To track or not to track

Agile project tracking that really works

I have measured my life with coffee spoons
T.S.Eliot 1917

Note from Rob: I wrote this paper for a client in the mid-1990's and upon re-reading it I have updated it with some thoughts about the all-pervading traffic light model.
Warning: not for the faint hearted or hard core traditionalist.

In the 1990's, many of the project management practices being applied in business and administrative projects are still based on traditional engineering and construction project management concepts unchanged from the 1960's.

While, the underlying concepts of project management such as estimating and scheduling of tasks via Gantt charts[i] are still valid, many of the traditional project management concepts are blindly implemented in contemporary organisations without little critical evaluation of their relevance to the complex and dynamic business and IT projects of the 1990's.

An example of this problem is the approach to *project tracking* (determining of the status of the progress in the project) and, as a result, *project reporting* (the provision of information regarding project status to senior management and stakeholders) implemented in the majority of organisations.

Earned Value ...what value?

The traditional approach to project tracking and reporting involves three related concepts:

- the estimated effort and cost of the task ;
- the actual effort and cost expended at time of tracking ;
- the estimated remaining effort and cost at time of tracking.

Using these simple concepts, the project manager and team agree to review or track on a regular basis (e.g. weekly) the tasks undertaken during the tracking period. Then depending upon the various measures such as percentage complete and estimated effort to complete, the project manager, team member and various stakeholders are able to determine the status of the project .. or so the theory goes.

Let's assume that Mary has one task estimated to take 5 days that was scheduled to commence on Wednesday and be completed the following Tuesday. Let's also assume that the agreed tracking process occurs on Friday at 4.30 pm.

As shown in Figure 1, by Friday, Mary would have completed 60% of the effort and cost estimated for the task (assuming accurate estimation and no other unscheduled activities for Mary). This also means that Mary has 40% of the effort and cost remaining to be completed by her deadline. Commercial project scheduling tools such as Microsoft Project, Artemis and Project Management Workbench provide the capability of displaying actual progress visually and also include more advanced (?) project tracking techniques such as Earned Value Analysis which are variations on the % complete model.

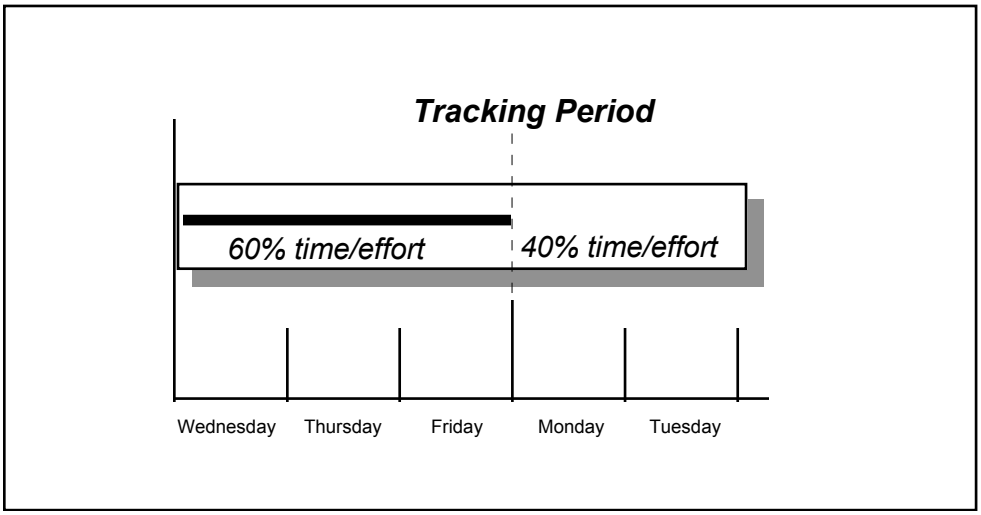

Fig 1. Project tracking (theory)

As any project professional will understand, there are significant conceptual and practical problems with this approach.

Problem 1 - Patterns of completion

The underlying concept in this approach to task/project tracking is the concept of *linear completion*. In other words, the idea is that Mary produces work at a regular rate over each day of the 5-day task and that all work is of "equal value" or "deliverable-ness". Given that these project scheduling and tracking concepts were developed in the logistics, engineering and construction sectors, this makes some sense. For example, if you have a wall to paint or a concrete pour to complete and it was estimated to take 5 days, then at the end of the 3rd. day, you would expect 60% of the wall to be painted or 60% of the concrete to be poured.

It is our experience and that of thousands of business and information technology professionals, that there are few tasks, if any, in business and information system projects such as developing a

new organisation policy or an information system that exhibit the same completion patterns as painting walls or pouring concrete. Figure 2 shows some of the task completion patterns observed in tasks such as requirements analysis, design, programming, drafting legislation and so on.

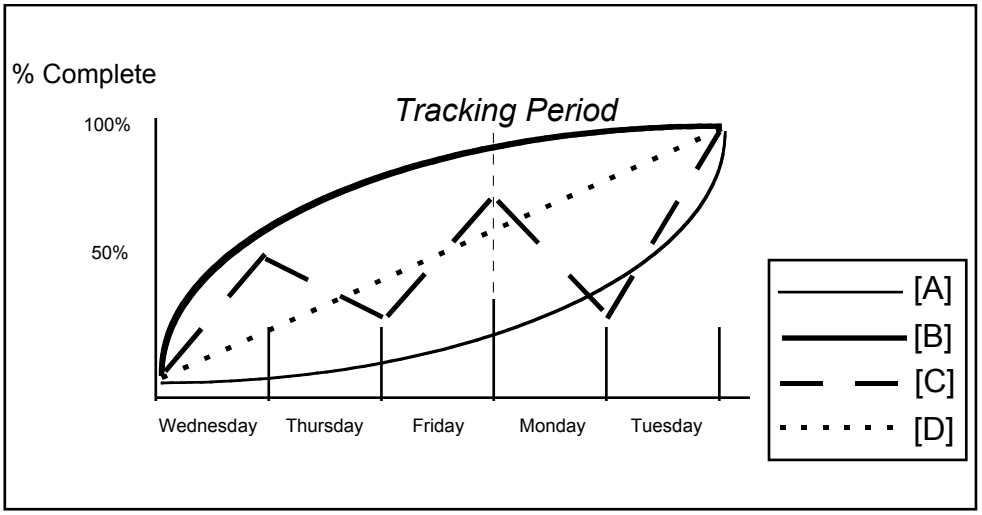

Fig 2. Project tracking (reality)

Task A has a completion pattern where on Day 3 around 20% of the product is complete. Task B is 90% complete on Day 3, Task C is 70% complete on Day 3 and 25% complete on Day 4. All are on track and will deliver completed products on the deadline!

It is our experience that complex problem solving activities such as programming, design and writing documentation tend to exhibit completion curve A. Systems analysis, some problem solving processes and product or system testing tend to have a curve B completion pattern. Many complex problem solving techniques involving making assumptions (that may prove incorrect later) follow completion curve C.

As stated earlier, in our experience, few tasks follow the linear completion curve D, which is the default tracking concept in most scheduling tools.

Problem 2 - Deliverables aren't always deliverables

One solution that is offered for overcoming the patterns of completion problem is the concept of tracking completed deliverables not tasks. While, this variation on the 0/100% complete tracking method avoids the % complete problem of completion curves A, B, and C, it is more suited to smaller deliverable cycles.

This approach often results in project teams having to produce deliverables on a regular basis (say, weekly) simply to reflect the tracking cycle rather than the specific nature of the task or project. However, for longer projects and activities the deliverable approach still poses problems.

For example, a systems analysis exercise in a project involving 10 interviews of a number of business experts may require a two months elapsed effort. The tracking of interim deliverables such as interviews completed, for example, may indicate that only 3 interviews were conducted in the first month (i.e. project appears behind schedule). However, in the second month, the remaining 7 interviews go much quicker as the team has gained most of the critical information in the first 3 interviews (completion curve B) and the systems analysis is completed on time. What is key here is that until the other 7 interviews are complete and the information gained in the first 3 interviews is confirmed, the team does *not* know whether they are in schedule or not at the end of the first month!

In addition, as shown in Figure 3, the % completion of interviews has a completely different curve to the % completion rate of understanding the business requirements which is the key to the final deliverable being produced - Systems Requirements.

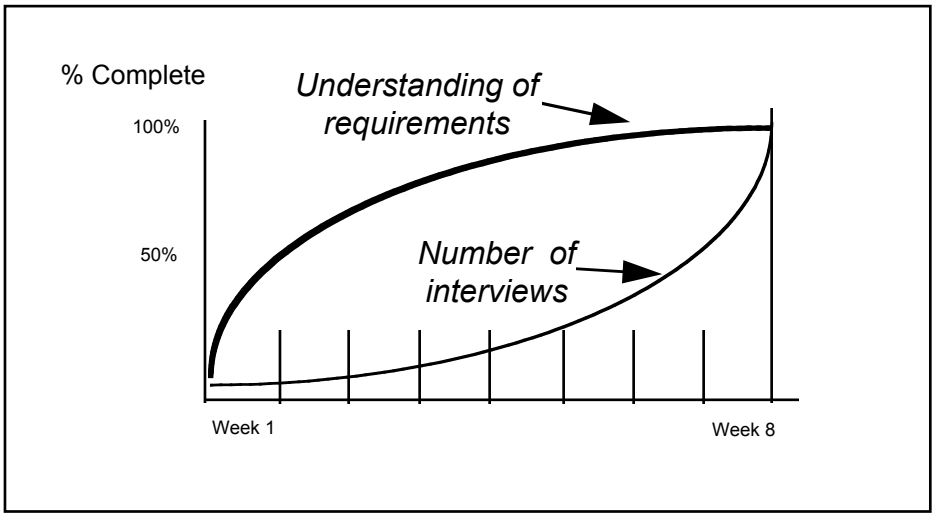

Fig. 3 - Completion versus activity

In effect, no matter which tracking technique is used, the assessment of progress remains mainly *subjective*, not objective.

Problem 3 -Scheduling tool implementation

Some scheduling tools allow the default linear completion method to be over-ridden. For example, in Microsoft Project, you can use an

actual effort option, which enables you to enter 20% complete on Day 3 (curve A). However, because of the difficulties of programming these tools, once you enter a 20% complete figure, the task is then shown to be 40% over schedule (as it "should" have been 60% complete).

Alternatively, some tools allow options such as the 0/100% complete method which involves a binary state for tasks - either they are completed or not started. Another option involves all tasks having been commenced being treated as default 50% complete until they are finished. Clearly, these options do not recognise the complex patterns of completion and, in reality, are poor compromises.

As a result, most scheduling tools cannot allow for real world tracking.

Problem 4 - Tracking the negatives

Traditional tracking mechanisms focus almost exclusively on estimates and actual costs, effort and time. In effect, the negative side of the project "balance sheet".

As will be discussed later, by tending to concentrate mainly on costs and schedules, most project tracking and reporting methods miss the issues of benefits, risks and other key project management concerns.

Daily builds: micro-deliverable tracking

Microsoft, Open Software Associates and other innovative companies develop systems using advanced tools that enable the rapid and incremental building and testing of components on a daily cycle. This technique does go some way to avoiding many of the problems with tracking raised in this paper. However, this technique cannot be used until detailed specifications and design are completed - so tracking still remains a problem for a large part of the development cycle. Of course, Agile folk will also talk about daily-standups.

In many organisations, the monthly reports to senior management tend to only show the variances between original estimates and current reality and the factors that caused the variation.

Put simply, most tracking reports have the same structure:

Original Estimates/Costs *Revised Estimates/Costs* *Excuses*

This emphasis on costs and deadlines has placed many project groups such as information technology in a position where, at the end of the project, the only information retained on the development process is the costs and effort - the negatives.

Problem 5 - The costs and all the costs

Finally, the long-term emphasis on monitoring, tracking and recording costs has resulted in many project teams *hiding* the real costs of projects.

As observed by Capers Jones (1996) and Thomsett (op cit) many IT and business projects covertly "blow" budgets through unpaid and unrecorded hard work. Our research confirms Caper's findings that projects are typically 30-50% over cost as a result of unplanned evening and weekend work and that the formal project cost tracking systems often do not record this cost. Our group has also identified similar covert cost blowouts in the effort required by business people in project-related activities such as requirements specification, testing, conversion of data, documentation and job redesign. As a result, many of on time and in budget projects have indeed failed to meet the estimated budget though the formal cost tracking process indicated that they had met budget.

The project is fine … go away!

Because of these problems, the reality of project tracking is far removed from the theory of tracking % complete or Earned Value.

In most organisations, the most common approach to project tracking involves the project manager or a senior manager asking the team a simple question:

How is the project going?

The answer is generally:

Fine[ii]

However, the apparently simple answer often hides a much more complex answer :

Fine (translated).

(We don't know what we are doing at the moment and are a little confused because our key clients changed the requirements yesterday. But, don't panic. Last time we got a requirements change we managed to drop off some other requirements so we seemed to be back on track. Tomorrow things will change anyway. To summarise, no one has gone crazy yet and we are pretty confident that we'll all survive .. so go away)[iii]

Clearly, when considering that the company may be investing many millions of dollars on the project and the product that is being produced may be vital to the strategic survival of the company, the problems of project tracking are very serious.

The operation was a success but the patient died

This well-worn medical joke has a rather serious implication within the project world. As discussed by Thomsett (1995, 2002), traditional project management focused on the development component of the project life cycle. As shown in Figure 4, contemporary project management adopts a whole of life perspective.

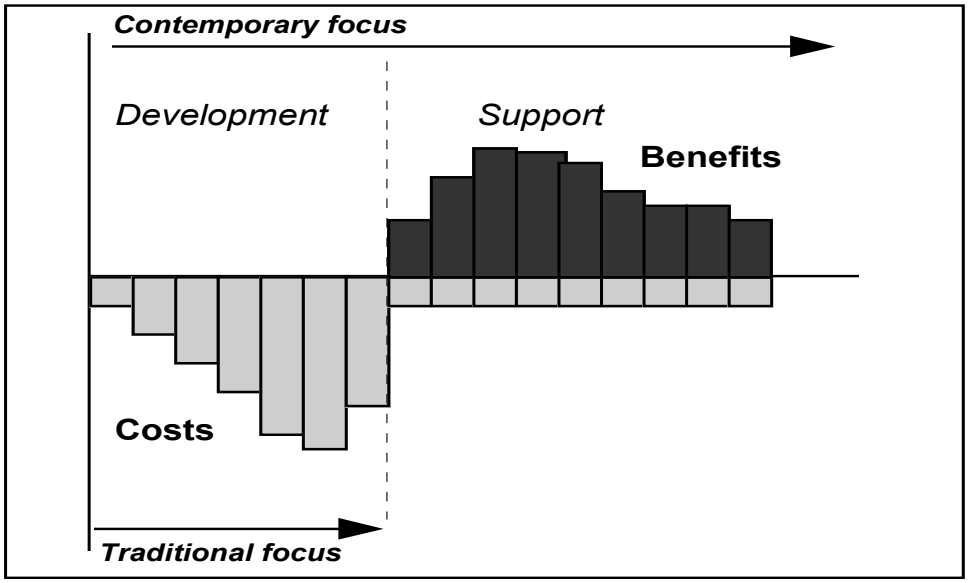

Fig 4. Tracking perspective

In traditional project management, as the project progressed through Feasibility Study to Implementation, recording of costs, duration and effort and reporting of these *process* metrics was common. However, after Post Implementation Review, tracking and reporting mechanisms were stopped as the project was *finished.*

Few organisations tracked either the **support** costs or the **benefits**. The absence of on-going measurement in the production cycle has left many business people with little hard evidence of the value added through new products or information systems. It has also enforced the view that many executives have of I.T. and other project groups as *cost* centres.

Moreover, the focus on development effort, cost and duration often resulted in the compromise of long-term project success (see later) with short-term cost and duration or deadline control. For example, the tracking of development progress reveals that the project is

behind schedule and over cost. To get the project back into schedule and to reduce the cost-blowout, the project manager and team consciously degrade quality and de-scope the project's requirements by cutting back on testing and documentation and by dropping out what is perceived to be non-essential functionality.

The project is now on schedule and in budget and potential long-term benefits and support costs have been compromised. The project tracking reports show that the project in now in budget and on schedule. In reality, the project is back on track for *failure*.

The key to effective project tracking is that we should be tracking whether the project is going to be **successful** not just is it on time and in budget.

Defining project success – an agile perspective

As discussed by Thomsett (1995.1) most definitions of project success focus on the traditional concerns of:

- meeting requirements (process, capability, data and function);
- keeping within budget (people and capital); and
- delivering within estimated deadlines.

As a result, the tracking mechanisms focus on these factors.

The contemporary definition of success developed by our group reflects the difference between an internal or software factory view of success such as the one above and an external or added value chain or service view of success (Thomsett, 2002). The fact that a project has met the requirements (objectives), budget and deadlines is simply a measure of the internal development process effectiveness and is not a measure of the product or added value effectiveness.

In other words, for a project to be successful, it must meet all the following criteria:

- satisfied stakeholder/clients;
- meets quality expectations/requirements ;
- meets functional requirements ;
- within cost (paid, unpaid and business expert costs) ;
- within deadline ;
- delivers sustained and actual benefits (added value);
- provides team satisfaction.

More importantly, it is our experience that different projects will require a different *combination* of the above success factors to be considered successful by the business groups.

For example, in a project where the need to develop a new product quickly for a competitive market from the perspective of the business experts funding the project, added value, deadline and minimal functionality may be the key success factors. As a result, the business group makes a pragmatic decision to "trade-off" these factors against more functionality, higher quality and lower costs. In another project, the quality and functionality may be paramount. As a result, the deadline and costs may be less important to the business client.

Tracking success: an agile view

Given this contingent and pragmatic view of project success, the entire concept of project tracking becomes different to the traditional and more common approach.

Case 1: Over budget but fine

As shown in Figure 5, a project has a very high Return on Investment ($200,000) and the project tracking reports indicate that a cost blowout of 30% will be incurred in the development phase. The project will still be successful as the high return on investment ($170,000) is still acceptable to the business groups.

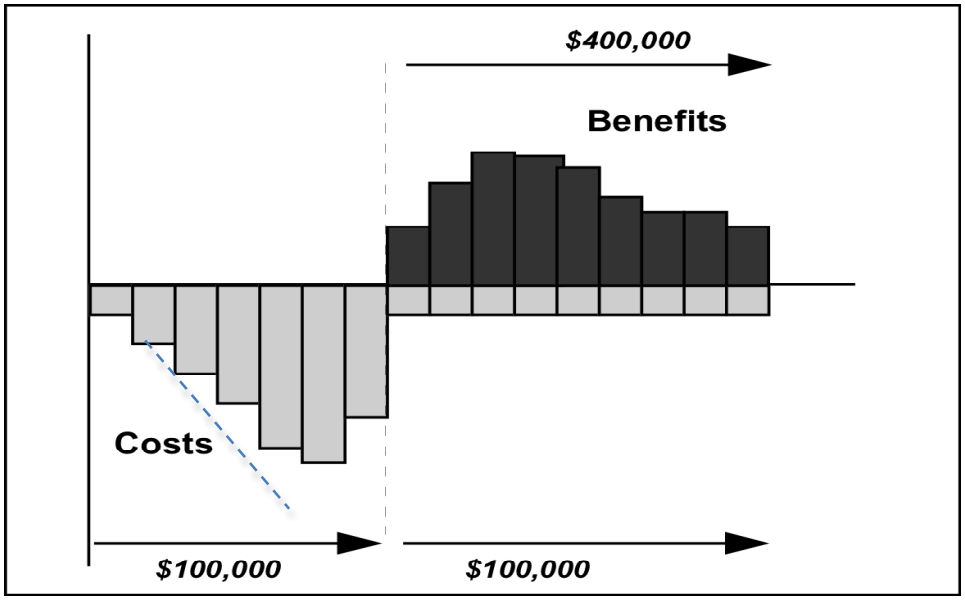

Fig 5. Over cost and OK

Case 2: Over budget and not fine

As shown in Figure 6, a project has a marginal Return on Investment ($50,000) and the project tracking reports indicate that a cost blowout of 30% will be incurred in the development phase. The project will not be successful as the reduced return on investment ($20,000) is not acceptable to the business groups. In this case, the project costs would need to be reduced or the project cancelled.

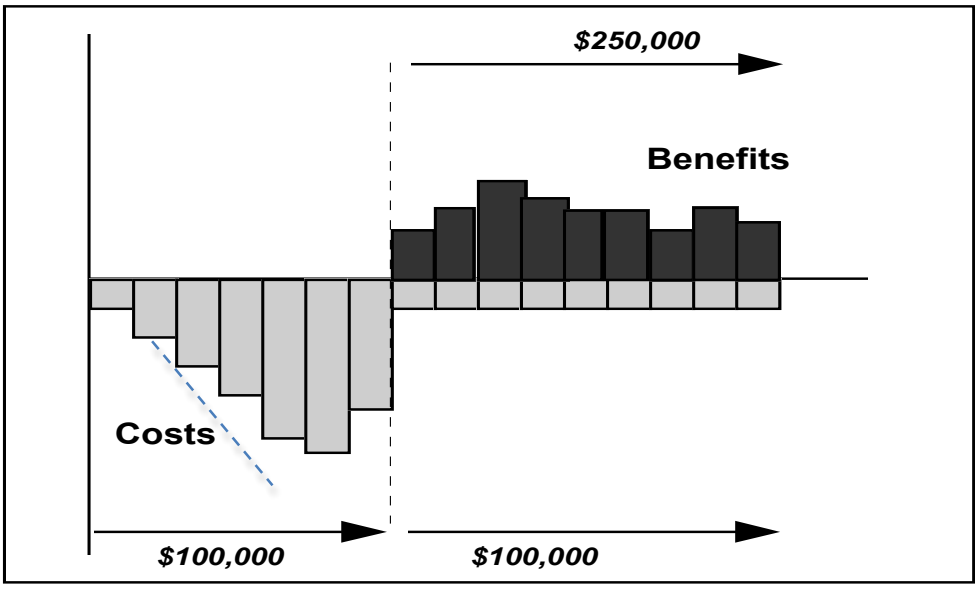

Fig 6. Over cost and *not* OK

Case 3: Behind schedule and in trouble

In Figure 7, a project has a high Return on Investment ($250,000) but this is dependent on getting to the market before a competitor. A blowout in the schedule and delays in shipping the product will result in a loss of $100,000 and the Return on Investment will be compromised.

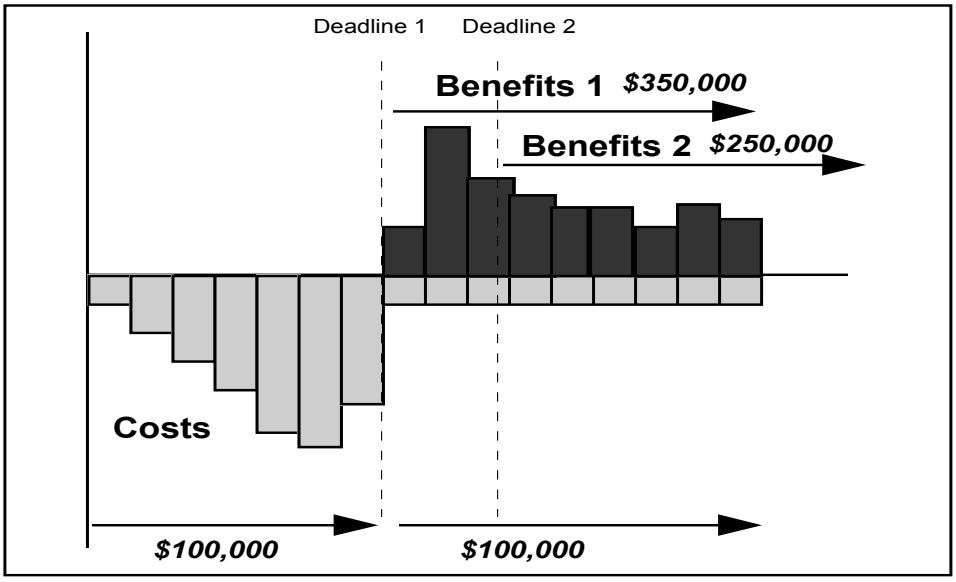

Fig 7. In schedule and *not* OK

Case 4: On schedule, under quality and in trouble

In Figure 8, a project has a high Return on Investment ($200,000) but to keep the development costs, which are blowing out within budget, the team degrades the quality of the system. As a result, the support costs are blown out from $100,000 to $250,000 and the expected Return of Investment is reduced to $50,000.

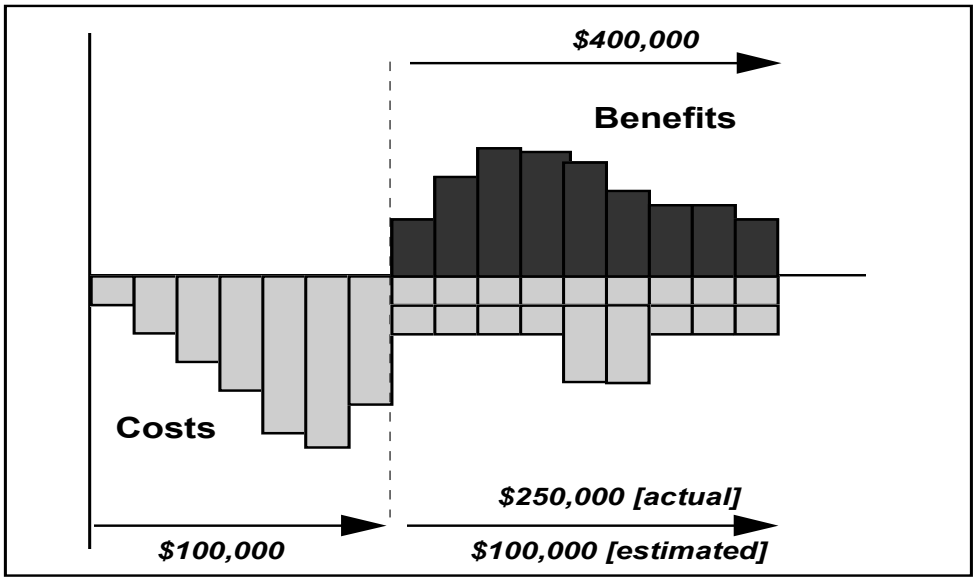

Fig 8. In schedule and *not* OK

Case 5: Behind schedule, over cost and fine

In Figure 9, a project has a high Return on Investment ($150,000) based on estimates of development costs of $100,000 and support

costs of $100,000 but faced with potential development cost blowouts the team has two options.

In Option 1, which is similar to Case 4, the team can degrade the quality of the system. This will result in the support costs being blown out from $100,000 to $200,000 and the expected Return on Investment being reduced to $50,000.

Option 2 is that the team allows the development costs to blowout by 30% to $130,000 and to delay shipping by a few months. The support costs are maintained at $100,000 but expected benefits are reduced to $300,000. Option 2 results in a better Return on Investment of $70,000 than Option 1. However, the team would be shown as behind schedule and over budget in traditional tracking reports.

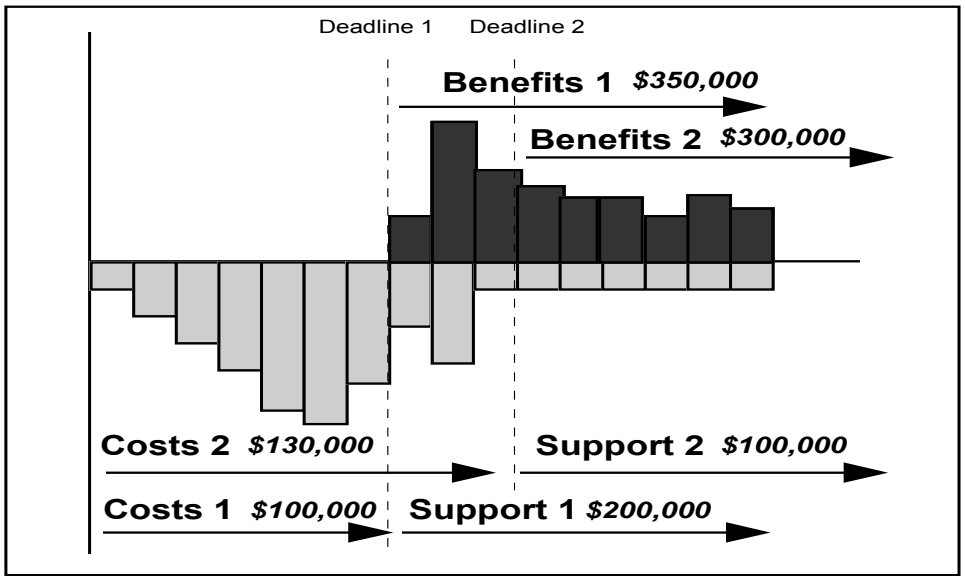

Fig 9. Behind schedule, over cost and fine

There's more - project risk changes everything as well

As discussed by Thomsett (op cit), project risk is a structured assessment of the factors that will influence the cost, effort and potential success of a project. Typical risk factors include system or product size, level of innovation, team skills, business client participation and support, development platforms and so on.

Research by Jones (op cit) and others has shown that there is a direct correlation between project risk and estimation accuracy. Simply, high risk projects have lower cost estimation accuracy and

experience higher degrees of schedule slippage than lower risk projects.

A project tracking report for a *low* risk project that indicates that the team is 30% over budget and 2 months behind schedule has completely different implications for the project manager and executives than a *high* risk project (of a similar size) that is 30% over budget and 2 months behind schedule.

In the case of the low risk project, the report is indicating serious problems whereas for the high risk project the report could be indicating that the project is performing *normally* for high risk projects.

While the concept a project being behind schedule and over budget being considered acceptable may still upset some senior managers, the cases discussed above should have made the point.

So what should we track?

We should be tracking information that enables senior management, the project manager, team members and stakeholders to assess whether the project is on target to be **successful**.

An agile project tracking and reporting mechanism would focus on the factors of success specific to each project. However, as a general rule, the following information would be the minimum that should be monitored, tracked and reported:

- quality expectations/requirements;
- functional requirements;
- project risks;
- development and support costs (paid, unpaid and business expert costs);
- schedules - estimated and actual; and
- expected and actual benefits.

Given that this set of information should have been developed during the initial project planning and approval processes (in our group's approach, this information is included in the Project Business Case), the tracking process is a natural extension to the planning process.

In addition, there should be reports on the other success factors such as stakeholder and team satisfaction.

More importantly, as shown in the five cases earlier, by expanding the focus of project tracking and reporting, the project manager and team can provide more useful information to senior management and stakeholders, enabling them to make more informed decisions about their projects.

Isn't that what project tracking and reporting should be all about?

On the use and meaning of traffic lights

One of the most common forms of project reporting is the use of traffic lights.

I first saw this model was when it was used by Accenture consultants in the mid-1990's[iv]. Over the past decade, it has become the default model for project reporting in most of our clients. However, my concern is similar to the one I raised earlier about the adoption of tracking models from construction and engineering by business and IT without really thinking through the implications and relevance of those models.

Fig 10. The ubiquitous traffic light reporting model

I have always had some concerns about the underlying theory of this model and, more practically, *what does it all mean*?

Let's deal with the practical issue first.

What do each of the signals really indicate?

The "green" looks pretty easy but, as the saying goes, the devil is in the detail. To most people, it means that the project is progressing according to the *plan* or *schedule* but does it also mean that the other critical success factors such as stakeholder satisfaction, benefits assumptions, support costs, quality and so on are meeting agreed levels as well?

What if we were facing the situation outlined in Case 1 earlier where costs are blowing out but the R.O.I. is still positive? Is that a case for a yellow/amber signal or a green signal? What is there is a scope and objectives change that can be met within the contingency of the project schedule and costs? Is that a yellow or a green? What if the project risks have altered and a new major risk emerge, for example, there is a change of project sponsor. The project is still meeting all expectations (i.e. it is green) but a new sponsor has to be bought into the project? Yellow or green or red?

What does yellow/amber mean?

Well, it depends … to some project managers, it means that there are concerns with the project but the project manager will be able to deal with the issues. To other project managers, it means that new risks have emerged that have not been resolved. To others, it means that there is some compromise of project expectations but overall the project will deliver "something useful". Yet, to others, it means that things are not quite OK but "I just can't put my finger on the specific issue just yet".

However, for the majority of experienced project managers that we have discussed the true meaning of amber, it simply means:

covering your ass.

When the project inevitably moves from amber to red (see later), the project manager when facing the wrath of a senior executive/sponsor can simply say "I was warning you … remember the amber traffic lights?"

In the organizations that we have consulted with, it is really up to the project manager (or, in some cases, the Program Office) to determine which colour to flag the project. There are few, if any, guidelines as to what each colour means and what are the precise rules for changing the project's status.

Probably, the most common interpretation is as follows:

- Green - nothing to worry about;
- Yellow – some concerns but don't worry too much; and
- Red – Worry a lot.

However, this seems to clarify what worries me about the three colour model for status[v]. I have always stated that when involved with projects "It pays to be paranoid".

A radical rethink of traffic lights

Many years ago, I read a *Scientific American* article about Rene Thom, a French mathematician, who had developed a concept called Catastrophe Theory[vi]. In the *Scientific American* article, Thom's model was used to explain how two dogs behaved during a confrontation. In effect, as long as both dogs kept their ears up, it was just two dogs barking at each other. However, as soon as one dog flattened its ears, a "catastrophic event" occurred and the dogs were now in a fight to death.

Since that article, I have observed the same "catastrophic event" behavior in every project I have consulted on. In addition, thousands of project managers who have attended our Agile Project Management workshops also confirm their experience.

Consider the following common project events:

- the sponsor changes the scope and objectives of the project;
- the sponsor leaves the project for a new sponsor to pick up;
- your key business analyst leaves your project;
- the project budget is cut;
- the schedule is shortened; and
- a critical stakeholder moves their priority and support to another project.

The question as to whether your project is now green, amber or red is simply the wrong question.

Using Thom's concept of catastrophe, your project really exists in two states or, as shown in Figure 11 below, *you* (the project manager) have the power, financial delegation and resources to deal with the problem i.e. it is In Control (Green) or you don't and will have to ask someone else (sponsor, your executive) for assistance is making the

necessary decision to get your project back onto the In control plane – until then, your project is Out of control (Red).

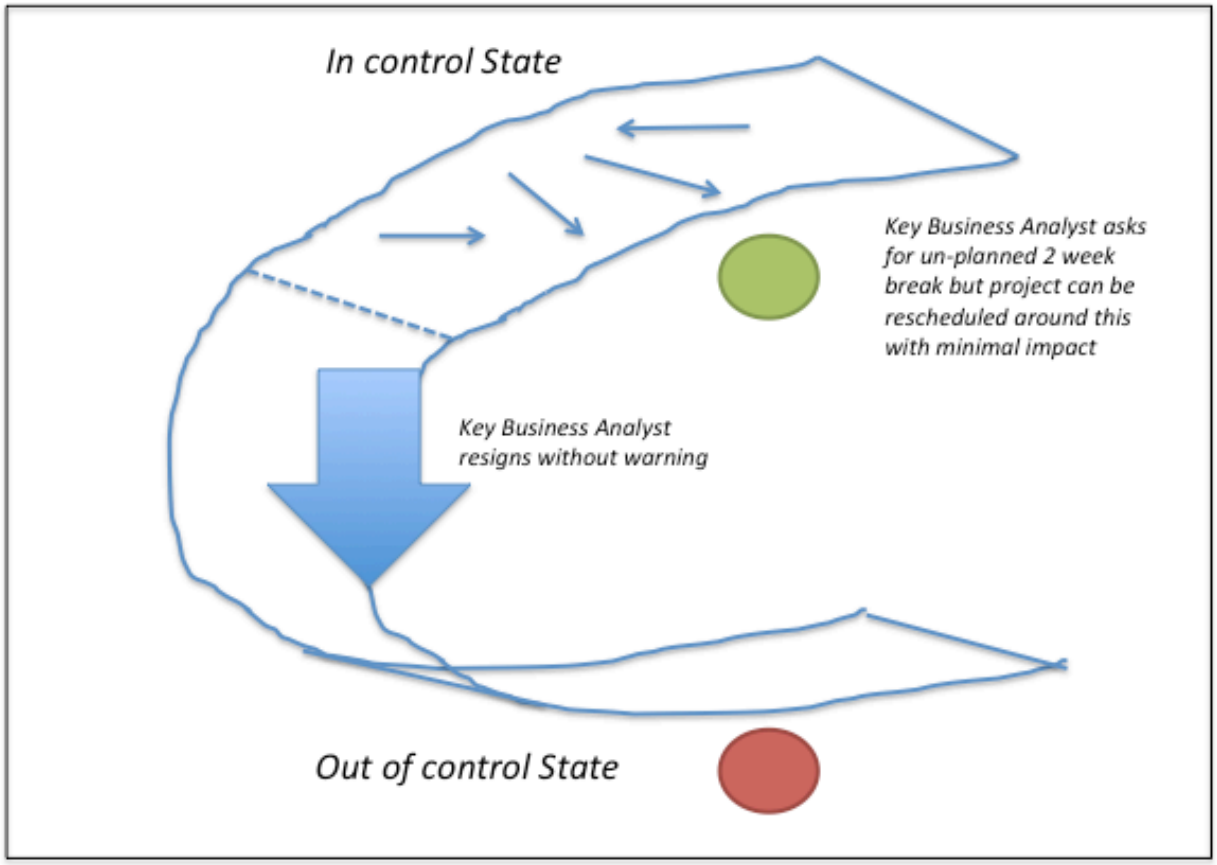

Fig 11 – Catastrophe Theory and project states

The real behaviour of projects is that they can move quite significantly around the "In control" plane. As long as you can take the necessary actions to keep your project on that plane you are in "Green". However, as shown in Figure 11, just as when one dog lowers its ears, there will be events such as the ones above that move your project to the "cusp" where it *immediately* changes state to "Red". Welcome to the real world.

An agile traffic report

So we have a simpler and more useful model of project state and project status reporting.

Fig 12 – Agile traffic lights

While you struggle with this model and perhaps reject it, let me update you that we have 4 major corporations using this two colour traffic lights approach. As one very senior executive, who had sponsored many major projects, said to us:

"You know, I was always suspicious of that amber light. I never know how to interpret it so I always assumed it was Red. That just avoided surprises for me."

Just think about it please.

References

Capers Jones, *Applied Software Measurement* (2nd ed). New York, N.Y., McGraw-Hill, 1996.

Rob Thomsett, "Managing the new Information Technology Revolution", *Reinventing IT: Ed Yourdon, Larry Constantine & Rob Thomsett Seminar*, Australian Computer Society, 1995.1

Rob Thomsett, Project Pathology: Causes, patterns and symptoms of project failure, *American Programmer*, 1995.2

Rob Thomsett, *Radical Project Management*. Upper Saddle Creek, N.J, Prentice-Hall, 2002.

---

[i] Gantt Charts were developed by Henry Gantt to assist in managing logistics during World War One.

[ii] Alternatives are " Pretty good", "OK" and "Allright"

[iii] Fans of Scott Adams *Dilbert* cartoons will remember a classic where Dilbert's boss is asking how things are going. Dilbert explains how much a mess his project and life is in. The boss replies that it is more customary for Dilbert to reply "Fine".

[iv] I am now convinced that the arrival of traffic lights could be another example of the Poonsbain Principle 1 that I discuss in my Blog section.

[v] In a great movie "Starman", Jeff bridges plays an alien who has taken over the body of a dead person. He quickly learns that Amber traffic lights mean "drive faster".

[vi] Wikipedia.