



IT PROJECTS – DO WE STILL HAVE A PROBLEM?

The global information technology and communications sector supports millions of professional workers in a wide range of industries and occupations, a large proportion of which are engaged in the design and delivery of information systems and technology projects. **John McManus** examines if there is still a major issue with regard to IT project management or if the situation has improved.

In the last five years the European Union has written off billions of Euros in failed information systems and technology projects. Yet, despite dwindling stakeholder and management confidence in information systems and technology projects, huge sums of money still get spent, and projects still get written off. Research by the author highlights that only one in eight information technology projects can be considered truly successful (that is meeting the original time, cost and quality criteria). Based on prior research conservative estimates put the cost of project failure at 142 billion Euros across the European Union. Yet while our understanding of the importance of project failure has increased, many of the underlying reasons for failure still remain an issue.

Does methodology matter?

The way projects are managed is generally conceived in terms of philosophy and methodology. When undertaking information technology (IT) projects, and software development, a number of different approaches and methodologies are used. The most common combination in use is the waterfall method supported by PRINCE2¹. Research by Dr John McManus and Dr Trevor Wood-Harper identified 80 per cent of IT projects undertaken in Europe adopted this combination for software development and project delivery. Following a project management methodology, such as PRINCE2 helps project managers and those involved in organising and delivering IT projects and software methodologies to assist developers in design and other technical

activities, but methods can become an almost insignificant factor in the face of stakeholder and personal politics.

The notion that project management methodologies like PRINCE2 provide everything an IT project needs is false.

First, I would question whether methodology is the single most important aspect of an IT project, to which all else is subordinate. In the view of Peter Kemp and myself, an IT project requires three things of equal importance: a project strategy/plan, a technical solution and a software engineering methodology. The technical solution (not the plan) demonstrates that the objective of the project is viable, and the methodology (not the plan) defines how the components of the solution are built and integrated.

Some practitioners will argue ‘that today’ we are dealing with more complex technology and requisite variety than ever before.

A common technique in project management is to state clearly all assumptions and ensure that the project takes these assumptions into account.

One might expect these assumptions to be realistic (even if they turn out not to be true), but increasingly it seems projects are willing to state and work to wholly unrealistic assumptions. Often these assumptions are adopted in order to provide a project plan that meets the required timescales and budgets. The sequence of events that seems to be increasingly common is that the initial project plan is either too long, too expensive and full of risk. So, the project manager asks what assumptions can be made that would improve things. Clearly this can lead to innovative or creative solutions. The danger, however, is that it leads to a set of completely implausible assumptions being made and the project being planned on this basis. The project plan, on paper, still stands up, but it has lost all relationship to reality. In many cases the assumptions are not only unrealistic; they are actually the opposite of what one would expect. So, in effect, the project is being planned on the basis of

assumptions that the project team knows to be false and in consequence leads to projects being abandoned or cancelled (refer to table below).

Reasons why projects get cancelled: A further consequence of unrealistic assumptions may be to exclude or marginalise those who understand the subject matter. To put it bluntly, someone who does not understand software development and processes is more likely to be comfortable with unrealistic assumptions about it than someone who is an experienced systems integrator.

Regretfully, a project may have to proceed without constructive input from its software experts because the assumptions made are only acceptable to those who don't understand the subject matter. IT specialists either leave the project or become so disenchanted that they no longer try to influence things. The very people who understand how to design and build IT systems become even further marginalised or sidelined altogether.

Things change, but do we change?

Some practitioners will argue ‘that

today’ we are dealing with more complex technology and requisite variety than ever before. However, the level of skill required by architects and developers decades ago was arguably just as high, albeit different, than that required today. It could be further argued that the more things change in software development, the more certain aspects of them should stay the same. There's a tendency among younger developers to completely discard anything from ‘ten years ago’ as being laughably irrelevant. Take for example, software quality/inspection. Even after more than 40 years and vast amounts of published data supporting their value, I find that only a small minority of people are even familiar with the concept of software quality/inspection, let alone practicing them, and practicing them well. This provides a continuing market for consultants and trainers, but only if the prospective customers know enough to ask for help. Some years ago, I spent ten months in the financial services sector leading process improvement activities in the web development. The people employed in this web development group were very smart people but it was interesting to see that in their migration from mainframe to client/server to web they had forgotten pretty much everything the industry had learned about configuration management. In some ways, running a web server is like running a mainframe if there is only one instance of the executables and supporting files. Therefore, it would seem that the usual configuration management (certainly change control) practices would still apply.

But instead, the group were using very haphazard software practices that led to the predictable problems such as those outlined above. To conclude, it's sometimes difficult to sell a better mousetrap to someone who doesn't know he has mice.

Reasons why projects get cancelled: European Study into Project Failure²

- Ability to adapt to new resource combinations;
- Differences between management and client;
- Insufficient risk management;
- Insufficient end-user management;
- Insufficient domain knowledge;
- Insufficient software metrics;
- Insufficient training of users;
- Inappropriate procedures and routines;
- Lack of management judgement;
- Lack of software development metrics;
- Loss of key personnel;
- Managing legacy replacement;
- Poor vendor management;
- Poor software productivity;
- Poor communication between stakeholders;
- Poor contract management;
- Poor financial management;
- Project management capability;
- Poor delegation and decision making;
- Unfilled promises to users and other stakeholders.

References

1. *Projects in a Controlled Environment*
2. See *ITNOW*, April 2009
3. *Research conducted by Dr John McManus and Dr Trevor Wood-Harper, 2008*