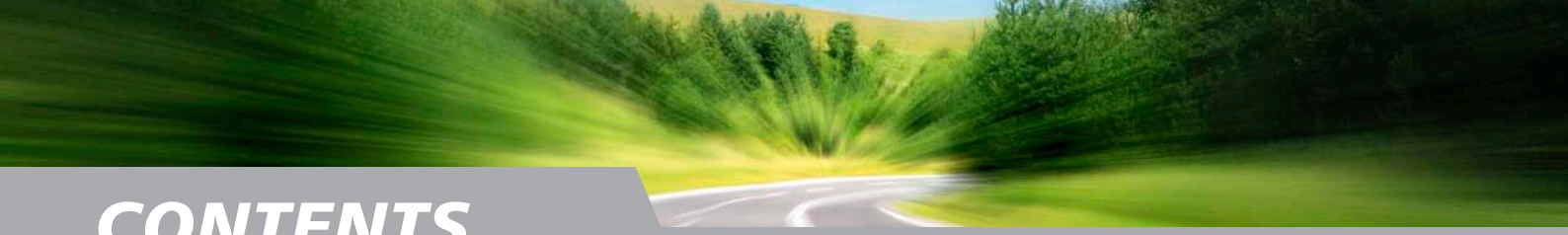
BUY VS. BUILD

The case for buying an 'off-the-shelf' AARTO infringement management software solution

Hewitt Roberts, R2X Incorporated, February 2012



CONTENTS

The case for 'off-the-shelf' software solutions: Cost

Today, most larger companies have IT departments filled with competent, capable and tech savvy IT specialists to provide for the ongoing and seemingly endless technology needs of a business.

If your business is confronted with the need for a specialist or niche software application – such as an AARTO infringement management software solution – why then wouldn't you use the existing resources available to you to meet that need? After all, they know IT, they know your business and they can be educated to understand the business need at hand.

Over the last few decades, this question has been asked countless times and in countless boardrooms around the globe. In some exceptional cases it is the preferred route but in the vast majority of cases it is not and here is why.

Often the case for building rather than buying is substantiated on the fact that your IT department is a 'sunk cost' and thus building in-house is a cost effective use of existing resources. Arguably, your IT department is a sunk cost but if that team of qualified, experienced and typically well paid employees are sitting around waiting for something to do or build, then your business has much bigger problems than AARTO, IT or software.

Development and maintenance costs

When considering the cost of building from scratch or buying a commercially available off-the-shelf (COTS) software package, in-house project proponents typically either underestimate the time really required to build a complex software application, fail to include the full lifecycle costs of a software project or both.

When estimating the time and cost of building from scratch, there are a number of very common pitfalls that rapidly increase the time and cost of a project. Each of these pitfalls exaggerates and compounds the next and exponentially increases the severity of time and cost overruns. Typical pitfalls include[i]:

- Unrealistic deadlines from management;
- Vague definition of project deliverables;
- Inadequate time for software design;
- Little or no beta testing;
- Internal politics;
- Poor time and cost estimating techniques;
- Lack of a quality assurance process;
- Incorrect (or no) calculation of actual development costs;
- Lack of proper project management;
- Insufficient resources for ongoing maintenance and support;
- Lack of up-to-date design capabilities;
- Documentation is overlooked or avoided;
- "Scope creep" and associated "cost creep"; and,
- Developers required to perform current job and develop new applications at the same time.

"Everybody knows that the more standardized you are and the more you buy off-the-shelf, the more cost effective it will be for both implementation and ongoing maintenance".

Mark Lutchen, former global CIO of PricewaterhouseCoopers

Another crucial time and cost factor is that when senior management and IT executives evaluate build versus buy, they rarely include the full lifecycle cost of the project. Often the thinking is that when the software is built and delivered to the business for use the project is done – right? Wrong – very wrong as this all too common approach does not account for the time required to fix bugs; develop features missed or misunderstood in the requirements analysis or development phases; add enhancements to meet business, market or legislative changes; and maintain compatibility with other internal systems.

In fact, and according to Mark Lutchen, former global CIO of PricewaterhouseCoopers, "when evaluating whether to buy or build, it's critical to thoroughly understand total costs during the software lifecycle - typically 7 or 8 years. This step is important, because 70% of software costs occur after implementation and a rigorous lifecycle analysis that realistically estimates ongoing maintenance by in-house developers often tips the balance in favor of buying… Everybody knows that the more standardized you are and the more you buy off-the-shelf, the more cost effective it will be for both implementation and ongoing maintenance[ii]".

Purchase, deployment and usage costs

A second and significant cost factor is the increasing availability of software as a service (SaaS) usage and pricing models. Unlike a traditional software purchase where the user purchases an up front and perpetual licence to use and then the requisite hardware to run the application, the SaaS purchase model is an on demand model where users subscribe to use a software application on-line and pay monthly or annually based on usage and as the need arises. With SaaS there are no hardware costs and you pay only for what is used.

Consequently, as the initial setup, maintenance and ongoing costs for SaaS are much lower than the equivalent for a perpetual licence software purchase, the total cost for a CAPEX software package normally far exceeds the total cost of a SaaS solution.

Opportunity costs

A third and significant cost factor often overlooked is the opportunity cost of your IT team's involvement in the development of a software product that is otherwise available as an off-the-shelf solution. If your IT department is busy building – in this case an AARTO infringement management software solution - they are obviously not able to dedicate time to other projects and needs of your business.

Clearly, if your IT team is amidst a development project, they would not be able to undertake other - perhaps more attractive – opportunities should they present themselves and of course it is virtually impossible to measure opportunity costs before the opportunities present themselves.

The rule of thumb in this case, and as again identified by PricewaterhouseCoopers' Lutchen, is to "buy applications to the maximum extent possible to cut costs -- freeing up resources for whatever really needs to be built in-house".

> "Buy applications to the maximum extent possible to cut costs -- freeing up resources for whatever really needs to be built in-house".
>
> Mark Lutchen, former global CIO of PricewaterhouseCoopers

The case for 'off-the-shelf' software solutions: Time-to-market

If your business has ever been involved in a software development project, one thing will be certain. The project cost far more than budgeted and it took at least twice as long to complete. The many reasons for this are simple and in most cases unavoidable but the key factor is that unless your development team regularly creates complex applications (and of the scale that would be required for an effective AARTO infringement management solution), they will grossly underestimate the time required to build the application.

In this situation, your best-case scenario would likely be a lengthy delay in the deployment and a very likely outcome would be complete failure. Furthermore, when developing software, research has shown (see Table 1 below) that there is an inverse correlation between the size of a project and the probability that it will actually ever be completed[i].

Project Size ($USD)	People	Time (months)	Probability of success
< $750k	6	6	55%
$750k-$1.5M	12	9	33%
$1.5M-$3M	25	12	25%
$3M-$6M	40	18	15%
$6M-$10M	250+	24+	8%
$10M+	500+	36+	0%

Table 1: In-house develop success rates

If time-to-market is not an issue then perhaps in-house development may be a viable option for your business. However, in most cases – as is certainly the case with AARTO that could very well become an immediate business problem if it is not already – time-to-market is a very real concern.

> If you build in-house, the design, development and deployment will take years not months but if you buy a readily available COTS solution, the application can be acquired and deployed almost immediately.

Ultimately if you build in-house, the design, development and deployment will take years not months but if you buy a readily available COTS solution, the application can be acquired and deployed almost immediately.

The case for 'off-the-shelf' software solutions: Product quality

Another key factor in the buy versus build debate is that of product quality. Yes your business might have an excellent IT team and may even have experience developing complex software solutions. But, just as running a successful business involves a lot more than a great product or service, developing a superior and niche-specific software product requires a lot more than great developers.

In the case of an AARTO infringement management software solution, success can and will only be achieved if you also have superior domain expertise about the problem being solved, an extensive understanding of the AARTO Act and it numerous and varied requirements, and an on-going commitment to the technical, legal, market and political changes in that domain.

Quite simply, unless your business's core competencies are building and maintaining complex software applications and AARTO infringement management, it is highly improbable that your business will be able to build a better application than that which could be provided by a specialist provider.

The case for 'off-the-shelf' software solutions: Ongoing product improvement

For arguments sake, let's assume that your business did have all of the resources and competencies required to design, develop and maintain the software product in question. Let us further assume that there are or were no opportunity costs that were expected to impact your business and the in-house development project was given the green light - all systems go. Then your team developed to specification (unlikely), budget (remote at best) and on time (exceedingly improbable) and the deployment was greeted with requisite applause, congratulations and champagne. Now what?

Your team will be ready for the next task at hand; they will devote at least as much time again maintaining their product over the next 7 or 8 years of its lifecycle but the business users can now happily manage AARTO infringements (and drivers, vehicles, demerits, penalties, etc.) effectively.

Over time however, business needs change and so too do the needs of even the most successful software products. In order to improve a software product over time it is the users of the product that will always provide the most considered and meaningful product improvement ideas. It stands to reason then that the larger the user base contributing product improvement ideas to a product, the better the product will become over time.

Consequently, the source of product improvement ideas for an in-house product will be limited to the fleet and/or infringement manager(s) in your business whereas the source of product improvement ideas for a COTS product will be all users in the entire the COTS product client base. In this situation, the COTS user base will undoubtedly be exponentially larger than that of the in-house system and the COTS product improvement ideas will come from a vast and specialist group replete with extensive fleet management, infringement management and AARTO experience[iii].

Understandably, the COTS product improvements will arguably be of greater value, will be developed far more quickly and the cost for such ongoing development is incorporated free-of-charge in the ongoing and marginal SaaS subscription rate.

The case for 'off-the-shelf' software solutions: Improved business processes

When attempting to improve business process, improve efficiency or reduce cost, executive management will often employ the services of business process improvement consultants. Invariably, the consultant will first ask 'why is it done like this?' The answer – in most cases – will be 'because we have always done it that way'?

Notwithstanding the fact that 'that way' may be the best way (although that is unlikely in the eyes of the executives who have just employed the consultant), if you design and build an in-house software product to improve a business process, the product design will, by definition, inherit the inefficiencies of the business process in question.

A COTS product however – and certainly in the case of the new AARTO infringement management needs - is designed from scratch and based on legislative requirements and influenced by wide range of best practice infringement management from a wide a varied pool of sources and inputs.

Consequently, a COTS product – especially when meeting the needs of a complex, new or multifaceted business problem as is the case with AARTO - will unlike an in-house build most likely provide a best practice approach to solving problems and will likely also provide very real opportunities for business process improvement[iv].

Conclusion

Without a doubt, choosing to build versus buy a commercially available off-the-shelf product is a complex decision. Making the wrong decision however, could have a huge impact on your company and your career. The buy versus build debate has been raging for decades and if you are confronted with this decision, at least be well informed.

- Building in-house will almost certainly be far more expense than a market-leading off-the-shelf solution.
- Where an off-the-shelf solution is offered on a subscription basis (SaaS), total cost of ownership will undoubtedly be cheaper than an in-house solution.
- With a SaaS solution, the up-front investment is exceedingly minimal and if you don't like the product after initial use, you can simply stop using – and paying – for it.
- An in-house solution will most definitely take far longer to develop than expected and require far more ongoing resources to maintain than anticipated.
- The opportunity costs of developing in-house can be enormous and debilitating to your company's competiveness.
- Niche-specific off-the-shelf products are designed and developed by companies that specialise in software development and the domain in question and an in-house product will rarely provide a product of better quality.
- Product improvement ideas for off-the-shelf products are borne of a much larger user base of domain experts and will always improve at a rate far faster (and more cost effectively) than an in-house solution.
- COTS products are typically developed to meet very specific business needs and often provide a means to employ best practice and provide for very real business process improvement opportunities.

ENDNOTES:

[i] http://www.insidus.com/BuyorBuild.pdf

[ii] http://www.infoworld.com/d/applications/build-or-buy-it-applications-676

[iii] http://www.baselinemag.com/c/a/Application-Development/Buy-vs-Build-Software-Applications-The-Eternal-Dilemma/

[iv] http://www.techrepublic.com/article/consider-these-points-when-making-the-build-vs-buy-decision/1045594