WHY MOST "IT PROJECTS" ARE REALLY I.T. WITHOUT THE PROJECT.

Paper presented at the 3rd World Project Management Conference Gold coast 2002

John R Smyrk BEc (Hons), MEc (OR) (Monash), MACS, CMC.
Principal: Sigma Management Science Pty Limited.
Visiting Fellow: National Graduate School of Management, ANU.

INTRODUCTION.

Background.

The success rate of so-called "IT projects" is stunningly poor. Various studies point to failure in over half of all such exercises. Continued acceptance of this track record by the business community lies in stark contrast to the expectations of, say, the construction industry where such a level of performance would not be tolerated.

Many causes for this sorry state of affairs have been proposed. In this paper I will explore the hypothesis that the very concept of an "IT project" is unworkable, and should not be employed in most business situations. I then go on to examine an alternative approach and discuss its implications for business in general, and for the IT profession in particular.

I propose that there are a number of fundamental flaws in the methodologies surrounding IT projects—falling into three broad categories:

- inadequate scoping;
- confusion between business engineering and software engineering; and
- a "solutions" mentality.

Together these flaws almost guarantee that any venture undertaken as an IT project will fail.

My proposals to deal with this problem are based on a new business initiative model, which has profound implications for the way that projects involving IT are undertaken in the future.

The characteristics of an "IT project".

To qualify as what is commonly identified as an "IT project", the consensus view appears to be that that an initiative needs only to involve the delivery of systems or technology. The conceptual problems with IT projects begin with their very definition—if we used that type of taxonomy elsewhere, we would have to conclude that it is only necessary for a car to contain plastic for it to be classified as a plastic car. Ward and Griffiths (1996, 441-489), provide a comprehensive picture of the conventional IT project.

Once a project is labelled in this way, it then, according to conventional wisdom, it has to be planned and managed under a framework which, amongst other things:

- sets an objective of "producing/implementing/installing a system";
- identifies IS and IT as the primary outputs—often describing them as "solutions";
- accepts software engineering as the underpinning methodology in which business stakeholders are described (somewhat dismissively) as "users". "Users" formally engaged in the project by having them "sign-off" some form of project brief; and
- relegates business outputs (if they are recognised at all) to a subordinate role—for example, "we have to reengineer our processes to get the system in".

It is intriguing that forty years of IT project failure have led us to the view that the approach is right—it is just our application of it that is wrong. Few participants in the IT profession have jumped to the more obvious conclusion that the methodology itself is wrong and shouldn't be used.

Classes of projects involving IS and IT.

Projects involving the delivery of information systems (IS) and information technology (IT) can be grouped into four broad classes, according to the type of IS/IT-related output that is being produced:

- development of a product for sale (or incorporation into a product for sale)—a commercial accounting package, for example;
- development of systems components required by a business project;
- assembly of technical infrastructure—upgrading the office's LAN cables and network cards falls into this class; and
- configuration/installation/implementation of applications systems—such as a CRM (Customer Relationship Management) package.

The first and second of these are the classical domains of software engineering. These exercises are clearly and unambiguously true IT projects. It must be noted, however, that in the second case, software engineering is relevant only to the systems "bits" of the project, and not to the business components. The third are more like construction or engineering exercises, and should be run as such. The IT project approach has absolutely nothing to offer here. The last group is the target of this paper.

IT PROJECTS AND THE SCOPING PROBLEM.

The product-process dependency model.

A discussion of business projects involving IS and IT requires a clear view of how systems and technology fit into the organisation's operational environment. I propose the following Product/Process Dependency Model.

WHY MOST "IT PROJECTS" ARE REALLY I.T. WITHOUT THE PROJECT.
Paper presented at the 3rd World Project Management Conference Gold coast 1992
John R Smyrk

Layer	Description	Elements		
Outputs	The firm delivers outputs to its customers.	**Products**	**Services**	
		These two layers are completely interconnected		
Process	Products and services are designed, assembled, delivered and supported through the firm's processes.	**Business processes** (such as Procurement).	**Functional processes** (such as the regular activities of HR)	**Decision processes** (such as those involved in strategic planning)
		These two layers are completely interconnected		
Enablement	The firm enables its processes by: describing how they are done, assigning staff to them and supporting them with technology.	**Business rules** (describing how things are done—for example a manual of standard operating procedures)	**Organisational model** (describing who does what—defining roles, responsibilities and reporting lines)	**Applications systems** (Such as an ERP) & **Technical infrastructure** (such as a LAN)

The Product/Process Dependency (PPD) Model.

The model views an organisation's operational environment as made up of three Layers (shown as rows) and eight Elements (shown as shaded cells). It is used in two distinct modes: top down—to show the nature of dependencies between the elements of business operations, and bottom up—to show the impacts of isolated changes to any of those elements.

The top down interpretation.

The arrangement of the Outputs and Process Layers makes it clear that, to create, produce and deliver products to customers, an organisation must have an appropriate set of processes in place. These processes are of three kinds: business ("end-to-end" work), functional ("support" work) and decision ("directing" work). Amongst other things, processes:

- take the form of structured activities, some carried out by staff, others by technology;
- consume resources; and
- are controllable.

Any proposed change to the organisation's product/service portfolio can only be effected by reengineering relevant processes. Processes, in turn, are enabled by the elements shown in the Enablement Layer: Any desired process change must be accompanied by complementary changes in business rules, organisational arrangements and supporting technologies.

The conclusion we must draw from the model in top down mode is this: "**Any desired change to the Product/Service or Process Layers, cannot be achieved without also making complementary changes in the Enablement Layer."**

The bottom up interpretation.

The model also shows that change cannot be isolated to the Process and Enablement Layers. Because each Element is constrained by those in the Layer below, any change that begins in the Process and Enablement Layers will result in undirected impacts on all Elements in the Layer above. By "undirected" impacts, we mean that they will be quite random in their effect and thus more likely to damage the Element on the receiving end than to enhance it.

Take, for example, the case of a firm that decides to install a new software package, but which has not decided on the future shape of its processes. When implemented, the new system will force changes on the organisation's processes—possibly damaging them. The altered processes will, in turn, force changes on the quality of products and services—possibly degrading them as well.

The conclusion we must draw from the model in bottom up mode is this: "**Changes in the Process and Enablement Layers cannot be isolated because they will automatically impact elements in the Layers above. Changes that begin in the Process and Enablement Layers can have undesirable impacts on organisational performance."**

Implications for "IT projects".

The PPD Model has a serious implication for IT projects. Projects that are either limited to the delivery of an application system, (or treat process change as subordinate to systems change) are most likely to result in a degradation of processes and a lowering of organisational performance. IT project methodologies are unable to deal with the scoping demands of business projects. Not only are they unable to detect and prevent inadequate scoping, but worse still, but they also appear to actively encourage it.

This then brings into question the strategy of "using IT as a driver for change". The PPD model predicts that such exercises will be indistinguishable from conventional IT projects in terms of their outcomes (which are just as likely to prove unsatisfactory).

Implications for Return on Investment in IT.

But there is an even more profound implication. If systems projects make no sense because they are incomplete, then we must also conclude that the concept of investment in systems also makes no sense. This then implies that one of the concepts held most sacred by the IT profession—that of ROI in systems—has no meaning. ROI can only be determined for entire projects, not for the individual components of projects. This view runs completely counter to accepted wisdom as described, for example by Willcocks and Lester (1999, 551-572). The situation is similar to that for, say, a transport company. It makes sense to measure the ROI in trucks, but it makes no sense to assess the "investment" in front wheels in isolation from the assessment of the entire truck.

WHY MOST "IT PROJECTS" ARE REALLY I.T. WITHOUT THE PROJECT.
Paper presented at the 3rd World Project Management Conference Gold coast 1992
John R Smyrk

IT PROJECTS AND THE PROBLEM OF A SOLUTIONS MENTALITY.

The ITO model.

My Input-Transform-Outcomes (ITO) Model offers an explanation of how projects and outcomes are related.

Input	Transform			Outcome
➔	➔			➔
INPUTS	**PROJECT**	**OUTPUTS**	**UTILISATION**	**TARGET OUTCOMES**
Resources: o Acquired product. o Team labour.	The work of the project.	The project's deliverables.	Utilisation of outputs by the project's customers.	Target outcomes = benefits.

The Input-Transform-Outcome Model.

The model explains the (eventual) generation of benefits from execution of an approved project in the following way:

1. inputs (resources) are acquired and deployed to the activities defined in the project's Work Breakdown Structure;
2. the team undertakes the work of the project—so as to produce the outputs (deliverables) defined in the project's scoping statement;
3. outputs are delivered to the project's customers;
4. the project's customers utilise the outputs that have been delivered to them, and in so doing, generate the target outcomes that have been defined in the project business case; and
5. when target outcomes represent a "flow of value" to a beneficiary, we call that flow a benefit.

Some observations on the ITO model:

- we assume that the project is controllable, and so outputs can be guaranteed;
- the project manager is accountable for delivery of outputs that are fit-for-purpose to customers;
- realisation of target outcomes is not guaranteed, because we can't control the way utilisation occurs. In fact, if circumstances are against us, we may get no outcomes at all;
- the person who is held accountable for the realisation of target outcomes is called the project owner. In general, the project manager is not accountable for target outcomes; and
- the project manager is the supplier of product/services to the project owner—who is the client.

It should be noted in passing that, according to accepted business practice, the client always commissions the supplier—not the other way around.

WHY MOST "IT PROJECTS" ARE REALLY I.T. WITHOUT THE PROJECT.
Paper presented at the 3rd World Project Management Conference Gold coast 1992
John R Smyrk

The business case and the ITO model.

A project is scoped in the business case, by working through the ITO model, right-to-left. This means: setting target outcomes, analysing how customers will utilise a defined list of deliverables, describing the broad work of the project. and estimating the resources that will be required.

The business case then defines three critical project relationships:

1. the funder makes a budget available to the project owner in return for the eventual realisation of target outcomes;
2. the project owner then "contracts" with the project manager to deliver the project's outputs, in return for underwriting the project's outlays; and
3. the project manager buys resources from suppliers and team members.

The nature of a "solution" to a business problem.

According to the ITO model, in a project that seeks to solve a business problem, the solution to the business problem is an outcome—not an output.

Take for example the case of a firm faced with the problem of rising production costs for a key product. A project to address this issue would typically be centred on manufacturing and logistics processes, and would also involve new systems. Before going any further, we need to separate out the outcomes and outputs from this exercise. Target outcomes will be expressed in terms of reduced productions costs. Committed outputs will include new processes and new systems to support those processes.

We can now assign responsibilities for these outputs and outcomes: The project team would be responsible for delivering reengineered processes, supporting systems, agreed organisational changes and change management programs. Specific business areas would be made accountable for realising the target cost reductions by utilising all the projects outputs delivered to them by the team. In other words, these business areas (and not the project team) would be responsible for solving the original business problem.

IT "solutions" to business problems.

How are "solutions" treated in IT projects? IT project methodologies accept the concept that a solution can be either delivered as a project output, or acquired as a project input—as evidenced by casual observation:

- the systems deliverables of IT projects are called "solutions"; and
- IT vendors claim that they sell solutions—not products.

From the perspective of senior executives who are under pressure to generate target outcomes from business projects, the message from the IT industry (supported by the IT profession) is intensely appealing. By adopting the "IT project" approach, I can buy outcomes—I don't have to go through the pain, effort and uncertainty of engineering solutions from a business project.

Of course, the consequences for the project are tragically predictable. What is purchased from vendors is product, not solutions, but because the utilisation part of the ITO model has been scoped out of the project, there is no mechanism for defining, and facilitating outcomes (much less for assigning accountabilities) and so benefits never appear and the initiative fails.

CONFUSING SOFTWARE ENGINEERING WITH BUSINESS ENGINEERING

The methodological underpinnings of IT projects.

During the early years of business computing in the 60s, the most common projects were directed at simply "automating" manual processes. The target outcomes concerned with cost reduction, and the key deliverable was a faster, less labour intensive version of the old manual process.

Given that (very simple) project environment, the most novel, challenging and difficult outputs to produce were the computer systems. Changes to business rules, processes and organisational structures appeared relatively simple and easy by comparison.
Progressively, the DP industry (as it was known then) developed specialised methodologies to make the production of application systems more predictable, repeatable, faster and less costly. Thus, out of the first IT-oriented business projects emerged the discipline of software engineering. Soon a whole new software products industry evolved—with software engineering as one of its cornerstones.

The nature of business projects changed—gradually becoming less concerned with the automation of manual processes, and more concerned with innovation. Software engineering, however, was becoming more entrenched. The IT profession, with its focus on systems, became the standard bearer for the methodology—seeing business projects as little more than systems projects that just happened to have a business flavour.

But there is a fundamental issue here. While software engineering is a methodology for engineering software, business projects, require business engineering methodologies—approaches that support product and process innovation.

Three aspects of software engineering are particularly unsuited to business projects: the concept of a "user", user requirements and the procedure of "getting users to sign off".

The concept of a user.

When developing software, the concept of a user probably makes sense. A system interacts with a human who is seen as using that system. But in an IT project, the concept of a user is carried across into the business environment and so the only stakeholders whose involvement is recognised directly are users. Furthermore the only role of users is to specify, sign-off, test and take delivery of systems.

The conventional view can be expressed in this way: "In a business project, the key deliverable is a computer system. Any process is there to enable the computer system". My proposition is that is the converse is true. "In a business project, the key deliverable is a process—any computer system is there solely to enable the business process.

WHY MOST "IT PROJECTS" ARE REALLY I.T. WITHOUT THE PROJECT.
Paper presented at the 3rd World Project Management Conference Gold coast 1992
John R Smyrk

The key players in a business process constitute a much richer and more complex stakeholding than being just users of the supporting system.

Amongst the key stakeholders in a business project are the organisation's process agents. Process agents are those who participate in the execution of a process. Their role might well include certain steps that involve using a system, but that is only a minor part of their participation in the overall process.

The role of user requirements.

IT projects recognise that the systems they seek to deliver must be specified in some way. The accepted approach to this issue involves developing a "user requirements specification" by asking "users" what they want the new system to do. But there are some serious difficulties—not only with identifying users, but also with preparing a cohesive, meaningful specification.

Presumably users are people who operate the proposed system. But if a system is not yet specified who are these users? How are we to classify those who might have a lot to contribute to such a discussion but who will never actually operate the system. In what sense are they users?

But even with some sort of community of users identified, the problems of developing their requests into a requirements specification abound. All collective lists of user demands include requirements that are: inconsistent, mutually exclusive, incomplete, unrealisable and infeasible. It appears unreasonable to expect a disparate group of stakeholders to produce a clear, unified and precise specifications for a system, when they do not share a correspondingly clear, unified and precise vision of the processes that the system is to underpin.

Eventually the IT function has to step in to resolve these problems—with the result that the users are alienated.

If this sort of problem was restricted to the odd project, then we would conclude that sometimes, if not applied correctly, the approach can go wrong. When the evidence suggests that user requirements specifications fail in most cases, then we have to conclude that the approach is flawed.

Getting users to sign off.

In an IT project typically the IT function is the supplier of the IS and IT and the community of users is the customer. IT project methodologies require that, when the IT function has sorted out the problems with the user requirements specification, that they then incorporate it into some form of project brief (or maybe even a business case) and get "the users to sign off".

The IT industry is possibly the only one where the supplier commissions someone to become a client. Accepted practice in all other industries involves the customer commissioning someone to become a supplier.

I suggest this is a likely cause of the common phenomenon of "the users not taking ownership".

A REPLACEMENT FOR THE IT PROJECT.

I propose that a structure I call the business initiative, should replace the IT project—except where firms are developing software for sale, or for integration into other products for sale. The differences between the two approaches are highlighted in the following table:

Characteristic	Business initiative	IT project
Target outcomes	Measurable improvements in business performance.	Unclear.
Accountability for outcomes	Project owner.	Unclear.
Primary deliverable.	New products/services. New processes.	New applications systems.
Accountability for primary deliverable.	Project team.	IT function.
Secondary deliverables (driven by primary deliverable).	New organisational model. New business rules. New applications systems.	New processes. New organisational model. New business rules.
The role of the IT function.	As facilitator of ideas for process improvement. As "IS/IT subcontractor" to the project team.	As sponsor-promoter of the project.
Foundation for new systems	Process requirements.	User requirements.
Key players in project.	Process agents.	System users.
Core methodology	Business and process engineering.	Software engineering.
Stakeholder engagement	Business owner commissions the project team to supply outputs defined in the business case.	The IT function commissions the users to become customers for the new system.

Contrast between the conventional "IT Project" and a business initiative.

The business initiative is driven by a desire to enhance the measurable performance of the organisation. If no-one feels strongly enough about that goal to take accountability for its achievement, then the project cannot proceed. The IT project is driven by a desire to implement new technology. If the business doesn't feel strongly enough about that to take ownership, then the IT function will step into the breach.

The business initiative has implications for business, and for the IT profession. For business it means that senior non-IT management have to be held accountable for achieving the target outcomes of all projects—regardless of whether or not they involve IS and IT. For the IT profession, it means abandoning much of the methodological baggage that has been accumulated over a long time. It also means that IT staff have to become more facilitative in

their promotion of ideas for business improvement through technology. They have to learn to sell ideas for business improvement through IT, not ideas for IT projects.

It is business managers who face the biggest challenge with the new model. For them IT projects are very comfortable because if a project accidentally succeeds, they get the praise, and if it fails, IT gets the blame.

CONCLUSION.

Forty years of unsatisfactory performance has told us that the IT project model is a poor approach to business projects, but forty years of learnings have given us a a good idea of what we should use to replace it. The business initiative may point out the path that we now need to follow.

References.

Ward, J. and Griffiths, P.(1996) Strategic planning for information systems, Wiley, Chichester.

Willcocks, L. P. and Lester, S. (1999) Information technology and organisational performance, in R. Galliers et al (eds), Strategic Information Management: Challenges and strategies in managing information systems, Butterworth Heinemann, Oxford, Chapter 20.