Advantages and Disadvantages of off-the-shelf software

A white paper by DCSL Software
## Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Pages</th>
</tr>
</thead>
<tbody>
<tr>
<td>INTRODUCTION:</td>
<td>1</td>
</tr>
<tr>
<td>SECTION ONE:</td>
<td>2 - 3</td>
</tr>
<tr>
<td>The fundamental differences between bespoke and off-the-shelf software</td>
<td>2 - 3</td>
</tr>
<tr>
<td>SECTION TWO:</td>
<td>4 - 5</td>
</tr>
<tr>
<td>The three biggest myths of bespoke software</td>
<td>4 - 5</td>
</tr>
<tr>
<td>SECTION THREE:</td>
<td>6 - 7</td>
</tr>
<tr>
<td>When is off-the-shelf software a better fit?</td>
<td>6 - 7</td>
</tr>
<tr>
<td>SECTION FOUR:</td>
<td>8 - 9</td>
</tr>
<tr>
<td>Avoiding the common pitfalls of commissioning bespoke software</td>
<td>8 - 9</td>
</tr>
<tr>
<td>SECTION FIVE:</td>
<td>10 - 11</td>
</tr>
<tr>
<td>Finding the right partnership – and what to expect from it</td>
<td>10 - 11</td>
</tr>
</tbody>
</table>

---

## Introduction

In today’s competitive business world, it can be difficult to decide whether to purchase off-the-shelf software or order a custom-built solution. As technology becomes more advanced, there is an increasing amount of ready-made applications available to subscribe to or purchase, but often the convenience will come with a compromise on functionality or support.

In this paper we will outline some of the most common decision factors that come into play when choosing business applications, to help you navigate the sometimes complex landscape of business software.
The fundamental differences between bespoke and off-the-shelf software

Does software start with the user – or with the technology?

It sometimes seems like we live in a world ruled by software. Our everyday lives and businesses are constantly driven to become ‘smarter’ with the help of technology. But does this mean that we, as a society, are adapting to the software revolution – or is the software adapting to us?

In reality, both of these things are happening. Take the average Facebook user, for example. With every new functionality or change, the individual adapts their usage – whether they like the change or not. The platform alone dictates how it is being used.

Your unique software needs

A business application needs to support specific processes that may or may not be the same as for other organisations. Having a system that is pliable to your own particular requirements is often much more cost-effective than trying to change your processes while also introducing new software. However, not every business process will require a specialist application. For basic, non-critical functions, you may find that an off-the-shelf solution works perfectly well.

The differences between bespoke and off-the-shelf software

In some scenarios, choosing what type of software to invest in can be challenging. To help guide the decision-making process, let’s take a look at the differences between bespoke and off-the-shelf options.

Area 1: Fitness for purpose

Generic applications can be extremely sophisticated, as they are developed for a wide audience and have had considerable investment going into their construction. They are typically designed sometimes means the software comes with lots of features that many users will never need or want. It’s extremely rare that a customer is able to extract the full potential of an out-of-the-box application. Some software packages are believed to have 90% of their functionality unused, as they are so feature-rich. This can in itself be confusing to the end user, if they have to navigate around ‘no-go areas’ in the user interface.

While the average off-the-shelf solution is packed full of features, it still often fails to be specific enough to support all the unique processes of an organisation. In some business areas, the average off-the-shelf solution only meets 80% of the customer’s needs. This means that you’re always compromising on the added value of the remaining 20% - which could make a big difference in productivity.

Area 2: Training

When buying off-the-shelf, many organisations fail to factor in the cost of the training needed to use the software. While many solution providers do offer free training after purchase, this still requires staff to be pulled out of their day jobs, sometimes over several sessions, in order to be trained on the system. In some cases, the vendor only provides pre-recorded video tutorials and online FAQs to guide the user – which may not be specific enough to help them understand how they can achieve the result they need.

The difference for a customised application is that the software features are already designed to match the existing business processes. This means that, in theory, minimal training will be required. Should the product prove to be too difficult to navigate, this would typically be discovered in the testing phase and rectified before rolling out the software to the entire business.

Area 3: Support

The centralised support services of an off-the-shelf application may be remote and slow to respond, and the time zones might not suit your business. If the support team is located abroad. On the other hand, you may have access to an online user-based community to complement the baseline support service.

Area 4: Development influence

A popular off-the-shelf solution will have a wide audience of users who can request functionality and integrations that the vendor can choose to include in future versions or add-ons.

For some businesses, it can be tempting to use a ready-made solution that’s ‘almost perfect’, as it means they won’t have to spend time directing a custom-built software project. But for the individual user it can be incredibly frustrating to not have access to a simple function that could make a huge difference to their daily work. While they can contact the vendor and suggest adding a function, the vendor is unlikely to meet their request unless a large portion of the user base was to request the same function.

Area 5: Price

Price is calculated very differently for the two software models. The typical pricing for off-the-shelf applications is based on the number of user licences, which can quickly add up to large amounts. An implementation of a typical, complex, business-wide system will also incur substantial consultancy costs – especially if it requires integrations or customisations.

With bespoke software, there is an up-front development cost but no licensing, as the customer owns the software outright. However, there will still be some ongoing costs in terms of hosting the application in the cloud as well as having a suitable support agreement in place.

Area 6: Software updates and upgrades

One of the main benefits of off-the-shelf software is that it is usually updated on a regular basis, to improve or add features and take advantage of new technological developments. While this often benefits the end user, it can just as easily cause problems if a key feature is replaced or discontinued.

As a customer, you have no control over the roadmap of off-the-shelf software. A sudden, unexpected functional change can disrupt operations and impact business performance.

Area 7: Ownership

One of the key benefits of off-the-shelf software is that you don’t own any responsibility for it. It will always be developed, maintained and supported by the solution vendor, as a fully managed and – hopefully – proactively improved solution.

Interestingly, however, this same aspect is also one of its key disadvantages. You never own the right to modify or streamline off-the-shelf software, you can’t copy it, and you can’t resell it.

Area 8: Trials

When searching for an off-the-shelf solution, the chances are you’ll find a free trial version to try before you buy. However, it isn’t like trying cheeses at the deli. Somebody will have to invest time and effort into making sure the product is put through its paces, and determine if it actually does what you want. Also, it’s worth remembering that trials are by their very nature designed to make the product look as compelling as possible, in order to trigger a desire to purchase. In a worst case scenario, the solution you choose turns out to be unable to support the business in the long term.
The three biggest myths of bespoke software

Uncover the truth about custom software development

As the software landscape has become busier with off-the-shelf offerings, organisations have often become more likely to look for their ideal solution among the range of existing applications in the market.

Many companies don’t even consider having their own system custom made.

In fact, many small and medium size businesses don’t believe they would be able to have applications fully tailored for them. This reluctance is sometimes based on their belief in these three lingering myths of bespoke software.

Myth One: Bespoke software is expensive

The pricing model for developing a bespoke software system is obviously very different to that of a mass-market application. When designing an off-the-shelf product for a large customer base, the vendor can invest heavily in the technology knowing that the costs will be recovered as the user base grows. Bespoke software, on the other hand, is developed for a single client who will have to fund the entire project.

However, this does not automatically mean that a bespoke system is more expensive for the customer – especially when you consider the subscription or licence costs for off-the-shelf systems over time. The below example shows how quickly the licence costs can escalate for a medium size business.

Cost example: Typical off-the-shelf ERP system

<table>
<thead>
<tr>
<th></th>
<th>1 month</th>
<th>1 year</th>
<th>3 years</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cost per user</td>
<td>£50</td>
<td>£600</td>
<td>£1800</td>
</tr>
<tr>
<td>Cost per 100 employees</td>
<td>£5000</td>
<td>£60,000</td>
<td>£180,000</td>
</tr>
</tbody>
</table>

As a comparison, it is possible to get a basic bespoke business management system from £50,000, where the software is fully owned by the customer.

Many developers also offer the opportunity to finance the purchase over a reasonable amount of time.

When costing up the options, it is also worth considering how many of the system features will actually be used. If an ERP system comes with a large number of unnecessary functions, you will not be getting as much value from it compared to a bespoke system where you choose exactly what you need.

Myth Two: Bespoke software is lower quality

It’s easy to assume that a brand-name software application will perform better than a bespoke one, simply because it has a larger number of users that provide feedback for improvements. Surely, a brand name solution will have quality assurance processes that far outweigh the ability of a smaller bespoke software developer? Well – not necessarily.

In many cases, software companies will only invest in improvements that make financial sense for them as a business in the long term, whereas a bespoke development project will allow you as a customer to define the quality criteria.

Some of the most common software quality complaints for mass-market software are around missing features or system limitations, but these are often linked to specific needs of the customer – which may not be represented by others. These could be incredibly important to the individual user, but not matter too much to the software vendor.

Working with a bespoke development team, however, gives the customer control over features and functionality.

Myth Three: Bespoke software is not as well maintained

There is a misconception that a bespoke software project means a ‘hit-and-run’ approach, where the developer abandons all responsibility for the software once the final solution is delivered and the invoice is paid. While this scenario may possibly play out when it comes to less serious players in the software marketplace, it is not what the average customer can expect.

A customer who has done their homework and found a reputable and accredited development partner will find themselves in a long-term relationship where both parties are engaged in resolving any future issues and improving the software over time.

Meanwhile, licensing an off-the-shelf software product does not automatically guarantee the level of support and maintenance that you’d expect. We have seen countless examples of where software vendors have decided to end support agreements and stop upgrading applications, despite the software being used and relied upon by many customers. Often the customer is encouraged to swap out the solution for an alternative – which can end up being less aligned with what they want.

It’s also worth finding out where the software was built and where their support department is located. If the team is based overseas, you may have to factor in timezone challenges for maintenance-related issues.
When is off-the-shelf software a better fit?

Invest where it matters

There are many benefits to having a custom made software application, but it is not always the best investment. There are many business scenarios where existing, mass-market products do the job so well and so consistently that there is no real need to create the same functionality in a custom solution.

One example is accounting software, where market leaders like Sage, Quickbooks and Xero are dominating the industry with feature-rich and user-friendly applications. There is simply not much sense in re-designing accounting software that works well. Instead, companies typically choose to integrate a bespoke element into the existing system for any added functions they need.

Develop a use case

When your business identifies a need for new software, it’s important to take the time to develop a use case for it. Asking a set of initial questions and documenting the response from the key stakeholders involved will help you arrive at the right decision on where to invest.

Use case development process flow

1. What does the requirement look like?
2. Can the requirement be met using process change, reallocation of resource, or better utilisation of existing tools?
   - Yes → Describe how
   - No → Are the standard products available to them to fulfil the requirement?
     - Yes → List the existing products
     - No → Outline the key functions required
3. How will meeting the requirement improve business performance?
4. How will the results be quantified to measure ROI?

Defining what’s needed

By mapping out the key functions as described in this workflow will lay the groundwork for creating a software specification further down the line, once you start working with a developer. But at this stage, it will simply serve as an overview of the functions you need.

In some cases, you may not know exactly what it is you want to achieve, or even understand the potential possibilities available to you. However, a skilled development team will be able to help you analyse your problems and offer clear definitions with realistic timescales to deliver.

An important part of building a use case is understanding the financial benefits of having access to the new software. If you can identify how these functions will have a positive impact on the productivity or profitability of your business, you will already have a good view of what the application will be worth to your organisation. This in turn will help you make decisions on how to spend your software budget.

Ideal business areas for bespoke software applications

The more unique a function or process is to your business, the harder it will be to find an existing solution that does exactly what you need it to do. This is where bespoke development can really make a difference. There are some examples of functions that are notoriously difficult to match up to existing off-the-shelf applications, as they will often be very diverse and require a highly tailored approach.

Online self-service

If you have a service offering that has a unique or innovative edge, chances are that you will want to do things differently to everyone else. This is where you can bring your user experience vision to life with the help of bespoke software, web and mobile app development. The internet is a constantly growing hive of innovation, and you can choose to bring together all the best elements of the latest technology to your service pages and apps.

Operational systems

Nobody knows your processes better than you do. Chances are that you have developed an internal operational landscape that is so perfectly tailored to your teams and service delivery that there won’t be a single system out there that supports the way you work. Rather than patching together a set of different systems to deliver one outcome, it is often a good investment to create your own software.

Internet of Things (IoT)

We’re seeing a dramatic increase in the number of IoT innovations in British industry. Hardware and physical equipment is now increasingly being designed with connectivity in mind, and needs specialist software and integration applications. Many companies are now able to create a powerful competitive edge with not just great products, but with smart technology that brings them to life for the user.
Avoiding the common pitfalls of commissioning bespoke software

The checklist for successful software development and rollout

Once your business has decided to move forward with developing bespoke software, it’s likely been identified as a strategically important project – and should be approached with great care and attention to detail from both customer and developer.

Write a brief

The first step is to take the functional requirements you have identified and turn them into a clear brief. This document will form the basis of your agreement with the developer, and will outline the high-level scope of the project. This will make sure things like pricing, timescales and resource planning can be accurately estimated.

Shortlist suitable development partners

As well as taking time to analyse the challenges the software needs to address, it’s important to shortlist suitable software developers early in the process. Find partners that have the relevant credentials, certifications, online reviews or awards. The developer you choose to work with should be able to show that they are working to good industry standards as well as offer successful case studies and client references.

Identify stakeholders

It’s important to clarify who the key stakeholders are for the software development project. Consider who will be signing off on various decisions throughout the project such as functionality, cost, design, and implementation. There should be a clear communication schedule where team members are kept up to date with progress, to ensure everyone is engaged.

Ensure continuous planning

Planning never truly stops. All the way throughout the project, you should keep planning the next phase – in terms of development as well as implementation – with constant feedback and improvements. The developers will also be able to provide suggestions as they listen and understand your requirements.

Communicate

Communication is a fundamental element of any software development project. As a client, you need to know what goes on between meetings – and the developers need you to communicate clear direction.

There should be progress updates and feedback loops built into the development model, to make sure there is transparency throughout. Often timescales are critical, which means that the software team must keep you informed of any unexpected issues or delays – while escalating the problems effectively for discussion and resolution.

Get involved

When you engage with a software developer, you should expect to be included in all strategic project planning meetings. It’s important for you as a customer to have visibility of the team working on your system, and be kept up to date with their progress.

The team will inevitably need your guidance and clarification throughout the development, to ensure you’re happy with how the brief is being interpreted. Your involvement is key to the success of the project.

Test

You should be able to see prototypes early on in the project, to encourage experimentation, testing and discussion. Once you are able to test and evaluate a tangible software model, you will be able to suggest features or modifications as the project gradually evolves. Testing obviously not only helps with the development of new features, but also helps ensure the end product is reliable and free from bugs and faults. Most professional software companies implement an Agile methodology, which encourages change and involvement by all stakeholders. It’s crucial that you test the product every step of the way, not just at the end. If you do that, it’s likely that the software will behave in unexpected ways and cause disappointment – not to mention additional costs.

Deploy and on-board

It’s crucial to plan for what happens once the new software is deployed. You need structure around managing user adoption, training, future integrations, and support. You should appoint a set of internal “champions” of the new software, who can help others get the most out of the solution. Make sure users get enough training to understand and learn the functionality of the new system, so as to get the best return on investment as quickly as possible.
Finding the right partnership – and what to expect from it

The things to look for in a software development partner

Regardless of the reputation of the developer you consider using for your software projects, there are certain things you should not compromise on. As a buyer, it is easy to get enticed by promises of great product at a low price, without having any tangible guarantees.

Rather than rely on handshake agreements, look for the provider to offer the following three things:

1. Create an MVP (Minimum Viable Product) to prove that your commercial and technical model will work. Once the market has been tested, you can move on to the next phase of complete feature development.
2. Leverage any work already done by taking a simple test version of your software, perhaps written quickly with limited skill or resource, and turn it into a solid, reliable product.
3. Provide finance solutions for funding the development project, where external funding or own funding is not available.
4. To pass the intellectual property over to you to ensure you fully own the software that has been developed.

Get a clear cost overview

Be clear on what you expect to pay for the software you want. If the cost is over your planned budget, make sure you understand what the added benefit would be for spending more. Always aim to have an open conversation with the developer about cost — and potential discounts for multiple applications or longer term projects. Costs for bespoke software can overrun if you don’t keep a close eye on what the stakeholders are asking for. Ensure whoever is running the project from your side is aware of the budgetary limitations.

Consider security

Depending on the sensitivity of the data handled by the software and how business-critical the application will be, you may want to take the time to consider any potential security concerns. Think about how the application will be used, what security risks there will be, and how the application can limit any vulnerability. Bear in mind that the General Data Protection Regulation (GDPR) is coming into force in 2018, which means that application and data security is now paramount to your business as the fines for data breaches are eye-watering.

Consider integration

Get a good understanding of how the new software will fit into your existing IT landscape. You should always have a clear view of how you expect any new application to impact the business and existing processes. This is particularly important if the application needs to function alongside existing software, and share data with it.

Discuss the possibilities with your developer of how additional software integrations — if any — will be added. Be clear on which APIs should be delivered with the initial product, and who will be responsible for future integrations.

Consider documentation

Coders write code, but they also need to write good documentation. This could be technical information for internal use, or it could take the form of user guides and help features. If the coders don’t document their work, the business should employ specialist writers to do it. A good software development partner will ensure that documentation is not only developed, but that it is easy to understand by the client and is kept up to date.

Consider support

It’s important to establish who owns the support process, and how you will manage software issues on an ongoing basis. In some cases, your business may want to request a support contract with the developer. This would enable you to get fast-track help if anything unexpected should happen. However, if you have your own IT support team internally, you may have a tiered approach where a certain type of issues are managed by them and only major issues are escalated to the developer.
DCSL Software Ltd. is a limited company registered in England and Wales. Registered number: 04126477.