



EuroSTAR99 November 8-12 Barcelona Spain

Goals, Risks and Testing: Managing Projects against a Fixed Deadline

Isabel Evans

IE Testing Consultancy Ltd

Email: isabel@ietesting.co.uk

Web Site: <http://www.ietesting.co.uk>

Abstract

"To focus testing we need to understand goals and risks".

It is easy to say this - but how can we do it?

This paper discusses the success factors and lessons learned on a fixed deadline and high profile project. "Hard decisions were made throughout the project to meet the fixed deadline by focusing on minimising risk to essential business processes." The paper covers the main points raised in the evaluation of this project as well as improvement ideas used in subsequent projects.

- Key success factors from a real project - what really helped and why
- How to clarify the real aims, objectives and risks of a project
- How to focus testing where it is most needed

Copyright: © this paper is copyright to the author

Acknowledgements: the ideas are generally other people's ideas adapted by the author. See the references at the back of the paper.

Contents

Goals, risks and testing: Managing projects against a fixed deadline	3
<i>Summary</i>	3
<i>Biography</i>	3
A. Introduction	4
B. A real project evaluated	5
<i>B1 About our project.....</i>	5
<i>B2 Immediate evaluation – “How did we do?”</i>	6
<i>B3 Two month evaluation results – the key success factors.....</i>	7
C. Improved ways to clarify the real aims, objectives and risks of a project	16
<i>C1 What was different about the next project?.....</i>	16
<i>C2 Risks – improvement ideas</i>	17
<i>C3 Goals, aims, objectives – improvement ideas.....</i>	21
D. How to focus testing where it is most needed	32
<i>D1 Understand goals and risks.....</i>	32
<i>D2 Use specialists.....</i>	32
E. Finally... ..	35
<i>Good luck for your next big project!.....</i>	35
<i>Approaching the next big project.....</i>	36
References and further reading.....	37
<i>Work referenced in this paper.....</i>	37
<i>Other Publications and Sources</i>	38

Figures

Figure 1 - Risk: <i>Post-it</i> approach.....	17
Figure 2 - Risk: In your garden	17
Figure 3 - Risk: SWOT charts	18
Figure 4 - Risk: Know what you know - know what you do not know.	18
Figure 5 - Risk: Hans Schaefer's spreadsheet – fictional example of use	20
Figure 6 - Jayne Weaver's project planning triangle	22
Figure 7 - Targets defined move clearly (adapted from Gilb).....	23
Figure 8 - Goals and targets - triangle and arrow combined	23
Figure 9 - Multiple aims and objectives.....	23
Figure 10 - Aims: why are you doing this project?	25
Figure 11 - Specific aims: Specifically why are you doing this project?	26
Figure 12 - Objectives: What are you going to do to achieve your aim?.....	27
Figure 13 - Targets: How will you measure progress?.....	28
Figure 14 - Indicators: How will you know the project has made a difference?.....	29
Figure 15 - Risks Goals Testing.....	32

Tables

Table 1 - Examples of failures and risk evaluation comments	10
Table 2 - Project aims.....	25
Table 3 - Project specific aims	26
Table 4 - Project objectives.....	27
Table 5 - Project targets	28
Table 6 - Project indicators	29
Table 7 - Project aims and objectives exercise.....	30
Table 8 - Project aims and objectives exercise continued	31

Goals, risks and testing: Managing projects against a fixed deadline



Isabel Evans - IE Testing Consultancy Ltd. - England

"Common Sense Quality"

Summary

Isabel Evans describes the success factors and lessons learned on a fixed deadline and high profile project. "Hard decisions were made throughout the project to meet the fixed deadline by focusing on minimising risk to essential business processes." The paper covers the main points raised in the evaluation of this project as well as improvement ideas used in subsequent projects.

- Key success factors from a real project - what really helped and why
- How to clarify the real aims, objectives and risks of a project
- How to focus testing where it is most needed

Biography

Isabel Evans has over 15 years experience in the IT industry, mainly in quality management, training, testing and documentation. She has worked as an independent consultant for 7 years and is director of IE Testing Consultancy Limited. She is the co-author of TAP, the Test Assessment Programme.

Isabel 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 Quality Assurance Support, Release Management, and Customer Support for IT organisations. Isabel also provides training courses in quality management, project management, testing and documentation skills. Isabel Evans for IE Testing Consultancy Ltd is accredited to provide training for the BCS ISEB Foundation Certificate in Software Testing.

Industry sectors include Intelligent Systems, Relationship Marketing, Financial (Insurance and Banking, Stock Market Quote systems), Retail, International Shipping, Construction, Oil, Utilities.

A. Introduction

In this paper, I will describe success factors and lessons learned on a fixed deadline and high profile project which completed at the end of 1997. I was the test manager for the project. It was a critical project for the business, both for the client organisation and for the immediate customer, the IT services supplier. The lessons from this project have been useful for other time-critical projects.

Hard decisions were made throughout the project to meet the fixed deadline by focusing on minimising risk to essential business processes. Sometimes the "best solution" was an IT solution, but sometimes the risk was contained most effectively by choosing a non-IT containment/contingency plan.

This paper will discuss the main points raised in the evaluation of the project, sharing success factors as well as improvement ideas used in subsequent projects. The major areas covered in this paper are goals and risks, but I will also briefly reflect on people and relationships, project management, use of team specialists and work patterns.

In evaluating the project we identified reasons for our success but we also identified "things to do better next time...". I will cover successful methods and the ideas for improvement adopted in later work. The paper will concentrate on test management aspects of the project, rather than technical aspects of testing.

B. A real project evaluated

B1 About our project

The project was a major financial services conversion project, converting unit trust and PEP (Personal Equity Plan) holdings into OEICs (Open Ended Investment Companies). It was the second largest OEIC conversion done to that date. We converted £900,000,000 in funds for 30,000 customers.

The funds were moved between registrars and holding companies. Additionally there was a physical location transfer; we had to transfer the data from one computer centre to another computer centre across the UK.

The work had to be completed to a strict deadline, in order to meet the OEIC launch date. The financial value and the investments per client, per fund and in total had to be correct.

In order to manage the time scale, quality and budget we had to concentrate on preventing scope slippage.

We transferred the funds and customers to the new locations and companies. We finished on time. We were able to start the conversion and transfer at end of business on Friday 7th November and on the morning of Sunday November 9th we walked out of the new computer centre with the funds transferred and the customer organisation satisfied that it could open for business on Monday morning.

"It was the morning of November 9th - Remembrance Sunday in the UK - as we walked out of the customer building with a sign off as complete and the system ready to go live Monday morning. Returning home, I called into the local pub, very pleased with what we had achieved. The pub was full of war veterans - a timely reminder of what risk and team work under pressure really entails." – Isabel recalling the end of the project.

For the next two weeks we stayed on standby to deal with any teething problems. Gradually the team was "stood down" as the customer users' confidence in the transferred data increased.

During this time we carried out an immediate evaluation to satisfy ourselves that the project had completed correctly.

B2 Immediate evaluation – “How did we do?”

We were successful when measured against the two critical success factors - we finished on time and we transferred the funds correctly and completely.

The project was extremely demanding - and it was successful against those two success criteria because the team worked very long hours, the Project Manger absolutely refused to allow scope slippage, and the Customer Manager allowed the budget for effort to expand "flexibly".

"Don't tell me you're going to be late until everyone on the team is working a 60 hour week" - Customer Manager refusing to consider a change to the deadline - a comment ill-received by those who had just worked an 80 hour week.

Certainly the application of a tight time-scale and consequently a tighter grip on scope slippage than normal has had a positive effect on many projects recently. The idea that scope slippage and requirements change could be kept in check was a useful lesson to take on to later work:

"The tight time scales were refreshing - we could not allow ourselves to be diverted" - Project Manager on a Y2K project

Team members then moved on to other projects and it was some weeks later that we carried out a formal evaluation.

When is the best time to evaluate the success of a project?

Immediately after the project is completed the evaluation will often be of success. Particularly for a difficult or problematic project the immediate evaluation is success BECAUSE you have stopped – the agony is over...

It can be better to leave the evaluation until later. A project manager I worked for many years ago suggested that the end of project party should be held 6 months after the software went live – only then could the success of the project be judged.

Provided you have kept a log of all the problems that arose during the project and the implementation you need not worry about forgetting things. Your evaluation will benefit from a more neutral and considered view.

The compromise of an immediate evaluation followed by another evaluation later on helped both of us put problems into perspective.

B3 Two month evaluation results – the key success factors

After two months I met the project manager to re-evaluate the project and to decide what really helped and where we could see improvements for later projects. We evaluated against some of the areas which helped us to contain risk.

We had an outline risk strategy and we believed we understood what to do about risk:

- assess the risk
- plan what to do about the risk
- implement the risk management plan
- monitor and manage the risk

During the project we identified risk areas. A selection is listed below:

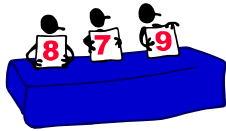
- The business profile of the project: The project was very high profile giving a high business risk associated with failure to meet tight time-scales which had been set by the customer and specifically in marketing and publicity schedules.
- Communication between groups: There were many groups in different physical locations across the UK, including four customer-business organisations, two computer centres and their operations groups, three IT development and test locations, two sets of internal audit and compliance groups, external auditors, the registrars, the executive, marketing and management teams in each participating organisation, several customer organisations for the IT teams and of course the 30,000 customers for the financial services.
- Capturing business and technical information: there were many sources of information for the project.
- Each group brought different company cultural backgrounds with particular expectations of reporting structure, communication styles, planning, problem solving...
- Standards and practices: There were different standards, practices and tool sets in the IT groups. Even the language and terminology used was different:

"My integration test is your link test and their string test, but your integration test is my acceptance test..."

- Platforms: There were many hardware and software platforms for the source and target systems. There were three source systems, one major target system and several minor target systems.

Note: we identified other technical risks but I will not cover those in this paper.

Risks and failures - how did we do?



When we evaluated we decided that there were several things which had particularly contributed to containing risk:

- good understanding of the real goals of the project
- tight project management
- no widening of scope
- good risk management throughout the project
- good relationships across teams and within the team
- early involvement of specialists
- hard graft and long hours were off-set with treats

We understood the project risks and we monitored throughout the project, including:

- Daily logging of any problems (test incidents, management issues, anything that prevented project progress)
- Logging risk history over time
- Reporting on and learning from reality
- Assignment of a special Risk Assessment Role on the major reviews

In evaluation we decided that our risk assessment and the team's assessment of issues as they arose could have been improved. We agreed in evaluating both immediately and after two months that the risks and priorities for the project were not clearly agreed. Our project came into conflict with other projects and our team with other areas of the same project with regard to scope and responsibility.

Example: there was competition for the system test environments between our project and other projects. Because there was not a common understanding across all parties of the risks and priorities across projects, there was conflict between the projects.

Our lessons learned were that on a very large multi team project it is even more important to communicate well. How could we do this?

On a later project we ran training sessions for the whole team to discuss risk and priority setting. This included training on the causes of risk and on the assessment:

As a team, we needed to make sure everyone had got a common definition of risk:

- *Anything that threatens the probability of succeeding in this project:*

- *something that threatens one or more of the project aims and objectives*
- *something that threatens one or more of the project success criteria*

As a team we needed to have a common understanding of "where risk comes from" - we needed to clarify to everyone across the project the sources of risk and an overall containment plan (assessment of own organisation, customer, supplier, 3rd party, government, legislation, press, employees, human error, commercial pressure, technical pressure, communication, domino effect, poor management of...)

- *know the project objectives and success criteria*
- *honestly identify risks / threats / weaknesses*
- *identify the likelihood of each risk*
- *identify the impact of each risk*
- *score the risks*

When we evaluated we thought about the causes of failure and whether any of those had affected us. The table below shows the types of problem we raised and how we started to think about resolving the problems, together with a comment on later actions I have taken.

Later we will look at improvement ideas we had and have applied in later projects with success.

Table 1 - Examples of failures and risk evaluation comments

Possible risk cause of failure	Did it affect us?	Next time we would...
Misunderstood aim and objectives (it was the wrong project)	No - for our team But other teams had conflicting aims and targets – example – conflict for use of system test area with other projects	Find a way to clarify the aims and objectives for ourselves and with other teams where there were interfaces or possible impacts.

In later projects Isabel has used the Project Triangle method described later in this paper to improve understanding and communication of aims and objectives.

Impossible target (unfeasible deadline or deliverable)	Nearly... But individuals came to the rescue... Is it reasonable to rely on a few good hearted workaholics?	Next project was also to a tight deadline ...How could this be managed? Can we find a way to help clarify targets with sales, the customers, the teams?
--	---	--

In a later project with some of the same team, a customer set even more challenging deadlines - use of the Project Triangle method helped clarify whether these targets were agreed and possible.

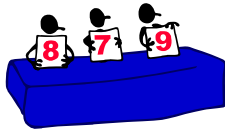
Poor communication (people did not understand each other)	Sometimes... We wrote a test glossary as part of the test strategy document, but we still had occasional misunderstandings with other teams	Clarify vocabulary earlier
---	--	----------------------------

The arrival of BS 7925-1 Test Vocabulary has helped to some extent - it has been easier to get teams from different backgrounds to agree to common terminology.

Late QM (e.g. poor / no / late testing)	No - before the TM arrived the PM ensured that early static test / reviews took place. The test specialists helped in early testing. Testers were diverted from this during system test and the unit testing (retest/regression test of changes for bug fix) deteriorated.	Get test specialists involved as early as possible - and keep them working with the developers throughout all stages.
---	--	---

The arrival of BS7925-2 Component Test Standard has been useful for both testers and developers in helping to understand how to improve unit test. In particular I have used the example documents in the guidelines to show developers that documented tests need not be wordy.

Understanding goals - how did we do?



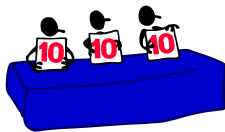
How well did we understand the goals of the project? We had a good understanding of the overall aims and objectives...

"everything goes"
"on the 6th November"
"finishing by 9th November"
"open for business on 10th November"

However, across the participating organisations other projects and other teams had their own priorities. How could we have helped others to understand our goals? How could we have understood earlier where goals might conflict?

We reflected on improvement ideas and these will be covered in a later part of the paper.

Project Management - how did we do?



How important is a good project manager and good project management? For this project the Project Manager and his style and techniques were critical in contributing to the success of the project. The Project Manager

- gave as much as he asked - he led from the front
- managed relationships and negotiated well
- had a clear view of the aims and objectives of the project
- focused on what had to be achieved today in order to achieve the deadline and monitored progress tightly
- asked each team member to become involved in planning their own tasks and gave backing to decisions
- focused on quality management and testing as a means to achieve success.

An example of this was the daily review of the problem/jobs log, with a daily posting of the Priority 1 list. As issues arose throughout the project, risk assessment was used to assess what to do.

- Sometimes the "best solution" was an IT solution – we changed and re-tested the conversion code.

- Sometimes it was best to do nothing – the problem was not a problem that would prevent the conversion completing
- Sometimes risk was contained most effectively by choosing a non IT containment/contingency plan.

The project manager, with the lead developer and myself as test manager met every morning before the rest of the team arrived to review what had to be achieved that day in order to meet the deadline. The three of us negotiated the priorities for outstanding problems by looking at:

- How important is this to achieving the aim of the project
- How difficult would an IT solution be
- Is there an acceptable non-IT solution
- Does it have to be done today?

For example the different source systems had different rules for managing name and address information, in one case the information being kept in free format, and gender was not mandatory. The target system had specific rules that the gender for each customer was mandatory.

It would of course have been interesting to write and test code to review the free form text in order to extract the title and name and hence deduce the gender information. However it was not a cost effective solution.

Instead, during the conversion, for any customer where a gender was not set, the gender was set to male, and if there was no title information the title was set to Mr. A report of all customers with the gender male was then printed post conversion load of the customer information from the target system. While we waited the 6 hours for the conversion load of the financial information to complete, we went through the report by hand marking any dubious entries. The customer sales representatives then checked these out during the following week.

Examples of gender-title-forename combinations:

<i>Rev. Hilary Smith</i>	<i>male</i>	<i>check gender</i>
<i>Major Evelyn Waugh</i>	<i>male</i>	<i>check gender</i>
<i>Mr Jane Jones</i>	<i>male</i>	<i>check title and gender</i>
<i>Mr James Evans</i>	<i>male</i>	<i>OK</i>

We kept a tight control of the scope.

For example, it was tempting as a tester to stray from testing the conversion itself into testing and reporting on defects in the target system. The target system was not our problem and not our responsibility. We had to hold back from the temptation to retest the target system.

The big lesson – from this project and from year 2000 projects is that we must be harder about how we deal with scope slippage: JUST SAY NO!

Customer comment in evaluation: “you were not open to making additional changes” PM responses: “that means I did my job well”

We did lose the tight grip slightly on an occasion at the end of the project. For next time we would improve our control of late changes – one late bug fix caused a configuration management problem.

“Don’t take your eye off the ball”

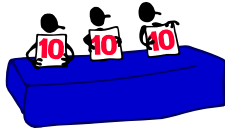
In later projects we tried to act on these lessons.

For example in containing the scope for a Year 2000 project we looked at limiting the scope. It was important not to get caught by the zero defects obsession. For year 2000 we want the system “warts and all” – existing bugs that the customers knew about and worked around were not up for change.

Another example was a later conversion implementation where the scope was limited by time and budget. We had to be very strict about not widening the scope. In the later conversion project we had to examine and re-cost changes with particular care.

However, a problem with controlling scope slippage tightly is that the scope of the project will change. Any project plan is just a guess about what is going to happen and how long it will take. What we want to control is the project moving away from its real goals and exceeding its constraints (for example time and budget) in an uncontrolled way.

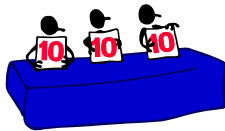
Relationships - how did we do?



How important are relationships? The good relationships within the team, with the customers and with the different IT teams contributed to the success of the project. We did not use social occasions to try to build bonds artificially. It was fortunate that the individuals in the teams generally worked well together.

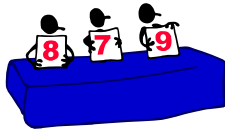
We made a big effort to keep a “No blame” culture. This can be difficult, but it did make a real difference when problems arose and needed resolving.

Team specialists - how did we do?



The impact of specialists on the success of the project cannot be under estimated. These included testers, performance experts, database experts, business experts from the source and target organisations. I will discuss the use of testers later, but to summarise the other areas:

- Performance and database experts – for example reduced the run time for the data load from 13 hours to 5 hours
- Business experts - for example resolved precision and accuracy decisions for rounding and truncation, defined source and target fund decisions



We knew at the start of the project that the deadline could only be achieved by working abnormal hours. How did we manage this with the team? We worked "Hard graft with treats". The examples below may look trivial, but they are neglected in many projects and they do make a real difference to the team.

For weekend, evening and overnight working, we tried to operate a "do something different to usual" policy. For example, the project manager would do some simple but tedious/time consuming programming tasks under supervision of the development team. As test manager, I would check results and log incidents for the test team. This helped the team by direct effort from the senior team members. The rest of the team either worked on their current tasks or used the weekend time to familiarise themselves with other areas of the work (understudy system) or to "catch up" with monitoring / management tasks or to raise issues with team leaders and managers.

We also made sure that we had "treats for lunch / supper". This does make a difference. Does anyone want to eat bad British pizza in the early hours of the morning? It was worth getting good food in, especially light, simple, easy to eat food such as grapes, good bread, good cheese.

We made the effort to celebrate objectives achieved during the project – at least by a pause to mark the achievements, with a thank you to those concerned.

We did have a concern, however, that the bulk of the long hours fell to a small group within the whole team. This was for a number of reasons:

- Willingness and interest
- Technical or other expertise

In an ideal project, one would be able to spread the load - there would be sufficient people with the technical and business expertise to allow coverage across several shifts.

On a later project we made a greater effort to appoint "deputies" early on and to allow time for shadowing and skills transfer from the project experts. This proved worthwhile in the later project when several key members came down with 'flu at the same time. The deputies were able to take over and run with the project.

C. Improved ways to clarify the real aims, objectives and risks of a project

C1 What was different about the next project?

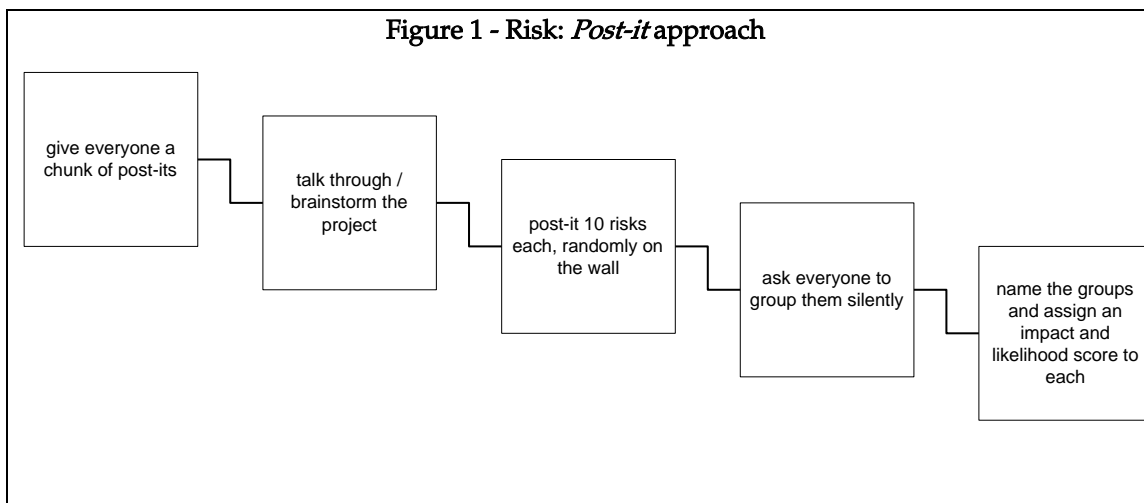
Evaluating this project and looking at my next and future projects, particularly Year 2000 projects, I could see the same set of issues and risks emerging. In fact all the projects I have worked on since have had at least some of the same characteristics - and often all of them.

For groups working on Internet, multi-national or e-commerce projects I suspect the list is the same:

- high profile project - high business risk associated with failure to meet tight time-scale
- many groups in different physical locations
- many sources of information
- many hardware and software platforms
- different company cultural backgrounds and no doubt for an increasing number of projects different country cultural backgrounds
- different standards, practices and tool sets in the IT groups.

C2 Risks – improvement ideas

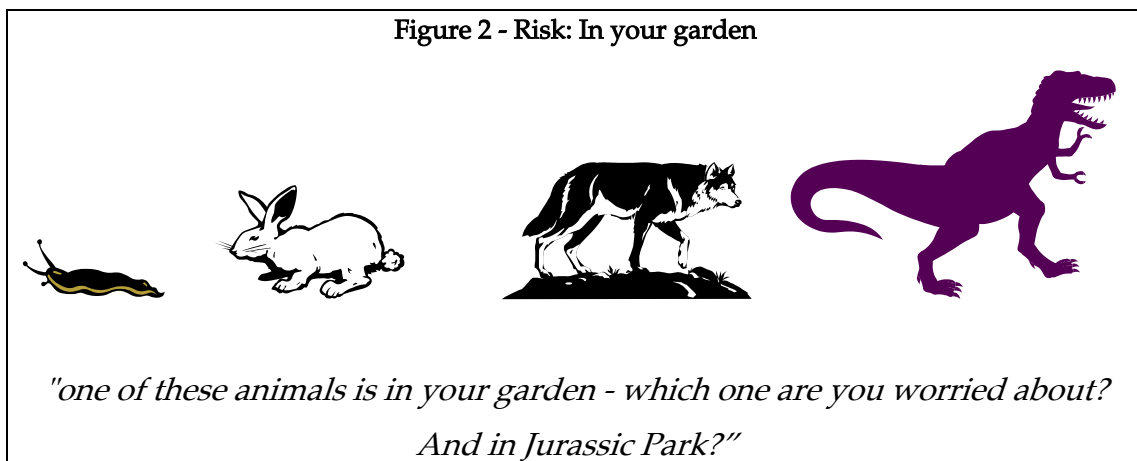
During the evaluation we discussed the improvements in risk assessment we had identified following the project – techniques that we would have applied had we known about them at the start of the project. We had used the Post-Its approach:



This helped, but both the project manager and myself had come across some other risk assessment methods which would have benefited the project.

In your garden?

When we ran the training sessions for the later project, I suggested an "In your garden?" model as a way to help teams distinguish between the impact and likelihood of a risk:



This has proved useful in projects and in training since 1997 in helping IT teams and customers assess risk.

Knowledge, strengths and weaknesses

The project manager suggested several models for looking at knowledge, strengths and weaknesses. We had known about these but not used them in the project. I have used them with limited success in a quality improvement project since then, but not in a testing project:

Figure 3 - Risk: SWOT charts

Strengths	Weaknesses
Opportunities	Threats

Figure 4 - Risk: Know what you know - know what you do not know...

Do not know it Know that you do not know it	Know it Know that you know it
Do not know it Do not know that you do not know it Think you know it but you don't	Know it Do not know that you know it

Schaefer's spreadsheet - a boon for testers

Just following the end of the project, I took a long weekend in Edinburgh - and started the weekend with a day at the BCS SIGIST in Edinburgh.

Hans Schaefer was speaking on managing risk and choosing where to concentrate testing. It was all painfully close to home as he described the "bad game" where the tester either is blamed for being late or for missing errors... I sat there listening and thinking "if only I had heard this 6 months ago - a month ago - last week".

Schaefer asked:

“what is important and what is the worst that could happen?”

He talked about assessing risk by looking at the impact of a failure:

- Criticality - catastrophic, damaging, hindering, annoying
- Visibility - how many users/customers/public
- Usage - unavoidable, frequent, occasional, rare

He then talked about identifying the places where errors might be made and assigning a likelihood. Different teams might come up with different lists, for example:

- Experience levels
- Complexity of the design / system area
- Known defect prone area

He then combined these into a simple spreadsheet where areas of the system and problems could be assigned priorities for test. He also suggested assigning a score to the non functional as well as the functional areas of the system to be tested. I have adapted this spreadsheet, used it in projects and handed it on to others to use. The figure below shows a fictional version:

Figure 5 - Risk: Hans Schaefer's spreadsheet – fictional example of use

How CRITICAL?		<i>catastrophic</i>	<i>damaging</i>	<i>hindering</i>	<i>annoying</i> *
How VISIBLE?	<i>no. of users?</i>	<i>high</i>			<i>low</i>
	<i>no. of customers?</i>	<i>high</i>			<i>low</i>
	<i>public?</i>	<i>yes</i>			<i>no</i>
How much USED?		<i>unavoidable</i>	<i>frequent</i>	<i>occasional</i>	<i>rare</i> **
PRIORITY		1 <i>must fix</i>	2	3	4 <i>leave</i>

When assessing where to put most effort into testing:

	high	medium	low
<i>changed areas</i>		yes	
<i>familiarity</i>			no
<i>people - number</i>			no
<i>people - experience</i>			no
<i>people - turnover</i>			no
<i>time pressure</i>		yes	
<i>defect prone area</i>	yes		
<i>complexity of design</i>		yes	
<i>other factors....</i>			

Area to test	defect prone	complexity	usage	criticality	Sum
Weight	5	3	10	5	
installation tutorial	3	3	1	5	59 ***
rates calculation	1	2	2	1	36
letters	1	1	1	3	33
perf. installation tutorial	1	3	1	1	29
perf. rates calculation	3	3	2	5	69 ***

It has helped in both development and maintenance projects to allow customers, developers and testers to set their priorities and I would recommend its use as a simple and effective aid in prioritising tests.

C3 Goals, aims, objectives – improvement ideas

Earlier I mentioned that we asked in evaluation whether we could have defined our goals more clearly.

I heard Tom Gilb speak at a Quality Forum conference in 1989 and one point he emphasised was the importance of stating the project goal on one page in a way that everyone understands. Since then, as a quality manager and as a test manager, I have encouraged projects to do this. The difficulty is that projects often find it very difficult to write down the project goal.



Jayne Weaver's project triangle

What follows is an outstandingly useful idea for setting goals and monitoring against them which I have used over and over again on successive projects.

Around this time, I was working with a local charity, ARCOS, to look at quality management and process improvement. ARCOS provides speech and related therapies to aid children and adults with brain injuries to improve their oral and communication skills. ARCOS received funding from BBC Children in Need for a particular project and as a result, were required to set up, monitor and evaluate the project for their own benefit and in order to report back to BBC Children in Need on a regular basis.

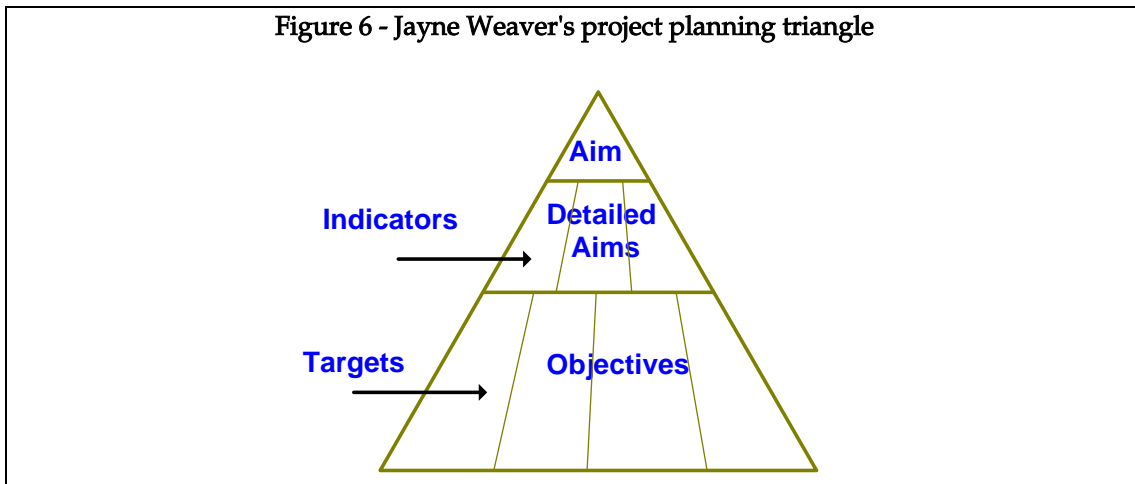
Jayne Weaver, an independent consultant who works with charities to help them understand how to set goals, monitor against them and evaluate projects, provided training in a method she had developed.



This method sets out the aims and objectives for the project, together with the targets and indicators in a manner that allows them to be communicated, agreed and understood by many people.

It is the Jayne Weaver Project Planning Triangle:

Figure 6 - Jayne Weaver's project planning triangle



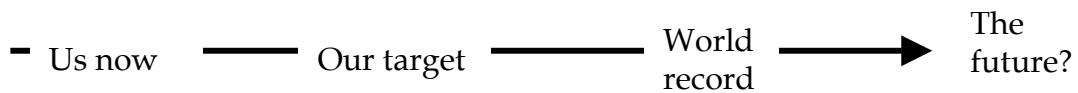
- The aim of the project describes the difference the project is expected to make - the reason it is being done.
- The detailed or specific aims describe specifically why the project is being done - they break down the aim.
- The objectives are the actions that will be taken in order to achieve the aim. Each specific aim must have at least one objective associated with it - else it will not get done. Each objective must have at least one specific aim associated with it – else why are you doing it?
- Measurement of success: The aims are measured against indicators - what difference has the project made? The objectives are measured against targets – is it on time, within budget, affecting sufficient people?

Note: targets alone do not measure success - you could meet all your targets, but if you had the wrong objectives, you will not meet your aims. Look for indicators that match your aims. You will be looking for quantitative and qualitative measures and you may want to set targets against the indicators. You will need to consider what evidence and measurements to collect. You also need to consider how much of the evidence can be anecdotal, how much is measured against an agreed standard.

Suppose you have an indicator "fewer problems with..." The target for this indicator might be "30% reduction in problems by 2003", but if we do not have a metrics programme in place which would help us gather the evidence we cannot measure against this target or know whether it is reasonable. But trying to define it will help us see other specific aims (e.g. "have better knowledge and control of our problem volume") and other objectives (e.g. "start to measure problem numbers"). We can start to measure numbers of problems now, and look for a reduction from now through the project. Any reduction is an indication of success, so next you would ask "how successful?". (Example adapted from a recent test process improvement project.)

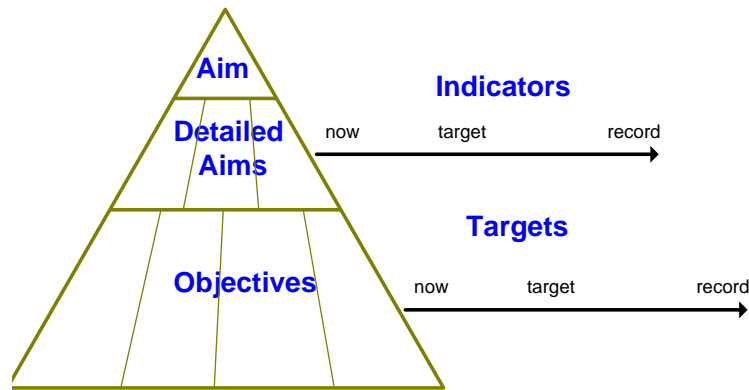
I have combined this with some ideas which I saw Tom Gilb present on setting targets. Gilb suggested that when setting targets one should put down on an arrow the current situation, where you want to be and the "world record". This quickly helps focus your mind on whether your target is achievable.

Figure 7 - Targets defined move clearly (adapted from Gilb)



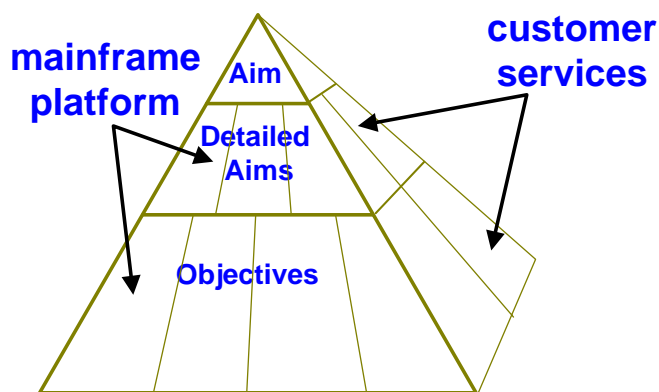
The Jayne Weaver Triangle and the Tom Gilb Arrow can be combined, if we want, or we could draw them separately:

Figure 8 - Goals and targets - triangle and arrow combined



It is also possible to set goals using multiple aims and objectives, if there are other projects and teams in the organisation, which will have their own detailed aims and objectives all leading to the same overall aim:

Figure 9 - Multiple aims and objectives



For example if you have many projects within a programme, each project has the same overall aim, and the specific aims for each project of the overall programme feed into meeting the overall aim.

Monitoring progress for the project now becomes relatively easy. At the start set up a monitoring mechanism, so that on a regular basis (quarterly, monthly, weekly, daily) you ask "How well is the project doing?". The ongoing process throughout the project is measured as:

- Targets against Objectives
- Indicators against Aims

- Have the targets been achieved?
- If not, why not?
- What can be done?

Inevitably the project will change and having the aims and objectives so clearly set helps to make the changes in a clear and controlled way.

It also helps when inevitably re-planning is required. It becomes much easier to examine targets that are not being met and if necessary redefine them.

Evaluation at the end of the project should hold no surprises as it will be the accumulation of the monitoring. Evaluate against aims and objectives, using targets, actuals and indicators and ask "What difference did the project make?"

The figures and tables below show some examples from projects, both IT and non-IT, to illustrate the types of aims and objectives that can be set. In each case the Jayne Weaver Triangle is drawn up first on a flip chart and then used both as an aid and as a means to document the aims and objectives. Then a tabulated summary with numbered aims and objectives can be drawn up, if you wish. The examples are drawn from four different projects:

- A fictionalised year 2000 project (as we have all been touched by this problem I hope you will find the example understandable even though it is not timely!)
- The testing component of the same project
- An IT process improvement project
- A non-IT voluntary sector project for ARCOS

It is followed by an exercise with examples for you to try yourself.

Figure 10 - Aims: why are you doing this project?

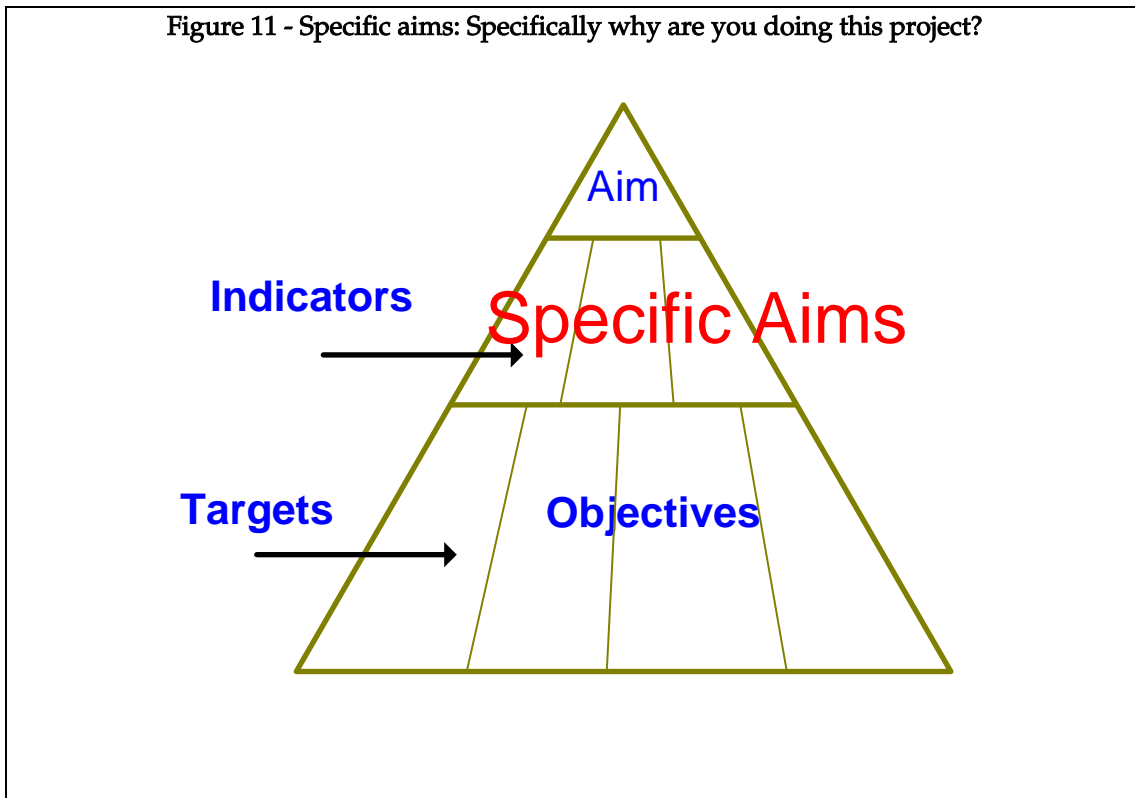


What is the reason for doing this project? What difference will it make?

Table 2 - Project aims

What is your Aim? Why are you doing this project? What is your reason?	For a Year 2000 project: <i>To enable the business to function into the year 2000 without (significant) loss of business revenue from IT software or hardware failures</i>
	For the testing on a Year 2000 project: <i>To enable the business to function into the year 2000 without (significant) loss of business revenue from IT software or hardware failures</i>
	For an IT process improvement project: <i>"To improve the delivery of software to customers, making the delivery more effective and efficient"</i>
	for ARCOS: <i>To improve the quality of life of people who have brain injury (acquired or developmental) and of their carers and families</i>

Figure 11 - Specific aims: Specifically why are you doing this project?



Specifically - why are you doing the project. Each specific aim must meet the overall aim.

Table 3 - Project specific aims

<p>What are your Specific Aims? Specifically why are you doing it? List the reasons. List the things that will make a difference.</p>	<p>For a year 2000 project: "... to have contingency and mitigation strategies in place for Y2K..." "... to enable the critical systems to function..." etc..</p>
	<p>For the testing on a Year 2000 project: "... to have a test strategy in place that addresses the identified risks" "to enable the critical systems to function..." etc..</p>
	<p>For an IT process improvement project: "... 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..."</p>
	<p>For ARCOS: "... greater independence..." "... improved oral skills ..." "... increased skill range / knowledge..." etc..</p>

Figure 12 - Objectives: What are you going to do to achieve your aim?

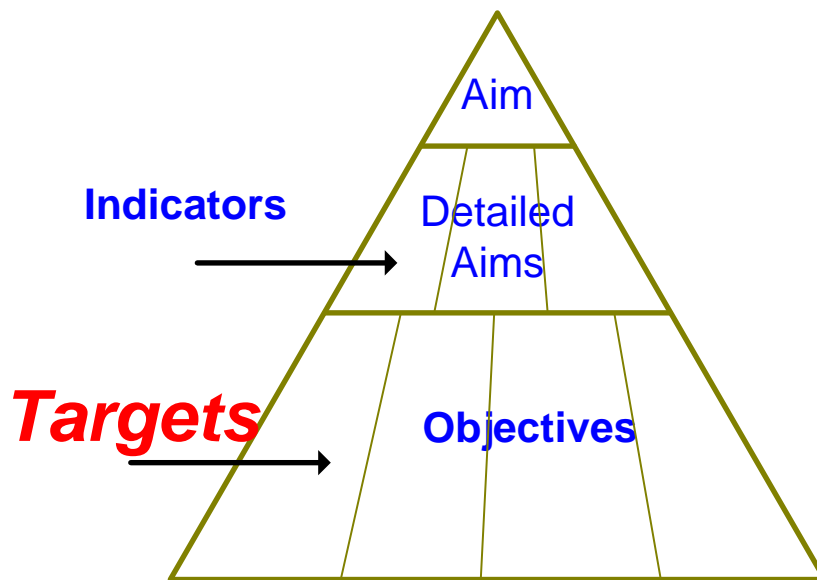


Each objective must help you meet at least one specific aim - or else why are you doing it? Each specific aim must have at least one objective - or you will not achieve it!

Table 4 - Project objectives

<p>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.</p>	<p>For Year 2000 project: "... carry out a business audit to identify all components affected..." "... carry out risk assessment for the business" "... perform test planning and organisation..." "... perform change design and re-code..."</p>
	<p>For the testing on a Year 2000 project: "...to set in place a mechanism to identify all the date related processing changes for retest ..." "... to identify and carry out a regression test strategy..." etc..</p>
	<p>For an IT process improvement project: If you have a specific aim to understand the risks associated with the ABC sector, you will have objectives which makes that happen: Objective O1: running a risk assessment programme covering ABC and comparing it with the risks of the DEF sector. If you have a specific aim to increase the benefit of early testing among the testing staff you will have objectives which makes that happen: Objective O2: presenting awareness seminars to all testing staff Objective O3: training all testing staff in the standard review technique</p>
	<p>For ARCOS: "... providing independent advice..." "... running a therapy centre..." "... providing outreach services..."</p>

Figure 13 - Targets: How will you measure progress?



Set targets against each objective. Check they are measurable and achievable.

Table 5 - Project targets

What are your Targets? -
to help you measure
progress of set targets
against each objective.

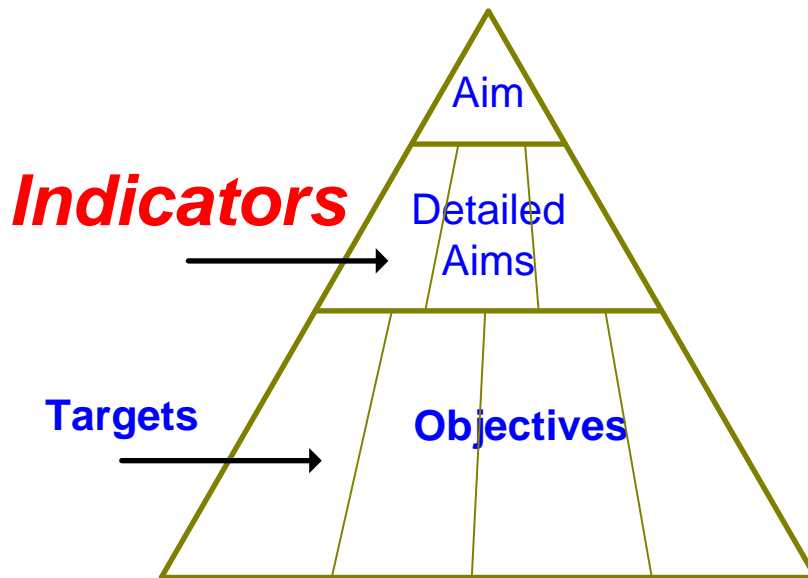
For year 2000 project:
by 1/5/1998 identify and classify affected systems
by 1/12/1998 all 20 critical systems compliant
business loss not greater than £ / hours...
systems "mtr" less than ...

For the testing on a Year 2000 project:
by 1/9/1998 have completed test planning in detail
by 1/10/1998 have built the regression test suite
by 1/11/1998 have run tests against 85% of code by statement
measure

For an IT process improvement project:
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

For ARCOS:
run 20 days of training courses
therapy centre open at least 200 days per year
provide therapy to 50 children during the year

Figure 14 - Indicators: How will you know the project has made a difference?



Each aim should have an indicator - something that shows the project has made a difference.

Table 6 - Project indicators

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.

For year 2000 project:
project team motivation remains good (MIP measures)
positive comments at progress meetings from user community (meeting minutes)

For the testing on a Year 2000 project:
Tester – developer relationship remains positive throughout project
Falling outstanding-test-report measure during second and third regression test runs

For an IT process improvement project:
Examples might be increased satisfaction of customers, increase in compliments to customer support, increase in profit during the next 5 years, developer/ tester relationships improved, reduced number of problems reported by customers, reduced on-site support time.

For ARCOS:
Carers / family / professionals better able to cope
Clients improve oral & communication skills, become more independent

Table 7 - Project aims and objectives exercise

Try this for yourself... This exercise uses the project triangle to help us define a project, how we will monitor during the project and how we will evaluate it at the end.

- 1. Choose a project.** This could be a current project, or a new project, or you could reflect on a completed project. Write the project name here:

2. Define the aim of the project

Ask your self - What difference will this project make? Why are we doing it? Write here the aim of the project:

AIM A0: To... _____

- 3. Define the specific aims for your project.** - Specifically, what differences will this project make? Write the specific aims of the project here; making sure each specific aim matches the overall aim:

AIM	This project will make a difference ...	Matched to A0? - tick
A1	To...	
A2	To...	
A3	To...	

Hint: you will be looking for words that describe a change, for example *better - increased - improved*.

- 4. Define the objectives for the project** - Now think about the objectives for your project - what will you need to do in order to meet your specific aims. Each specific aim must have at least one objective (thing you will do) if it is to be achieved. Each objective must meet at least one specific aim - else why are you doing it?

OBJECTIVE	What are we going to do?	Why are we doing it? (Which specific aim?)
O1		
O2		
O3		
O4		
O5		

- 5. Review progress** - It is worth stopping at this point and reviewing your definition of the aims and objectives with a colleague. Draw your project triangle with its aim, specific aims, objectives. Review with a colleague thinking about: Can you tell WHY the project is needed? What difference will it make? Can you see an objective that will help to achieve each aim? Can you see a reason for each objective - WHY are you doing this? Will it help achieve the aim?

Table 8 - Project aims and objectives exercise continued

6. Targets - measuring the objectives - Now think about how you will measure achievement for your objectives - you need to look at targets. Each objective should have at least one target associated with it. Write at least one target for each objective:

TARGET	How much of it do we want to achieve	For objective:
T1		
T2		
T3		
T4		
T5		

Hint: look at quantity questions: How soon/when by? At what maximum cost? Affecting how many people?

6. Indicators - How will you know you have made a difference? You need to set indicators for each aim. Each aim should have one or more indicators that you will track.

INDICATOR	Have we made a difference?	For Specific Aim:
I1		
I2		
I3		
I4		
I5		

7. Review progress - Now review again to make sure you are on track and everyone is in agreement.

Draw your project triangle with its aim, specific aims, objectives, and show the targets and indicators. Pair up with your neighbour and review each other's projects. Are the targets clear? Sensible? Measurable? Achievable? Matched to objectives? Are the indicators clear? Sensible? Measurable? Achievable? Matched to aims?

8. Summarise your project

Project name:

Aim:

Specific aims:

Objectives:

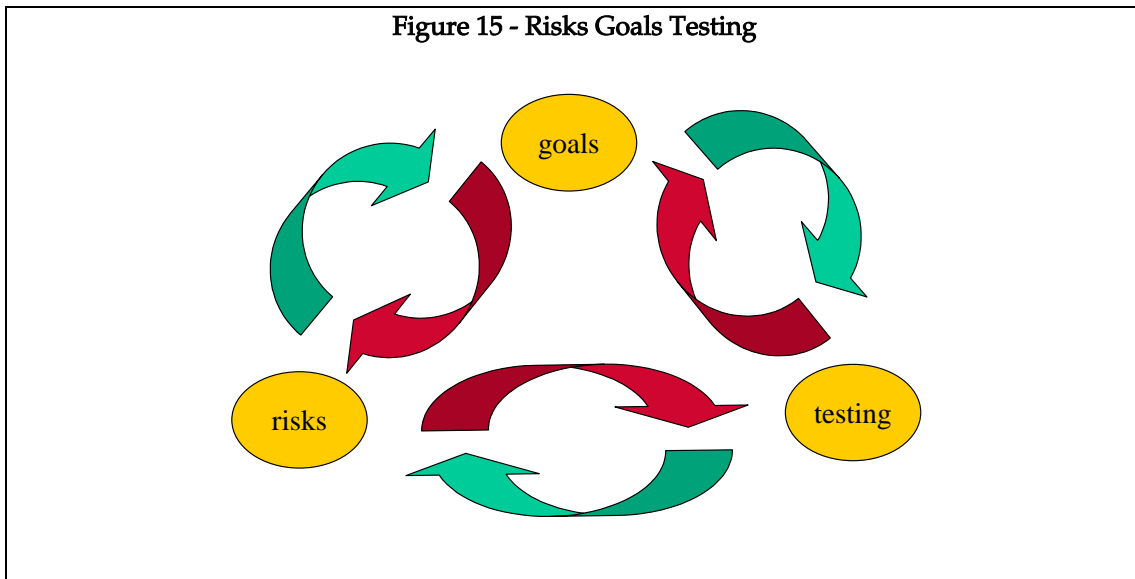
Targets:

Indicators:

D. How to focus testing where it is most needed

D1 Understand goals and risks

The goals, risks and the testing effort in the project have to be inter-linked:



During the conversion project, and specifically throughout the testing, the test team worked with the business, the project manager and the developers to assess and reassess where to focus the testing. By the end of the system test phase, we were reassessing on a daily or twice daily basis. This tight focus of the testing against the goals and the risk was very important.

That this could work successfully on this project indicates to me that it is the way to work on other projects. The “not enough time for bureaucracy” excuse does not work. We found that keeping careful and detailed logs of all problems and decisions really did help us to keep the project on track.

I have used the Jayne Weaver project triangle on all projects in the last two years, even if only for my own benefit. I have also used the Hans Schaefer spreadsheet successfully on IT projects. These two tools together have enabled me to understand goals and risk in order to focus testing. They have also proved invaluable when discussing goals and risks with customers, project managers and developers.

D2 Use specialists

I would definitely get early test specialist involvement again. Here are some of the areas where test specialists helped the project, and where testers were aided by other specialists:

- Production of the overall test strategy: As test manager, I produced a detailed overall test plan and strategy focused on the issues for this project. I reviewed this with the other testers, the senior developers and the project manager. They were agreed that they would not have taken the plan to such detail or, consequently, uncovered the same issues at an early stage. I find occasionally that as a test specialist I am called into projects late on, perhaps when there are already problems. At that stage it is still

worth spending the time on defining a test strategy, assessing the overall aim of the testing and the project and assessing risk.

- Tester help to the developers in writing the unit tests: In the conversion project, one of the testers wrote the unit test plans for/with the developers and this was extremely successful. The developers ran the tests. The unit testing was as a result extremely good and code arrived into system test in a very solid state. Use of BS7925-2 to help developers understand improved test design methods has proved useful. I know other testers have also found the methods described in the standard to be well received, especially as the standard has good examples and is focused at developers.
- Developer help to the testers in identifying risk areas for the system tests: The developers were actively interested in the tests being as effective as possible and very co-operative in identifying what they saw as risk areas in the code.
- We had a dedicated development person for test data extraction. He worked from test specialist requests – the testers defined what data was required to meet specific test conditions and the developer extracted the data. In this way we moved from the original developer plan to run tests with a random 10% of the data base (about 3000 records) to a series of test databases starting with a 200 record test database that covered all the investment combinations in the whole database.
- Understanding of the test control and management process. The testers defined a series of test cycles leading to a dress rehearsal of the full conversion. Each test cycle had a specific goal – to look at a particular conversion data set or to look at non-functional areas such as performance.
- Testers controlling the test management but with an aim of communicating clearly with project management and developers. On the 1997 project a test management tool was not available and I used a simple Excel spreadsheet. This was very successful, and I would use it again. Using simple tools which are available to everyone to log and track test progress and test problems means everyone involved in the project had easy access to the information and it could be emailed to remote parties if needed.
- Business expertise and aid to the testers in defining expected results. The business experts set share prices and calculated the expected results for the test databases and for the dress rehearsal.
- Developer tools available to the testers. The developers had a number of database manipulation and interrogation tools, which they used to help the testers check test run results.

As well as noting these advantages, I would also say that I now see a (*new for me*) key reason for writing a test strategy early in the project. The test strategy is not just a document for the testers, nor is it just a document informing the rest of the project of how the testing is to be carried out. Writing it early on in the project is not just for early defect prevention in the product. It is for defect prevention in the process and in communication. It is to *educate* the rest of the project team about *how* the testing will be run and *why* it will be run in that way.

For example - The purpose of testing is to find defects or faults in the product, therefore the greater the number of errors the larger the celebration.
Unfortunately this can lead to some misunderstandings...

Test finds 0 problems - the tester is miserable

Test finds 120 problems - the developer is miserable

Project manager evaluation comment :

"I felt in control until we reached system test"

Tester reply:

"the more you find now the less there are to find later"

"if you could predict problems you would prevent them..."

I would therefore use the test strategy to:

- Set the test vocabulary with reference to BS7925-2 and to any standard (published or de facto) used by the teams.
- Set expectations and educate the team and the customers about what software testing can and cannot achieve.

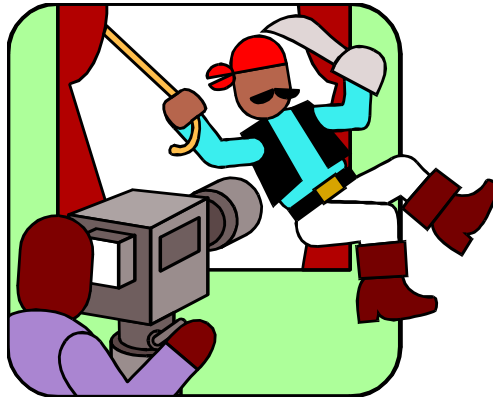
During the conversation described above I explained to the project manager about the concepts of :

- *Testing S curve*
- *Positive and negative testing*
- *The purpose of testing to find faults*

His response was immediate and positive - it was simply that he had not looked at testing in that way. We testers take this for granted and forget that other specialists do not necessarily have the same taken for granted assumptions.

- Define the use of standards. In the 1997 project, we used a set of skeleton documents which I had based on the test documentation standard ANSI IEEE 829 (1989, re affirmed 1991, updated 1998) Although this has its disadvantages (for example it does not explicitly tell you how to adapt the documents for re-test and re-use) it is a good basis for documenting tests in a standard consistent way. Test specialists are aware of the standard, and non-test specialists can understand documents written using the standard and use it themselves. Since 1997, of course, more standards have arrived on the scene - the Component Test Standard BS7925-2 is a tremendous aid to testers and developers
- Set out the test process and then use test specialists to educate developers, project managers and customers to understand that there is a test process
- Describe how the testing will be managed and the responsibilities for test incident logging and tracking. In large projects with teams drawn from many backgrounds there may not be a common view about the test management process.

E. Finally...



A question to ponder... Why is it that when you deliver a project *on time, to budget, requirements delivered as requested and as needed* sometimes, depending on the company culture, you can find that the customer is happy except that

they think the IT project is an action movie ... and they want the suspense into the last reel.

Here are some particularly interesting evaluation criticisms from customers:

"I would have liked to see you pull the stops out"

"You never looked under pressure"

"People left work on time..."

"Your estimates were correct"

Good luck for your next big project!

To succeed in your next project you need many skills and qualities. Among them are:

- Good understanding of the real goals of the project
- Tight project management therefore no widening of scope
- Good risk management
- Good relationships across teams and within the team
- Early involvement of specialists
- Hard graft and long hours off-set with treats...

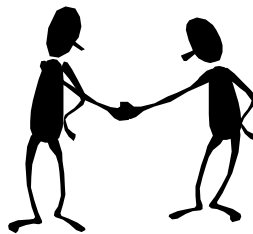
Because if you don't...



Approaching the next big project

In summary, what I do now on each project is:

- Focus on goals - Using the Jayne Weaver triangle to help understanding and communication of aims and objectives and the Tom Gilb arrow to help define targets
- Focus on risks - Using the Hans Schaefer spreadsheet to help understanding and agreement of focus from risk assessment and the "In your garden" model to help people understand the difference between likelihood and impact
- Focus on education and communication - Using the test strategy document to communicate to and educate the rest of the team, basing it on the Test Documentation Standard and the Component Test Standard
- Focus on co-operation - Encouraging project managers and developers to see testers as their allies not their enemies.



To finish, I would like to thank my clients for their help in putting together the evaluation and earlier presentation on which this paper is based. I also thank the other practitioners and the experts in testing, quality and management who have helped me over the years either directly in projects or in conference presentations where they have shared ideas and good practice. Thank you.

References and further reading

Work referenced in this paper

The works and individuals specifically referenced in this paper are:

Jayne Weaver's project planning triangle:

Jane's contact information is on the IE Testing Consultancy web site www.ietesting.co.uk.

I include the triangle in my training courses.

Hans Schaefer risk spreadsheet:

Specialist Interest Group in Software Testing 24 November 1997

"Surviving Software Testing under Time and Budget Pressure"

Hans Schaefer's email is Schaefer@c2i.net.

BCS SIGIST see the BCS website www.bcs.org

Tom Gilb's target arrow:

Quality Forum 20 June 1989

"Software Engineering Design - a practical handbook based on process control of object oriented sets"

Quality Forum see the Quality Forum website www.quality-forum.co.uk.

Standards:

BS 7925-1 Test vocabulary

BS 7925-2 Component test standard

ANSI IEEE 829 1998 Test Documentation Standard

Other Publications and Sources

Risk:

If you would like further illustrations of risk assessment Edwards Tufte's Visual Explanations (0 9613921 2 6 Graphics Press) and Ian Stewart's Royal Institution Christmas Lectures (1997) both use the Challenger accident to show the statistical nature of risk assessment. The Royal Institution Christmas Lectures video tapes can be obtained from BBC Videos for Education and Training. The whole lecture on statistics is useful particularly to those of a non-mathematical background.

I have also found the following useful:

Title	Author(s)	ISBN number and Publisher
Just about managing	Adirondack	1-872582 17 6 - London Voluntary Service Council
Black box testing	Beizer	0-471-12094-4 - Wiley
Software Testing Techniques	Beizer	0 442 206720 - Van Nostrand Reinhold
Mythical Man Month	Brooks	0 201 83595 9 - Addison Wesley
Software Test Automation	Fewster & Graham	0-201-33140-3 - Addison Wesley
Software Inspection	Gilb & Graham	0-201-63181 - Addison-Wesley
Complete Guide to Software Testing	Hetzel	89435 110 9 - QED
How to lie with statistics	Huff	0 14 02 1300 7 - Penguin
IEEE Standards Collection - Software Engineering	IEEE	1 55937 253 2 - IEEE
Testing Computer Software	Kaner, Falk, Nguyen	0 442 01361 2 - Van Nostrand Reinhold
Software testing in the real world	Kit	0-201-87756-2 - Addison-Wesley
Testing an IT Service for Operational Use - IT Infrastructure Library	MacFarlane, Warden CCTA	0 11 330560 5 - HMSO

Brief descriptions of some projects - but not all that I refer to in the paper - appear on my web site: www.ietesting.co.uk.