

1. Buy not build - a fallacy in IT procurement

1.1. Summary

In my other article, Debunking the myths of IT development, Myth 3 was 'Off the shelf solutions will work out of the box'. This article is an extension of that myth and specifically deals with a common mantra in business and government of 'Buy not build'.

Despite the claims of how well applications work out of the box, specifically complex domain applications such as commercial lending software for a bank or general ERP and CRM applications, our experience has shown that buying an out-of-the-box solution is not delivering on the promise.

And what exactly is the promise? That the solution will work out of the box and deliver some high percent of your needs. However, we have not found a single implementation of such a solution that has not required customisation that is at least one, if not two orders of magnitude greater than the cost of the licensing? It's reasonable to expect some customisation, but if the licensing costs \$5 million and your SI costs are 10 or 20 times that, is it still reasonable?

The policy of buy not build is a direct result of the failure of IT to deliver solutions quickly and efficiently that work the right way first time. Businesses figure that if they buy a solution that is mostly right and ready to go, and they need to make few tweaks here and there, then their chances of success are that much better.

In this article we cover why this mantra is currently in place and why it is indeed a fallacy. The question is not 'should we buy or build?'. In fact, it should be 'how do we precisely describe our requirements so we can pick the right solution, regardless of whether we need to buy or build it?'

1.2. Why business prefers to buy not build - a brief history of application development

When technology became mainstream in the late 80s and 90s, the only choice was to build applications from scratch. But projects were characterised much as they are today by being very complex, taking a long time to deliver and generally being very hard to use.

Requirements, as they are today, are poorly documented and understood, causing the failure rate to be so high.

Fast forward through the 90s to today, many business prefer, instead, to buy an existing system. Why? They believe:

- ◀ Years of application development by the vendor has ironed out all the 'kinks'
- ◀ There is a belief that the vendors have done much of the hard work and built a stable and flexible system they can use as a basis for their needs.
- ◀ There's much less risk going with a system that 'works'

They are jaded by custom developed solutions that only served a purpose for short time, never really worked properly, or couldn't scale or expand to suit changing needs. It's actually all very reasonable until you begin to look at why custom developed solutions didn't achieve the required success and then take a look at why buying an existing solution doesn't solve the problem either.

1.3. What are you really buying when you buy 'off the shelf'?

There are a lot of vendors out there in the ERP and CRM space, as well as in the domain specific application space, all claiming that their system is best, easiest to use, most flexible etc.

The buying process begins with the business documenting its requirements, usually at a fairly high level, and then goes to market to select a vendor and SI partner. We've covered the significant issues with this approach in our other paper 'A new approach to vendor engagement'.

The business has the following perceptions that influence their buying decision:

- ◀ They can embed their needs into the way the application works
- ◀ The way the application works really does reflect best practice in the domain
- ◀ That it's really easy to use and will reduce staff workload
- ◀ That it will increase compliance and standardisation of work delivery
- ◀ That it will deliver greater control over the enterprise.

What are you really buying? In our experience, we've found that many software vendors with products originally developed from a piece of custom developed software for a client a long time ago. Over time, they've enhanced their product in response to customer needs amalgamating them back into the core product.

What are the implications of this? What questions should you ask of the vendor's solution to judge its suitability for your business? Here are some...

What is the quality of the base application?

If the base application was developed from custom software developed long ago, and if you agree that custom developed software generally exceeds the original budget, takes significantly longer than expected and doesn't work very well, then using it as a base for a generic product is very risky.

Does the application reflect best practice in the domain?

Further, if the original product was built for a customer's specific needs at that time, it must reflect their idiosyncrasies in running their business. Do you know how well they ran their business and whether it makes a suitable basis for generic software?

The application was undoubtedly left up to the developers to build and design the interface. We can be very confident they weren't experts in how to run a business, write a commercial loan, or process a claim. The application invariably reflects the developer's view of the world, not best practice.

Is it truly easy to use?

We've covered in our many papers the impact of usability of application success. The chances are that the application was developed by people who did not understand what truly makes a user interface easy to use.

How well has it been architected to allow flexibility and scalability?

Most of the boxed solutions we've worked on have been very poorly architected. What do we mean by this? Our approach is to get the user interface right first, then use it to direct how to improve the application. It's the same approach used in building and construction where the architect designs the building first and works with the concrete and steel guys to build it.

We generally find the following:

- ◀ The database is already in place and can't be changed without considerable effort
- ◀ The user interface and application logic are so tightly coupled that changes cannot be made. We've even found situations where a field cannot be hidden when not used. Special instructions had to be given to the users to 'ignore that field'
- ◀ The embedded processes cannot be changed without significant effort

1.4. 'Buy not build' really means 'buy and customise lots'

We have never seen a single installation of any boxed solution that went in without customisation. They all required modification to suit specific business processes, any in some cases, a base system was used for a totally inappropriate purpose, all because of how procurement was handled. We have covered an improved vendor engaging process in this article: 'A new approach to vendor engagement'.

At the end of the day, when you buy a boxed solution, you are inevitably going to customise it, if not, you need to change your business to suit the way the application works. Before you do that, ask yourself: 'what does Vendor X really know about running my business?'. If you're comfortable that you would invite the vendor to run your next business strategy meeting and report to the Board on how to achieve next year's 10% increase on sales, then you're probably sitting on a winner.

So as soon as you customise the solution, you have gone straight into build. You are custom building your solution, but the only difference is that some of it is built for you, and what is already built is likely to make your life difficult because of the inflexibility of the solution.

It is a very real possibility that by the time you have licences the base software and then pay the SI's time, you could have custom built it.

Either way, we've found when reviewing boxed solutions that they technically work in so far as embedding the business rules, but:

- ◀ As the business changes, they require further customisation,
- ◀ People do not find them easy to use
- ◀ Training mainly comprises how to use the software, not how to do their job better
- ◀ The solution is delivered years after the promised date and well in excess of the budget

1.5. It's not about 'Buy not Build'

At the end of the day, the decision to buy not build is purely a reaction to the fact that custom built solutions do not deliver. Since most organisations end up in the situation of having to make compromises, or open their wallet, after engaging a solution, it should be clear that it 'buy not build' is not the issue nor the decision.

The decision should be: 'How do we precisely define what we want to procure the right solution?'

I refer you to our other papers on how to define requirements. Only once you know exactly what you want, and you've defined your IT Blueprint to satisfy business, staff and customer needs should you look at the market to find a solution.

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