

1. **Managing the risks of *buying* enterprise application technology and *engaging* System Integrators: a guide for *private enterprise and Government***

How to avoid that sinking feeling you get when you realise that your vendor and integration partner can't actually deliver what you asked for.

1.1. **Executive summary**

Organisations typically embark on enterprise application acquisition and integrated service engagements armed with nothing more substantial than a set of high level business requirements and detailed technical specification. Why? Because they are relatively easy to document and evaluate.

Unfortunately, there is little or no mention of detailed user requirements or the user interface in any of the Tender documents we've ever reviewed. This proves to be the fatal flaw in the process because, as far as the end user is concerned, the user interface is the application. Poorly defined requirements or a poor interface are the two fundamental causes of the failure of technology based strategies.

For example, in spite of the hundreds of billions of dollars spent implementing ERP and CRM systems each year, Excel continues to be the corporate chewing gum of choice, thereby undermining the corporate strategy. Why? Because users hate complex, unwieldy interfaces. If you don't get the interface right the first time you face an endless process of iteration and version releases until the application does what was initially required – and it often never does.

Think of it this way: what's the first thing you do when considering an extension on your house? You get an architect who will draw up detailed design plans for you – the size of the extension, elevation, windows, doors and integration with the existing house – everything is covered and nothing left to chance. Armed with a blueprint, you then seek multiple quotes from builders. You don't go to the builders first, do you?

IT is the only engineering discipline that operates without a complete Blueprint. And what is the consequence of this? Some 50 – 80% of all major IT projects fail – they are either late, over budget, are cancelled midway or a rejected by end users.

This blueprint driven process is the same regardless of whether you build a home extension, an airplane, a new car or a 50 storey skyscraper – but, inexplicably, not with IT projects. Most IT projects start with a high level set of business requirements that simply do not contain enough detail, or contain too much detail. It is impossible to fully comprehend and visualise the end goal from so much detail.

Therefore, most vendors will interpret the requirements in a way that matches their technology. During build, the developers interpret the ambiguous requirements based on their own limited knowledge of your business and how the application environment works.

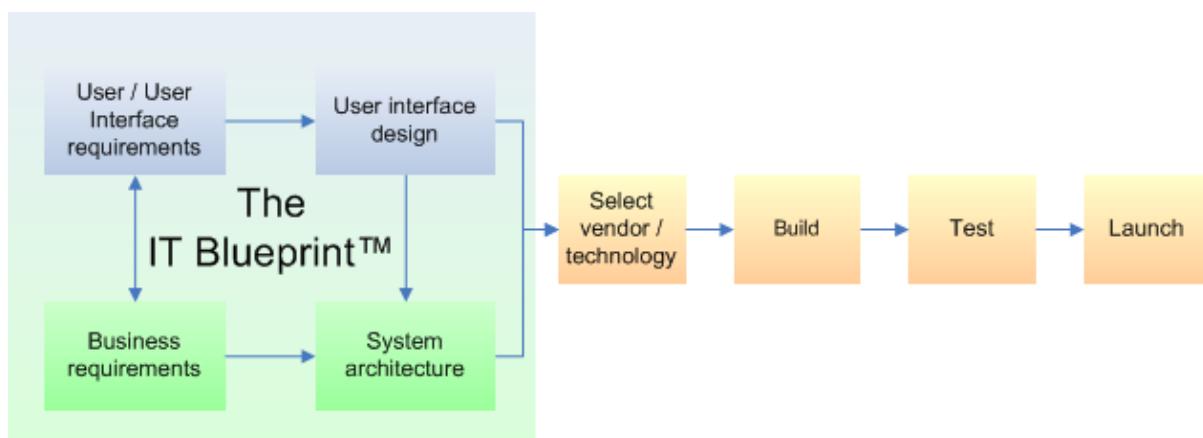
However, in reality, it is only when you can see the user interface that you comprehend the mismatch – key requirements missed, the application doesn't work the way your business does and consequently people cannot use it. The user interface governs application behaviour.

Technologists usually refer to the user interface as the 'presentation services layer' but it is much more than that. For the business, a properly designed user interface is management's means of ensuring that application behaviour reflects and drives the strategies central to the deployment of the technology. Getting the interface right ensures staff compliance with business practices because their use of the interface is both an intuitive and enjoyable experience.

Therefore, engage vendors and integration partners only after you have fully defined the business requirements, user requirements and created all user interface designs. Then you can pick the technology that will actually deliver what your users, business and customers actually want and can use – technology solutions that directly drive individual and business performance.

This kind of design specification is to IT what an architectural blueprint is to the building industry. This Blueprint is a complete and unambiguous specification of all aspects of the required solution. Prospective vendors know exactly what must be built, and can quote precisely. The Blueprint is used throughout the development process to give developers a clear view of the end goal, and to evaluate the application for compliance as it's built.

The following diagram shows how the IT Blueprint fits into the traditional development lifecycle after you've engaged your vendor. It includes the step of system architecture, which is critical for applications going into environments where there are legacy applications and prospective vendors need to know about it.



1.2. Introduction

In an earlier article, '*Rethinking the Software Development Lifecycle*', I proposed a new approach to the design and development of applications. In this article, I will show that exactly the same principles are applicable to, firstly, the selection of enterprise applications and, secondly, the engagement of systems integrators whose responsibility it is to deliver systems that contain all of the requirements, on time and on budget, and although often not mentioned, usable.

With 50-80% of all IT projects failing in some critical respect: cost, budget, functionality and user satisfaction a powerful means of making all parties accountable to deliver systems that work, the right way, first time is critical to turn this around.

Further, despite an annual investment by business in ERP and CRM systems implementations of around \$US100B, Excel spreadsheets continue to be the corporate chewing gum of choice. Take the time to look around and see just how many people record information into and perform critically important calculations on potentially 'buggy' spreadsheets and then rely on them for decision making.

The integration partner rarely gets blamed for the problems, as they pass buck to the base application and its constraints. It's the application vendor's logo that appears on the application start up screen, so it's only natural for the client organisation to view this as truth.

Let's take a look at typical vendor engagement experiences before outlining a new approach that will deliver successful IT applications.

1.3. Traditional vendor engagement experiences

In most large application development projects, there are typically two parties involved, namely:

1. A software vendor who supplies the base application, and,
2. An integration specialist who provides consulting services to define system requirements, customise the application and then implement it.

Let's cover a number of issues that occur during this process.

1.3.1. Management and its inability to precisely define what it wants

Management lacks the time and skills to precisely define what it wants from an application. While it can define the business outcomes, such as 'provide better service', 'X transactions per hour', 'platform for growth', or 'X terabytes of storage', there is a huge gap between **what** the application does and **how** it will work. It also lacks the means to express these requirements in a form that is useful to IT, and that they can directly code and implement.

As a result, the solution is often technically correct, for example:

- ❏ The hardware component can be defined in terms of its performance, such as processor speed, storage, RAM and the like,
- ❏ Defining the application component is straight forward as management can usually determine the feature set - that is, **what** it must do.

However, what management cannot specify is exactly **how** the application must work to ensure the solution is usable, requires little or no training and makes it easier for people to do their jobs. This occurs because:

- ◀ Defining the drivers of human performance is rarely in their skill set,
- ◀ Understanding the end users can be hard, as it is easy to get swamped by superficial differences between people, rather than focussing on the core activities that all people in a given role must perform,
- ◀ It is standard to assume that 'training will solve the problem' and get everyone up to speed – however training is expensive and 90% is immediately forgotten when they go back to their jobs,
- ◀ They think the vendor or system integrator can solve it, and it's their job anyway, and you do it at the end of the project, so you don't have to worry about it now,
- ◀ You can't specify these requirements up front. Anyway, they need to evolve and change throughout the development process. That's how you do development.

Management tends to think that the problems that need fixing in their business will be fully addressed by technology alone. After all, business is all about technology, isn't it?

No, it isn't; it is about people (i.e. end users and customers) and the way they think. It is also a considered blend of business, change management, people and technology. However, because of the ease of specifying them, most tenders contain precise 'technical' requirements, with little or nothing about the business, change management or people aspects.

Specifically, we rarely see actual usability and user interface requirements for the system documented in the specifications. Sometimes the tender document may include a throw-away one line requirement of 'it must be usable' or 'intuitive', but there is no specific detail about what really makes for a usable system for that organisation. As we often say, *'to the end user, the application is the user interface'*.

The way around this is to have the vendors demonstrate their software to show how 'easy it is to use'. The irony of this is that the only thing management and end users can make sense of is the user interface, yet there is **no mention of the user interface requirements in the tender document**. However, it's what management needs to see to evaluate its suitability.

Since most applications that have a specific purpose share many common features, it becomes hard to distinguish them. The decision comes down to price, relationships and promises of performance, instead of picking a system that is truly appropriate and usable. How can you pick a powerful system, if you only read material that documents these claims without experiencing the system yourself?

1.3.2. Vendor shortlists: Demonstrating software

Management spends considerable time reviewing the tender responses. Again, because of the precision of technical requirements, these are easily checked off first. Claims of business performance are reviewed as are reference sites and the user interface. A shortlist is determined and invited in for a demonstration.

The vendors demonstrate their software by showing the user interface. Notice how they don't show the code, or the memory inside the server, or the load balance statistics, even though the tender was probably full of technical requirements. Again, the only thing people can make sense of is the user interface, yet it does not appear in tenders

All vendors claim that they provide a 'superior user friendly interface', but we know from experience that the hype rarely matches reality. We have all had dreadful experiences with applications that are so poor they can't be used, even *with* training. So, where do they get the idea that it's usable? Is it because their sales team can use it, or that they determine that people can use the system following training, so it must be usable and intuitive? People mistakenly think that applications become intuitive. This means that *they've* learnt how to use it. In reality, an interface either is or is not intuitive - it doesn't become intuitive.

How many times have you seen a picture of an application's interface and when you've installed it and then actually used it, found that it was too hard to use and didn't do what you want? Here's why it's easy to be seduced by a seamless presentation:

- ◀ The demonstration is by the vendor who is an expert in the application. They breeze through and show how fast and easy it is. Never mind that she/he's been using the system for years and knows all the workarounds. After all, any system is usable after you've used it many times or been trained in it.
- ◀ The selection committee comprises the pay masters who probably won't use the application. How can they judge an application as usable if they don't do the actual work of the end users to compare it against how the application makes you do things?
- ◀ The selection group may include a 'representative' user. While this is beneficial, chances are the representative user will not get to actually use the application and try out their actual workflows.
- ◀ The demonstration usually involves demonstrating features and functions, but not doing actual workflows. They show you how this and that works, but never an end to end process that is considered best practice in the domain.

But if it does look hard to use, then any issues with the user interface and workflows are explained away by 'Oh, that's OK, we can do that. We just need to customise it here and here'. Meanwhile the integration services team is mentally adding up the change requests fees.

Despite the claims, the development process is typically characterised by iteration, multiple releases, delayed schedules. We cover more of this in our article 'Debunking the myths of IT'. Sacrifices end up being made, and the project is either canned when it is months or years overdue, or the project requires two or more times the initial investment to get it working

So there must be a better way, and here it is.

1.4. How to successfully engage a vendor and integration partner

We appreciate that organisations are often in the middle of a tender process, or have already engaged a vendor and integration partner. We provide recommendations for three situations you may be in:

- ◀ You are about to release a tender document for a new application or IT services,
- ◀ You are in the middle of a tender evaluation process, or,
- ◀ You are underway with the development.

1.4.1. Guidelines for issuing a new tender

Because strong business performance is a combination of people, business and technology, the user requirements and user interface design components need to be removed from the technology component of the tender. Technology vendors and integration partners know about technology. They do not know about business or people in the way business consultants and user interface designers do.

Consider doing the following prior to your next tender:

1. Determine the precise business and user requirements and the required user interface designs: The Blueprint

- ◀ Engage specialists to capture the precise business requirements:
 - Business processes and policies
 - Use case and UML, depending on the complexity of the application,
- ◀ Engage specialists to capture the precise user requirements
 - Understanding customer need and expectations
 - Understanding staff needs and expectations
 - Resolving these against the business requirements
 - Establishing a definitive statement of exactly how the application needs to work to allow people to achieve their goals and deliver business outcomes
- ◀ Create all user interface designs required to deliver the business processes and policies to the desktop
 - The information architecture
 - The workflows that reflect how people think and work, as well as best practice
 - All user interface designs for the entire application

These steps aggregate to what we call the IT Blueprint – a complete and unambiguous specification of what the application needs to do to support business and individual performance.

2. Go to market with the Blueprint

The Blueprint is just like a building blueprint. The builders know exactly what needs to be done as they have a clear view of the end goal.

- ◀ Release the Blueprint to market for vendors and integration partners to quote on. Because your requirements are now complete and unambiguous, you can easily determine who can actually deliver what you need to run your business effectively.

If there are confidential matters in it, then release it to a restricted group under NDA. In this case you would release it to both packaged solution and custom development vendors.

- ◀ The Blueprint means the prospective vendors can precisely quote on the assignment as they know exactly what is needed and how 'big' the application is. You don't need to pressure the vendors on price, because they give you a fixed price quote on a fixed piece of work.
- ◀ As the project progresses, use the Blueprint to evaluate what is built against what is required. If the gap analysis is large, you can easily enforce compliance, because you told them exactly what you wanted. There should be no room for interpretation or misunderstanding.

This may seem quite a bit of initial work, but the fundamental causes of IT failure are poor requirements and an unusable interface. You don't see buildings go up, or cars getting built without a blueprint.

If you don't get the requirements right up front, your project ends up experiencing high levels of iteration and multiple releases to get it right. The ongoing cost to fix problems is easily between 10 and 100 times more expensive than the initial cost, in terms of direct redevelopment costs and lost opportunity. It is always cheaper to invest properly up front, rather than pay for it many times over at the other end.

The initial activities that should occur up front usually account for a small proportion of the estimated project cost: typically around 10 to 15%. But the net effect of getting this part right, first time, means that iteration is drastically reduced, as are post launch change requests. This approach can halve the total cost of ownership, with everything required, available in the first release.

Unfortunately, many project managers fall into the trap of cutting development costs in an effort to curb the total costs of IT. Activities such as requirements and user interface are often, mistakenly, considered optional and unnecessary. But if you don't know exactly what you're building, then you'll never get there. What happens after launch is considered to be someone else's budget / problem, or it comes out of costs already incurred. Yet all of the costs contribute to the total cost of ownership, and affect the final return on investment of the application. If all you do is break even, why did you bother disrupting your business or change your business processes to suit the application's way of doing things. You might as well have left things alone and spent the money elsewhere, or not at all.

New IT applications provide an ideal opportunity to instigate change and new business processes. Therefore, make sure that they are the ones customers value and they create a positive benefit to the company.

1.4.2. Evaluating tender responses for the right vendor for your business, staff and customers

During tender evaluation, there are steps you can take to pick the vendor that can actually deliver on your business and user requirements.

Consider doing the following during your next tender evaluation:

- ◀ Engaging specialist usability and user interface designers to evaluate the application's ability to provide a usable, high performance interface,
- ◀ Engage business specialists to evaluate the applications' way of doing things for compliance with best practice in running businesses, that suit your business, rather than making you change your business to suit.

If the best result you can achieve is to reject all tender responses, then you've achieved the very desirable result of avoiding technology overruling your business. More importantly, you have achieved this for cents in the dollar, compared with embarking on an ill advised 'adventure' with an inappropriate piece of software, with an ill defined set of requirements. With the just described alternative approach, however, you are positioned to take alternative courses of action, such as a well defined custom development.

You have also avoided being trapped in a vortex that sees you having to do so much customisation to make the system work the way you need, that you are locked into continual upgrades with the base application; a process that can result in becoming unsupportable by the vendor. At this point, you throw it out and start again.

Alternatively, if there is a good level of fit, you can choose a vendor knowing that they can actually deliver an interface that people will enjoy using to effectively do their work and achieve business outcomes for both you and your customers.

1.4.3. Driving development towards success

If you've already chosen your vendor, then you have an opportunity to determine user requirements and user interface (UI) designs before a single line of code is written.

If you're into development, use requirements and UI design work can occur in parallel, informing the system architecture to ensure that the necessary interface can be implemented easily.

If you've finished development and are in UAT, then usability testing can identify any risks you may have prior to launch. The last thing you want to do is launch a product that people flatly reject. It's hard to recover from user rejection and re-launch a refreshed application – people are easily jaded and don't often give you a second chance.

1.5. In closing...

Vendor and integrator engagement is a complex and potentially very costly process; the key to success is to be found in an ability to understand and specify what you want / need, completely and unambiguously, up front. If you don't do this, then you can't expect anything different from what you'll get - an application that arrives over budget, late, missing critical requirements and unable to quickly deliver a return in investment.

If you want to get your application right the first time, then you need to do something fundamentally different in how you engage vendors.

2. About the Author

Craig is the founder and Managing Director of The Performance Technologies Group (PTG Global), with over 15 years in user experience, user interface design and change management.

Craig runs the R&D function at PTG, having produced a number of world firsts including XPDesign – the first systematic methodology for user interface design and Certified Usable – the first guarantee for usability and user experience.

Craig has been the primary architect behind many of Australia's most popular websites including CBA, Virgin Blue and ASIC and works on cutting edge technologies such as touch, medical and special-purpose applications.

Craig holds a Masters qualification in organisational psychology, is a member of the APS and the APS College of Organisational Psychologists and is a Registered Psychologist in NSW. He is also an Associate of the University of NSW and Macquarie University.



Contact Craig on:

Email: craige@ptg-global.com
Phone: +61 (0)2 9251 4200
Mobile: +61 (0) 416 266 216
Address: Level 16, 207 Kent St, Sydney, NSW, 2000, Australia