

# Successful Projects

BY PETER TAYLOR, ENABLEMENT MANAGER, COGNOS, BRACKNELL, UK

## WHAT DO WE MEAN BY SUCCESS?

How do you define project success? Customers typically refer to successful projects as being on time, on budget and to specification, enhancing their business practices and processes, returning specific measures of ROI.



## CRITICAL SUCCESS FACTORS

Based on the above definition, what are the critical factors for success? Many factors influence the outcome of a performance management project, and there are specific areas of risk – or to put it another way, significant success opportunities associated with each one. Here those “opportunities” have been categorized under five headings – People, Data, Design, Technical Infrastructure and the piece that glues it all together ... Project Management.

For customer implementations to be successful, Cognos needs to be good at all five of these things. Let’s take a look at some of the considerations in each area.

### PEOPLE

*Many people across the enterprise will have an interest in a performance management project.*

- Executive team – in most cases, the executive team commissioned the work and is ultimately responsible for its success.
- Business user community will be the main beneficiaries of the development – it will increase the value of the work they do.

- Project development team.
- Other IT teams whose help and support will be required – for example, the database teams who look after the source systems, and network and infrastructure teams.
- The development may ultimately affect the services provided to external parties such as customers and suppliers.

Addressing the interests of all these varied groups is clearly quite a challenge!

### *Manage the interaction with other groups of people*

The point to emphasize here is that it’s not just Cognos people that you need to work with to make your project successful. So keep an eye on the bigger picture. Be aware that you are not working in isolation but are dependent on other teams carrying out their responsibilities to support the project. The most important thing is to establish a clear understanding of responsibilities – don’t make assumptions about what others will do.

Try to anticipate and plan the time you will require from other teams. Be aware of other team’s priorities and keep those teams informed of what is happening on the Cognos front, so that it doesn’t come as a surprise when you ask them for help.

Of course this sort of thing is often communicated informally in an organization – around the coffee machine or in the lunch room. Always remember that a little networking can go a long way in progressing your project!

### *Focus on the business requirement*

A clear understanding of the business requirement is fundamental to success, as this understanding feeds straight in to the application design. If we don’t fully understand the requirement, there is a high probability that we will have to go back and redesign the application at some stage. This might not be a problem if you are building a couple of cubes for a Proof of Concept, but if you are building a sizeable data mart and hundreds of cubes and reports it becomes much more of an issue.

**COGNOS®**

THE NEXT LEVEL OF PERFORMANCE™

There is nothing new in saying that user requirements need to be clearly understood – any Cognos customer will have heard this, time and time again – but nevertheless it remains high on the list of things that are frequently not done well. There may be many reasons for this – the users might be very busy and are reluctant to allocate time, or if they are new to the products they could find it difficult to conceptualise the potential application. However it is worth persevering, as it is often the most critical success factor.

Talking to the users in business terms can often help to make user interviews or workshops more productive – requirements should wherever possible be couched in business terms, e.g. “What is the pattern of orders over a six-month period?” rather than “I want a report with the following columns...”

One thing to avoid is making assumptions about the user requirements. We all work in the business so we all have a view about what we think is important, or maybe someone on the team used to work in the business and you use them as your source – however, the business and the user requirements may have changed since then. The basic principle is don't second guess what the users want – get them involved!

More controversially, at the same time as interpreting the requirement, for best results aim to play to the strengths of the software. Sometimes a business will spend a lot of time prototyping – sometimes in great detail, often using some easily available desktop tool such as Excel, and finish up with a set of very complex presentational requirements, which they then want to reproduce using the Cognos tools.

Sometimes there are completely valid reasons why a company takes this approach, but it should always be challenged if it requires a lot of workarounds, or significant extension of product functionality, as it is almost certainly storing up trouble for the future in terms of update and maintenance, as the business changes.

Also, if the application is complex, the business community might simply choose not use it.

Once you have the user requirements, you need to review them regularly with the users throughout development to confirm that you have got it right, and to make sure the users have adequate training and support to make full use of the application.

Training needs to be at the appropriate time to allow the users to put their new skills to use right away.

### *Appoint a project sponsor*

Most successful BI projects have benefited from the appointment of a project sponsor – usually a senior person in the organization, a visionary who can see the long-term benefits and is committed to the goals, and a known influencer who will sustain motivation and remove any cultural or political obstacles encountered. For example, one project, which required data from 26 countries, all with different systems, offered a significant challenge. Just getting the data would have been a real headache, but the project sponsor – in this case the Managing Director - was able to move the project along by making a personal call to each country to make it clear that he wanted the full cooperation of everyone. So – appoint a sponsor!

### *Build a strong team*

A strong development team is, of course, a critical factor in any project. The team needs to comprise the full spectrum of good technical skills for efficient development and good interpersonal skills for working alongside users and interpreting their requirements.

Each member of the team should have a clear understanding of the development approach.

The cost of training is often a perceived issue if there are no existing skills in the organization. But these costs more often than not repay themselves quickly in terms of higher quality development work and early self-sufficiency.

## **DATA**

### *Understand the data*

Clearly, the people involved in the project play a vital role in its success or failure, but what about the data? This is a big subject in itself, so all I want to do here is emphasize one or two data-related issues that can severely impact the success of a project

Data issues are probably the greatest cause of project slippage, so it is well worth doing what you can at the outset to avert problems.

Once you have understood the requirement, spend some time on a data sourcing exercise. If the data doesn't yet exist, plan it in for a later phase of development. If it has to be in the current phase, show it as a high risk on the plan. Experience shows that it often takes a lot longer than anyone expects to capture



**THE NEXT LEVEL OF PERFORMANCE™**

additional data – and frequently involves the development of additional processes for update and maintenance. You really can't do much without data – a certain amount of design and prototyping perhaps but no true testing and validation.

Achieving consensus among the business community on the definition of business terms can be quite a challenge, particularly when discussing terms such as Margin, for example, where you might have a very different view depending whether you work in Sales or Finance. So allow plenty of time for these issues to be resolved – or get the project sponsor to make the final decision.

You should also take a hard view at the quality and complexity of source data. Issues around poor data quality, in particular, can take a disproportionate amount of time to resolve, and might even require amendments to the source systems and extended testing. You also need to have a strategy for dealing with any structural complexities in the source systems – and you'll likely need help from the DBA to do this.

## **DESIGN**

### *Align design to requirements*

Design is another important area that needs to be addressed when planning a project. Here we will provide a few tips for success. First and foremost, the design should be driven by the user requirements, not by the data, focusing on the specific business decisions to be made.

You have a range of design techniques available to you – make full use of them when designing a data mart. An Entity Relationship Model captures the enterprise business rules in a clean and logical way and is a very useful aid to understanding for developers, and during any staging process you might choose to store data in a relational structure.

For the presentational layer, a dimensional model – which reflects the way the information is to be used – delivers the best performance and is easy for the users to understand.

Lastly, document the design – in particular, noting the reasons behind any specific design decisions, so that anyone else joining the project can quickly understand how it all works. Good documentation also allows you to refresh your memory quickly if called upon to implement enhancements a year later.

### *Manageability*

Sizing and Manageability is another critical factor, especially for a first-time implementation. The old maxim of keeping things simple is as valid here as anywhere – and any unavoidable complexity should be transparent to the user. Business users will soon abandon a reporting system that is difficult to use or is slow in response.

A valuable principle is to start with a pilot development of manageable size and targeted at a specific user group. For a global project, this can require some politically uncomfortable prioritization, but from experience a “big bang” approach is likely to cause far more issues. Again, the project sponsor can be a useful intermediary here.

Once you have your design, you should produce a formal sizing estimate to feed into an infrastructure review.

## **TECHNICAL INFRASTRUCTURE**

### *Review technical infrastructure*

Following the design phase, you need to review the technical infrastructure to ensure it has adequate capacity. Usually the technical infrastructure team is completely separate from the BI project team, and most of the success opportunities here lie in the handling of dependencies between teams. One difficulty here is that in order to sign off the development budget, an estimate of required hardware and software is required, and until the design is complete it is very difficult to make these estimates with any certainty. In reality, the estimates are usually made on the basis of previous experience. Where usage patterns are unknown (and they usually are) – plan in regular testing and review, and involve the technical infrastructure team in the testing process.

If there is an incompatibility between the requirements and the technical capacity, and there is a budgetary constraint, be prepared to negotiate a compromise with the users – in this case, the issues should be carefully documented.

## **PROJECT MANAGEMENT**

### *Be realistic about development time*

Probably many of you will relate to this point ... scheduling your project from a fixed end date and working backward happens very frequently, partly reflecting the fact that performance



**THE NEXT LEVEL OF PERFORMANCE™**

management development is usually part of a much wider strategic change for an organization, where initial planning is at a fairly high level. If you can argue the case for planning forwards, not backwards, you'll remove the most critical obstacle to completing the project on time!

Be clear about the purpose of the plan – it should be a useful tool, not a burden. Sometimes a plan takes on a life of its own and becomes increasingly burdensome administratively, without adding value. Often this is because the plan is a way of communicating progress, but there are often better ways, e.g. a weekly progress report. If you feel your project plan is taking over your life, take a long hard look at why – it can be a sign that the project manager is too far removed from the project and doesn't know what is going on. That sort of plan can be a significant source of friction within the team.

Don't forget to add in contingency – again we sometimes allow our estimates to be influenced by what we think is expected rather than by the time we genuinely expect things to take, but if we secretly feel the plan to be unrealistic there is every chance it will turn out to be exactly that!

Remember those dependencies on other teams and on the business users, and plan in their time too.

Keep in mind that development is a continuing process, not a one-off exercise, and plan in iterative cycles – making use of this format as a technique for avoiding scope creep. If time and resources are fixed, be prepared to compromise on functionality. And though it is easier said than done, it drives home the fact that if you don't start out with a realistic plan, you are going to struggle throughout the project. Establish a formal change control process to help to keep the project on track.

### *Identify and Manage Risk*

Risk management is an often-overlooked project activity, possibly because it is perceived as a negative function. However, in reality it is a crucial part of good planning. Analyzing potential risks at the beginning of a project often means that people are more comfortable with handling issues later on because they have been foreseen and a strategy is in place to address them.

Encourage project team members to identify any risks as soon as they arise. Nobody wants to be the bearer of bad news – but bad news early on is much better than bad news in the later, critical stages of a project. Company and team culture have a big

influence in whether people are comfortable expressing what they may see as negative feedback, but it is in the project manager's interests to know about anything that might impact the initiative. And be sure to remember the potential value of the project sponsor in this kind of situation.

Accept that you can't necessarily find a perfect solution to every issue – some things you just have to live with. Some situations require special care, such as Merger situations, where there might be a culture clash between the different parties. You can't instantaneously resolve everything but you can put some thought into how to handle issues that are beyond your sphere of influence.

### *Manage Expectations*

Managing expectations is also key to success. Keep Executive Management in the picture (this is easier if there is a steering committee in place) and ensure sustained awareness and commitment by regular progress reports.

If things are going wrong, don't just try to bluff your way through and hope for the best. Instead, bring your risk management strategy into play – demonstrate that you are dealing with the situation. You will earn more respect that way.

The ultimate success of the project, not to mention the credibility of the project team, depends upon gaining the confidence and trust of the users, so keep them involved and let them know what is going on.

## **THE BALANCING ACT**

So, to sum up: Successful implementations require the careful balancing of a number of sometimes contradictory factors:

- Get the best advantage from your software but remain focused on the business requirement.
- Be aware of any limitations in technical capacity.
- Strike an acceptable balance between innovation and risk (both in design and in user functionality delivered).
- Spend time at the beginning in order to save it later.
- Think about cost in terms of the value you are adding to the business.
- Manage expectations carefully.
- And remember that a successful BI project needs a wide range of skills.



**THE NEXT LEVEL OF PERFORMANCE™**

