses on is *customer-supplier relationships*
- ⊙ one of the things the European Quality Model looks at is *vision and leadership*

If one of these seems to be your projects problem a mini-audit or mini-review on that topic alone may help you understand if it is a problem and where it needs to be addressed.

It is only too easy to reject the possibility of standards, either because “we have always done it this way” or because we believe that the failure to apply standards to date in the project means we cannot start applying standards to testing. Responses might be:

- ⊙ “we cannot test because there are no requirements / acceptance criteria / we did not use the V-model”.

*Scene: a small country lane
Motorist (lost) to Farmer: How do I get to Little Gidding?
Farmer: Well I wouldn't start from here!”*

It is no good - in the words of the old joke saying “I would not start from here!” The project requires a more practical and more pragmatic approach. The start of user acceptance testing is too late to say, “You should have used the V model”, but it is not too late to:

- ⊙ make a spot check on requirements against tests developed and run so far
- ⊙ derive acceptance criteria and overall requirements if none exist, the tester working with the customer, user, analyst and developer - not ideal, but a pragmatic approach
- ⊙ use mini-audits and reviews on previous testing
- ⊙ draw up test checklists.

Used in this way at stage in the project you can cherry pick the most appropriate parts of standards and apply to particular problems. Long term, of course, one of the benefits of standards is standardisation - to get the benefit you would need to look at applying standards consistently across work.

Do the standards slow or speed a project? As noted in the summary at the start of the section, misapplied standards will slow you down. However the well applied, appropriate standard will bring benefits in reduced rework and increased re-use. Use of standard terminology (for example BS7925-1) means that communication is quicker and clearer. Fewer misunderstandings mean fewer mistakes in the specifications and in the testing. It is certainly worth stating a project vocabulary - even if it is some way through the project.

It is important to choose or apply the standard in an appropriate manner. In a low risk project or business, or part of the project, we may choose to have a different degree of formality or of ceremony - these are two different things:

Marriages:

- ⊙ in a church, high degree of ceremony and also is formal (recognised by the law)
- ⊙ in a registry office, low degree of ceremony but is still formal (recognised by the law)
- ⊙ in a stone circle, high degree of ceremony, but is not formal (not recognised by the law)

So, we may want to use standards in as far as we want to have exit criteria from stages which include sign-off, and we want to say who is allowed to perform the sign-off, but we may want to have a simple sign-off (email?) or ceremonial sign-off (major meeting?). The formality is important. The ceremony is only important to mark an occasion, for example in a public or celebratory way.

Be careful that standards do not descend into rituals carried out when no one understands the meaning or reason for the actions being taken. Examine any standards you are using - why are you using them? What is the reason for them? What benefit do they bring?

- ⊙ An example from one project was to combine two of the standard documents required by the standard life cycle as much of the information was repeated - in most projects the two documents had different audiences, but in this project one person had taken two roles and therefore was the audience for both documents.
- ⊙ On another project, the testers added to the standards by setting entry criteria to the system test stage; entry criteria had not been used before. They were set with a low degree of ceremony, but were formal. The developers were required to provide evidence of particular systems areas exercised during component test.

Another possible disadvantage of standards is that they can induce pedantry and a tendency towards "medieval theology". Medieval theologians would argue about whether Adam had a navel or how many angels could dance on the head of a pin. Neither of these points gets to the heart of what is important about living a good and moral life.

Genuine query to IE "should I sign the release form on the bottom left or bottom right?"

IE reply "I do not care!"

Use the standards as a rulebook if you want external recognition, to be auditable and to have standardisation.

The SS Great Britain was the world's first large iron ship, and is built of iron plates welded together. Every part of the hold is a different size and shape - each was individually hammered to fit (BBC programme "What did the Victorians do for us" Adam Hart-Davis).

Query: is this suitable for a modern project or would a hull design with many identical pieces that can be made separately be more efficient? Or, is software changing so quickly that we need the SS Great Britain approach - and our standards apply to how we estimate and control the work?

Standards may be also used as a guideline rather than a rulebook. They are a tool-set of best practice and good ideas. Remember that times change but the principles stay the same. If you are interested in the spirit of a standard rather than the letter, you can look at what the standard is intended to help with or prevent and decide how you want to do it. Be sensible. Choose levels of formality and ceremony that are appropriate to your risk.

For future projects you may consider having an overall process and standards, which may be tailored by the individual projects to reflect their risks.

Example: On one project we disregarded the maintainability standards as the software was for one-off use, and concentrated on audit trails and reconciliation standards. This focused both development and testing effort.

Remember that standards are written by human beings - so may well have mistakes, wrong assumptions and bias in them! That's why they have drafts and then may have new versions! Use your intelligence and common sense, but do not discard them - instead - get involved in the standards working parties.

Some final thoughts about when to use and when to discard the processes:

Foot and mouth - The army office in charge saw a shortage of phone lines and diggers, so rang Yellow Pages and just ordered more ... he was confident that funding would be made. In contrast civil servants at the Ministry of Agriculture, when informed that the epidemic was out of control and that many more animals would have to be slaughtered, worried about where the money was to come from... (Times July 2001)

Foot and mouth - "At an early stage in the foot-and-mouth crisis, MAFF instituted a policy of slaughtering infected animals within 24 hours of diagnosis. At the same time it sent to all vets a flow chart setting the procedures to be gone through prior to slaughter and these procedures couldn't possibly take less than three days" (Times July 2001)

Does training help?

Yes: if focused, targeted, improvement indicators understood, prepared for

No: badly focused, delegates do not know why they are there, wrong people on training, too late, too theoretical

Possibly try: focused workshops - 1 new idea, work on project specific problem e.g. design tests, identify risks that testing can address, identify low risk areas

For the future: process rollout, professional training (theory / practical / industry best practice), workshops (business specific areas, awareness sessions)

Training is used for many purposes:

- ⊙ To ready someone for a specific or their current work
- ⊙ To prepare them for their next career development
- ⊙ To introduce new processes

Less usefully I have seen training used to:

- ⊙ "cheer them up" (a sweetener) - staff have been asking for training or generally unsatisfied and going away for a jolly seems like a quick fix.
- ⊙ To say that training has been done - the manager has to get some training done

If training is to work it needs to be prepared for properly - whether it is an in-house or external, provided by trainers, or an in-project workshop.

Real life examples that have worked are:

*Mini rollout of new processes followed by training
Train managers first then cascade down
Ask for volunteers for the training
Workshops on current specs - i.e. workshop to produce a set of tests using introduced techniques rather than course using external examples*

Real life examples where the training has not had as good an effect as supposed include:

*People not told why the training is happening
People not told their team going on the training
Wrong people on training course*

To get the most out of workshop, training courses and other sessions, especially a workshop for a troubled project:

- ⊙ Understand the aims and objectives of the training (use a Weaver Triangle as described earlier)
- ⊙ Set targets and indicators to show that the aims and objectives have been met
- ⊙ Discuss the training with managers and team leaders - their people will be away from project work during the workshop - ask them for their aims and objectives
- ⊙ Select delegates carefully - who needs this session and who can contribute?
- ⊙ Discuss the training with the delegates beforehand - let them know why they are going. Or, advertise the training workshop, its aims and objectives and ask for volunteers.
- ⊙ Finish the workshop with action plans - individual and/or group
- ⊙ Follow up the workshop - show that the changes matter to you

DOES GETTING A CONSULTANT / EXPERT'S VIEW HELP?

Yes: independent view, expertise you may not have, challenge your own experts' assumptions, facilitation, clarify mixed motivation and conflicting goals, easier to deliver messages, new ideas

No: outsider not welcomed, team not informed, cannot accept the idea of change

Possibly try: audit / review / spot checks, interpretation / introduction of standards, expert led workshops, mini-project in specific area

For the future: ditto, plus use as team members

Consultants and outsiders bring a new view to your troubled project - they may well have tackled just the problem you are struggling with at the moment and resolved it successfully. They can challenge your preconceptions about the problems and what would make a solution. Where your troubled project is suffering from political problems an outsider may help by facilitating between factions, clarifying the goals of the project and acting as a communication bridge.

The types of things that go wrong with consultants in the troubled project is that the managers or team may:

- ⊙ Feel they have had an outsider inflicted on them - some one who just does not understand what is going on in this project

"what do you know about the pressure we are under?"

- ⊙ Not be told why the consultant is there - causing suspicion and perhaps de-motivating staff

"Consultants kept appearing and having meetings in closed rooms - we were never told why they were there and it never changed anything for us"

- ⊙ An unwillingness to consider change

"We always do it this way"

Project problems where outside consultants can help include:

- ⊙ audit / review / spot checks independently either against standards or best practice, mainly with the view of offering advice rather than a non-conformance list

for example review of test specifications against functional specifications to look at coverage showed that some test areas were repetitive and others not covered

- ⊙ interpretation / introduction of standards

for example use of parts of BS7925-2 to show that component test can be documented and that this does not have to be wordy documents.

- ⊙ expert led workshops

for example, test design workshop with formal techniques introduced and then applied to a real specification; test scope workshop to explain risk assessment and fault prediction, followed by analysis of the functional specifications and listing them in risk order, with a predicted fault density for each specification

- ⊙ mini-project in specific area

in one project a fault database with automated reporting was required. An external person was brought in specifically to build and test the fault database to the requirements of the test team

Does de-scoping help?

Yes: reduce work load, clear plan of what is out of scope and why, target high risk and essentials, set expectations, reduce testing in low risk areas

No: reducing testing rather than the scope

Possibly try: risk assessment / fault predictions to check focus of development / testing, discuss scope/work loads with customer / end user

For the future: set clearer acceptance criteria, customer/user review of specifications, entry and exit criteria, product reviews

Descoping the project has several parts:

1. Identify and agree to remove things (tasks or deliverables or processes or administration) that are not part of the agreed scope - these are the "scope-creep" or "scope-slip" items. Unfortunately they may have become part of the customer's expectations so removing them does have to be managed
2. Identify and agree to remove things (tasks or deliverables or processes or administration) which, even if in the agreed scope, do not directly help the main aim of the project. To help, spend a short time doing a Weaver triangle for aims and objectives with main stakeholders for the project or this part of the project.
3. Identify with the customer/project sponsor whether time, budget, scope or quality is the main driver. Obtain written agreement that the other three can be compromised!
4. Identify and agree to remove things (tasks or deliverables or processes or administration) which although they aid the real purpose of the project are not associated with high risks. To help, spend a short time carrying out a risk assessment with technical and business.
5. Carry out a requirements review and identify requirements that can be delayed or removed. To help with this, revisit or carry out a cost-benefit analysis, with technical and business.
6. Identify administrative tasks that the project office function can perform for the technical and business members of the team.

Only reduce testing carefully - you will need to be sure of one of the following for example:

- ⊙ reducing the scope of the delivery - i.e. less to test
- ⊙ the test coverage was excessive against the risk and you are removing low risk tests
- ⊙ the tests are repetitive and you are removing repeat tests
- ⊙ fault predictions indicates low predictions for areas you choose to remove

Always make sure you have clearly documented and got agreement for what testing is out of scope and the increased risk.

Finally, some quick ideas for tackling problems:

1. revisit the test strategy

Typical problem: No clear agreed strategy for testing

Good practice: Entry and exit criteria, Acceptance criteria, Risks assessed in the context of testing

Real life example: Strategy written but never publicised, Not issued to team, Conflict for environments and test data, No clear completion criteria

For immediate results: Set entry/exit criteria, re-assess risks (what can testing reduce, what is aggravated by testing?)

For the future: test strategy skeleton, awareness of test strategy

It is certainly worth revisiting the test strategy. Review in particular the entry and exit criteria, to ensure clean handover at each stage. Even late in the project it is worth setting entry criteria for acceptance testing. Reassessing the scope of testing, especially risk assessment for each functional and non-functional area can help to change the focus for testing to the most important areas. These risks may have changed during the project.

In one project a decision was made to change the focus of testing from system test to component test in order to satisfy the testers that code build and change control was sufficient. This reduced system test, but the failures in live were low. In another project, introduction of improved hardware reduced the risk to performance and capacity and the project refocused to increase functional testing.

Also assess risks to the testing - for example if the project is late it may hold the test environment and impact on another project. Discuss the needs and strategy for your project with other projects to ensure you avoid conflict, for example by calling a cross project environments meeting.

2. revisit the test design

Typical problem: Unable to tell if sufficient testing has been done

Good practice: use of formal design techniques, coverage measures, completion criteria, test material is reviewed

Real life example: Poor coverage - too little or too much testing, repetitive tests

For immediate results: Audit test coverage (sample), teach techniques, review specifications against tests, cherry-pick to apply techniques

For the future: train testers, analysts and developers in test design methods

An audit or review of the test specifications against the requirements is useful and can be done relatively quickly, particularly by an expert. Sample rather than attempting to cover everything, and use the sample to gain a view of whether you have a problem and how big the problem is.

One audit of test material showed that not all the functional specifications had test specifications defined.

Pick out high risk areas (see Test Strategy above) and give workshops in test design techniques - look at coverage and focus on one or two techniques.

In another workshop, examination of a test specification showed that the 200 test cases were not good coverage. The number of tests by combination of inputs was in excess of 7000 million. Using equivalence partitioning very strictly against the specification gave 10 test cases. Use of cause-effect and error guessing got to 40 test cases, and this set was adopted saving time while covering the risk.

3. revisit test documentation

Typical problem: Key information not available

Good practice: document management system

Real life example: too wordy, repetitive, information in several documents, directory difficult to navigate, no version control,

For immediate results: audit/review/spot check on key documents, set minimal (at least) document management standards, naming conventions

For the future: set naming conventions, directory structure, management standards, use document reviews, use diagrams

Minimal document management systems include a set directory structure, so that test material is kept in a common rather than personal directories. Audit and review will quickly point out areas where documents are not readily available to the project, as opposed to individuals.

Naming conventions must be set, even if they are not perfect. The possibility of misunderstanding is increased many-fold.

In one project the business users and the developers were from three companies in a variety of locations, sharing information. Each group had its own naming convention for key information - and these conflicted, causing confusion and mis-specification. A naming convention was defined and enforced, enabling clear communication of requirements.

4. revisit measurement/reporting

Typical problem: unable to state current position or likely completion

Good practice: daily update, public progress graph, days to go chart

Real life example: Wrong target for test case progress, takes 2 days to produce statistics

For immediate results: Team need this now: face / publicise the scary bit - measure current status, display publicly, admin support / simple automation

For the future: document testing so that it is easy to collect statistics, automate statistics / test management, make accessible to all the team

Useful quick reporting ideas that work are:

White board daily update - number of tests still to run still to pass, number of faults outstanding

Graph of progress to end date zero - updating weekly

"Days to go until live" tear-off chart

Measurement and reporting only works if the right numbers are used and the measurements can be taken and publicised quickly. Real life examples of bad measurement are:

Wrong numbers of tests reported on - progress looked better than it was....

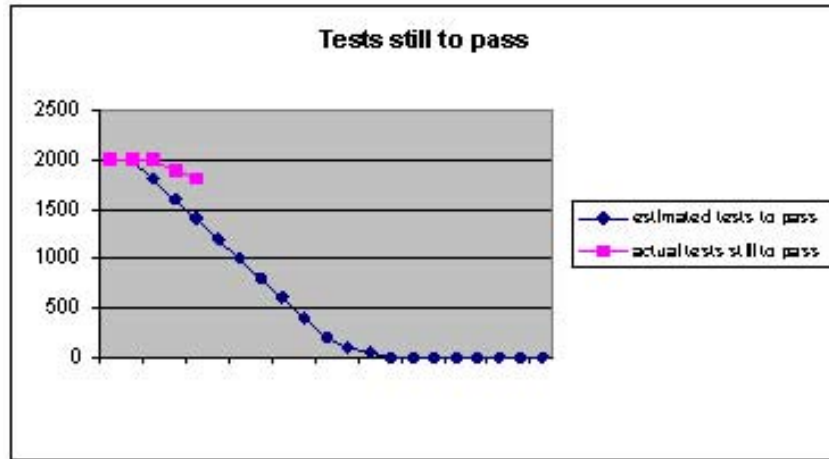
Takes too long to produce statistics (2 days to produce a test progress report by hand!)

For immediate results note that for progress, the team are more important than the project board... they need to know statistics and progress now, in order that they can gauge progress on their own plans. The macro-picture for the project board may be needed at longer time intervals - for example you may report to the project board at fortnightly intervals, but the statistics that are of use to the team are needed on a daily basis.

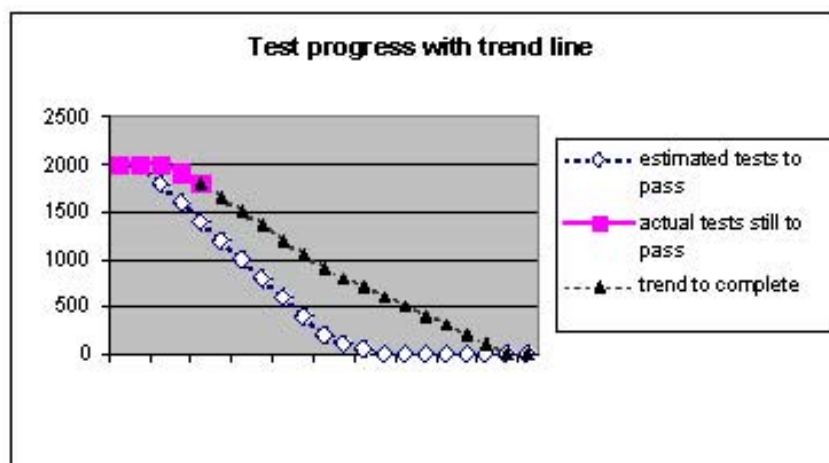
Display the statistics publicly, make sure they mean something to the whole team and update them frequently so that they are worth looking at. Face and publicise the scary bits for example number of days to go until test stages and live.

Automate statistics production as far as you can. In an on-going project, this may be limited, in which case use administration or project office support. In the example above we reduced the statistic report production from 2 days to 2 hours for the weekly report by using spreadsheets more effectively, and training the tester team leaders in how to use the spreadsheet graphics. We also asked for Access programming support to help produce fault reports and statistics more quickly from the Access fault database. A relatively small amount of effort paid immediate dividends in freeing up testers to test, while providing better information to the team and project board.

The types of graph that you may find useful are those that enable you to understand whether you are in trouble and how much trouble you are in. Look at predicting progress through testing and number of faults, then tracking against those predictions.



In this graph, the actual tests to pass have fallen behind predicted progress. We can add a trend line to help us understand how late we will be if we carry on at this rate:



This trend line was added manually by putting another line in the spreadsheet. You can add a trend-line automatically by right clicking on the line and selecting "TRENDLINE".

Finally think about when and how you give bad news or admit you do not know the answer:
Foot and mouth: Minister asked whether the outbreak was under control said Yes. Later when questioned he insisted he had been telling the truth as he saw it... "But just imagine if I'd said "I'm not sure" or "I don't know" or "No maybe its out of control, its running amok throughout the country" Do you not think that might have caused some alarm, and lack of confidence in the government's disease-control strategy?"

Query: does this response give you confidence? Did the minister make the right decision or not? What is your view?

Sources and references:

"Foot and Mouth"	- television review by Robert Hanks in Times 4 July 2001
"Triumph"	- article in the Guardian by AC Grayling (date not known)
"Are we working ourselves to death?"	- article in Time 28 August 2001 by Anjana Ahuja
"Mythical Man Month"	- ISBN 0-201-83595-9 by Brookes
"Goals Risks and Testing"	- presentation to EuroSTAR99 by Isabel Evans
"Nineteenth Century Britain"	- ISBN 0 582 31 49 84 by Wood
"Hitchhiker's Guide to the Galaxy"	- BBC radio series and book, by Douglas Adams

Websites:

See www.ietesting.co.uk under "useful sites" to see a list of source information sites including:

For testing standards see www.testingstandards.co.uk

For European Quality Model see www.efqm.org

For British Quality Foundation see www.quality-foundation.co.uk

For the work of W Edwards Deming see www.deming.org

Also thank you to my clients and colleagues for projects and anecdotes.

As the late Douglas Adams said:

"Don't Panic!"

(Hitchhiker's guide to the galaxy)