

Deltascheme Guides to Digital Transformation Best Practice:

Custom applications in the cloud - Microsoft's gamechanger

Custom applications in the cloud - Microsoft's gamechanger

In a pivotal and historic change of stance, Microsoft has embraced open source technology and in a big way. A few years ago Microsoft recognised the advantage of using selected tools from the matured open source technology pool to build Office 365. They then bundled these open source tools together to form the 'SharePoint Framework' ('SPFx'). This now offers their customers a similar opportunity of leveraging open source tooling to extend Office 365 and build custom business applications.

SPFx presents a massive opportunity for IT departments, tasked with delivering custom business applications. Microsoft has effectively activated Office 365 as a development platform for open source technologies. By doing so, they have made development on Office 365 available to the world's largest group of developers – the open source community. These developers are now able to design and deploy the front-end of their business applications onto Office 365.

SPFx presents organisations three significant advantages. Firstly, you can speed up development time by using open source tooling. Secondly, you can incorporate DevOps best practices into their development lifecycle. Thirdly, your custom business applications can easily leverage the existing Microsoft infrastructure. Now you don't have to reinvent essentials like authentication, web hosting, document storage, and messaging, yet can still enjoy the freedom and power of bespoke code. These advantages equip you more than ever to deliver enterprise-level custom business applications to your end users.¹

Microsoft launched SPFx in 2016 and provides the convergence of three familiar IT superpowers: open source technologies, the Office 365 platform, and Microsoft's cloud (Azure). The resultant new tech-stack enables an organisation to build great custom business applications faster than ever before. The following Venn diagram illustrates this convergence, showing how SPFx enables custom business applications to benefit from the these technologies.

¹ Notably, Microsoft decided to build their Office 365 platform with Facebook's user interface (UI) framework: React js, which it open sourced in 2013. Microsoft has subsequently used Facebook's React framework to build its open source library of UI components (text fields, buttons, lists, and so on) named 'Office-UI-Fabric' for Office 365.

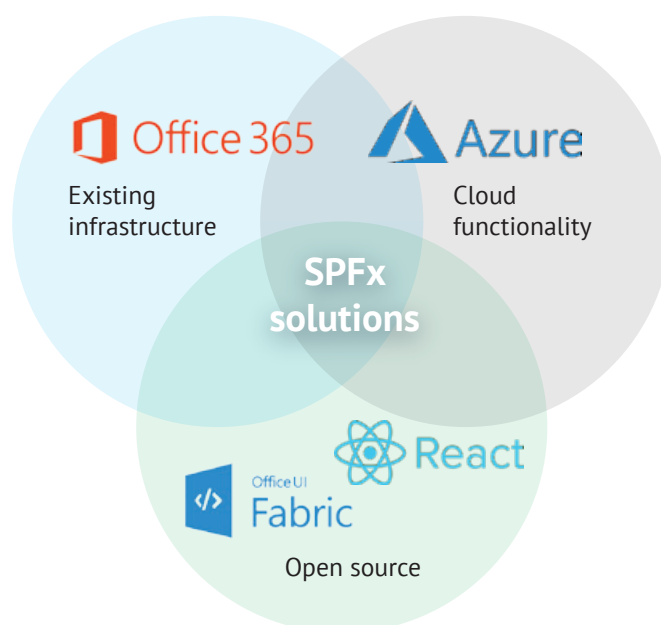


Figure 1 - Convergence of cloud, Office 365 and open-source enabled with SPFx

Why open-source is the new paradigm for application development

Remarkably, the developments to extend Office 365 has removed practically every hint of proprietary technology and methodologies. It enables Office 365 to be used as a development platform whereby a customised front-end can be deployed with a codebase that is unencumbered by proprietary technology.

Although the notion of 'open source' has been around for many years, the IT world took a special interest when Facebook and Google open sourced their software development frameworks, thereby making them freely available to all.

Tech start-ups were the prime early adopters of open source technology. As the open source effort matured, the level of trust and dependence on open source has now extended to major corporations.

These open source frameworks are designed to excel at achieving specific tasks. For example, Facebook's React is a framework for building user interfaces that makes it easier to write code to handle user interaction. Microsoft's Office UI Fabric is a library of UI components such as text fields, buttons, lists etc. that are pre-written (in HTML), pre-styled (with CSS) and have configurable behaviours (such as grouping common items together in a list), saving the developer having to do it.

Conversely, the frameworks and libraries don't limit the developer. Instead, they alleviate many of the complexities inherent in coding. They also provide developers with a starting point significantly closer to the finishing line, meaning much faster development times. It is still code but, by using these frameworks, developers are effectively 'standing on the shoulders of giants'.

Selecting the right approach to deliver a business application

Building bespoke applications on the Microsoft technology stack is, of course, not the only option to deliver a business application. Equally viable is the option to buy a pre-built solution off-the-shelf or to use the latest breed of 'low-code application platforms' (LCAPs) to build the solution. The following section reviews these three options in more detail.

Delivering a business application using a pre-built solution

Also known as a 'commercial off-the-shelf solutions' (COTS), these provide pre-built functionality that tends to meet an industry-wide need and can be configured to meet the slightly more nuanced requirements of the customer. This can be a suitable approach when the requirements of the business are both fully known and closely match the functionality of the package.

The real challenge can arise when the requirements do not closely match with the advertised functionality. The difference between the vendors' approaches is now of paramount importance. The mature vendors, such as Microsoft and Salesforce, have embraced open source frameworks and LCAPs to add value to their COTS offerings. Products from these suppliers can be safely extended. The reverse is true of the less mature vendors and customising their products can be prohibitively expensive.

Delivering a business application using a low-code application platform

These low-code tools come in various forms. Some are plug-ins that work within the eco-system of a large COTS package, while others are standalone and offer out-the-box integrations with different COTS packages.

They consist of configurable building blocks that provide a level of abstraction away from the code yet are still flexible enough to build a wide variety of custom business applications. They are primarily configured using 'point and click' and will typically include an application programming interface (API), which allows developers to extend the functionality.

They are often targeted at 'citizen developers' who bridge the gap between IT and the business. Consequently, they can deliver against some of the demand for simple to medium complexity business applications without having to hire developers.

Limitations of low-code application platforms (LCAPs)

Each LCAP has different levels of flexibility, yet they all impose some degree of constraint on the user. As the degree of complexity inherent in the custom business application increases, these constraints become more apparent.

You can overcome some constraints by injecting custom code, but this requires a degree of development know-how. This process is generally time-consuming because the code must work in the context of the underlying LCAP's code design.

Microsoft's LCAP – the Power Platform

Microsoft's low code application platform is known as the Microsoft Power Platform and consists of several components:

- PowerApps
- PowerBI
- Power Automate (previously called Flow)

Further components such as Robotic Process Automation are in the pipeline.

Microsoft also provide the Common Data Service which is built into PowerApps and includes a library of hundreds of connectors to different data sources.

The newly released PowerApps Component Framework (PCF) enables developers to develop custom controls using open-source technology (including Office-UI-Fabric) which can be used by a citizen developer to improve the user experience of the PowerApp they are making. PCF acts as a bridge between professional developers and citizen developers, enabling a citizen developer to go further and create better apps. Note however, that the citizen developer is still limited by the PowerApps environment, and the scope to which a professional developer can support them is constrained to adding new controls. Ultimately PowerApps is primarily designed to be used by citizen developers to help speed up the design and delivery of small to medium complexity business apps.

Delivering a business application using SPFx

The difference between the different approaches on the Microsoft platform is quite subtle. The functionality available in the Microsoft Power Platform (LCAP) including PowerBI, Power Automate, Common Data Service etc. are equally available to the SPFx developer, as well as a host of Azure resources such as serverless functions. The key difference is that the SPFX developer is not constrained by any proprietary software such as PowerApps.

As Microsoft has open sourced the 'Office-UI-Fabric' UI library they use for Office 365, custom business applications can now use the same 'building blocks' as Microsoft. As a result, the application looks and behaves almost identically to the native Office 365. This type of user experience (UX) is what end users are expecting from modern business applications, and Microsoft has empowered organisations to deliver it with bespoke development.

A few short years ago, we saw the arrival of Microsoft's cloud, Azure. Azure gives you the ability to almost instantly spin up SQL databases, on-demand compute power, and an array of other infrastructure resources that a typical business application would require. It has empowered developers to focus on building out the application's business logic, rather than provisioning the required infrastructure.

As well as empowering IT professionals with their Azure cloud, Microsoft have also added a myriad of tools and improvements to their Office 365 platform to enable end-users to collaborate and be more productive in their day-to-day work.

Summary of options

For an organisation already committed to Microsoft technology, the range of options to build and implement a new business application may feel a little overwhelming. At one end of the spectrum Microsoft's out-of-the-box solutions are available with minimal configuration. For the simpler, but unique business requirements, 'citizen developers' can rapidly build solutions using the Power Platform. Finally, for the complex and large-scale unique processes, custom development using the SharePoint Framework will deliver whatever is required. All options share a common user interface and draw on the rich functionality provided by Azure and other entities of the Microsoft platform. Opting to build bespoke solutions is no longer the high-risk game it used to be.

Given all the advancements made across the cloud, Office 365 and open source tooling, IT decision-makers now have more reasons than ever to embrace these technologies to deliver custom business applications.

The appendices to this document provide some further insights into this topic and a case study summarising a highly successful SPFx implementation in a very large health sector organisation.

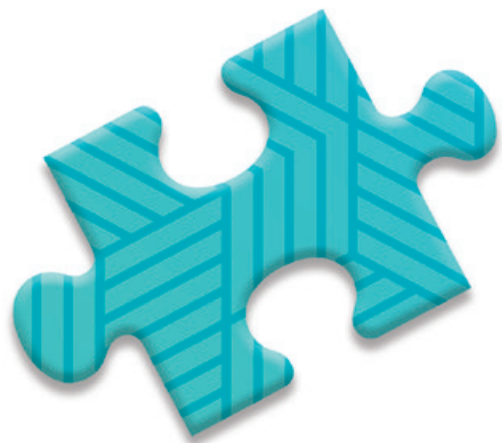
Next Steps

If you have a complex project that might benefit from SPFx, we at Deltascheme would be very pleased to offer further guidance and advice.

Deltascheme Limited

Old Estate Office
Harleyford Estate
Marlow
Bucks
SL7 2DX
United Kingdom

Main Office Tel: +44 (0)1628 401800
Email: Mike.Turnbull@Deltascheme.com



Authors:

Mike Turnbull - <https://www.linkedin.com/in/mikejamesturnbull/>
Paul Turnbull - <https://www.linkedin.com/in/paul-turnbull-1a73163/>
Stuart Robertson - <https://www.linkedin.com/in/ds-stuart-robertson/>

Appendix 1

Cost comparison between SPFx, LCAP and COTS

The following graph compares the cost of the different approaches of delivering custom business applications against the level of functionality (complexity) required by the custom business application. ²

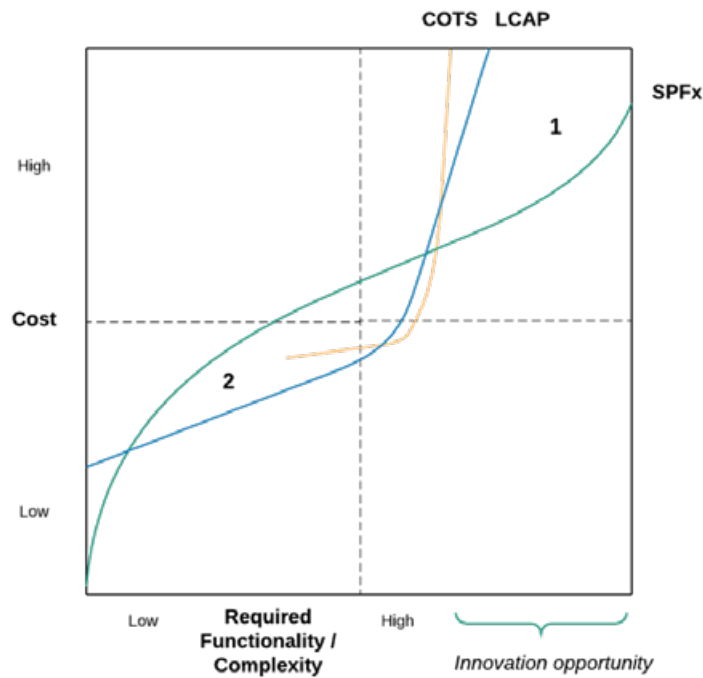


Figure 2 - Functionality vs Cost for delivering a custom business application

Figure 2 Summary:

	Initial Cost	Simple/standard functionality	Increasing complexity
SPFx	No licence cost ³	Expensive for simple apps	Mid cost for complex apps
LCAP	Low/mid licence cost	Cost-effective for simple apps	Expensive to build complex apps
COTS	High licence cost	Appropriate if functionality matches requirements	Very expensive to extend functionality

Zone 1: Favours SPFx development

Zone 2: Favours Low Code Application Platforms

² This indicative graph is derived from many years' experience delivering custom business applications using both bespoke code and LCAP's such as Nintex.

³ Assumes an Office 365 tenant is already available. Note other Microsoft licence costs may be incurred such as Azure depending upon the application that's built.

Figure 2 detailed notes:

SPFx

- a. The graph assumes an Office 365 or SharePoint online tenant has already been purchased.
- b. The relatively steep line at the start shows it is slower to build simple applications compared with LCAPs and there may be other Microsoft licences costs such as Azure.
- c. The shallow gradient further along the line shows fast development for medium to high complexity custom business applications. The gradient then becomes slightly less shallow when the complexity is very high since different architecture patterns are better for highly complex applications, such as using microservices.
- d. Zone 1 indicates where SPFx will generate a high ROI when there is no COTS package available, and the custom business application has a medium or higher level of complexity, and or, the requirements are not fully known.

LCAP

- a. The graph assumes there is a licence cost for the LCAP but until it is configured it provides no functionality out of the box.
- b. The shallow gradient indicates being able to build simple to medium level custom business applications very fast.
- c. As complexity increases development becomes increasingly slower, until the point where the gradient becomes steep, indicating where the LCAP must be customised with code.
- d. If the requirements are a medium level of complexity, then LCAP will generate a good ROI instead of using SPFx, as marked by the Zone 2.

COTS

- a. The graph assumes that there is a proprietary COTS package available with functionality that comes close to the requirements of the custom business application in question. The initial cost and functionality available out of the box determine the start position of the COTS line.
- b. Generally, there is a level of configuration to allow it to be customised indicated by the shallow gradient at the start of the line.
- c. If it can be further customised with code, this is very ineffective compared with alternative approaches, which is why the line's gradient quickly becomes very steep indicating soaring costs.
- d. If the inbuilt functionality matches the capability requirements of the custom business application, then this will generate a good ROI. If not, then customising the COTS package will lead to an increasingly poor ROI compared with alternatives.

Appendix 2

Comparison of application performance verses complexity

The next graph compares the performance levels against complexity for applications built using a LCAP and a SPFx development ⁴.

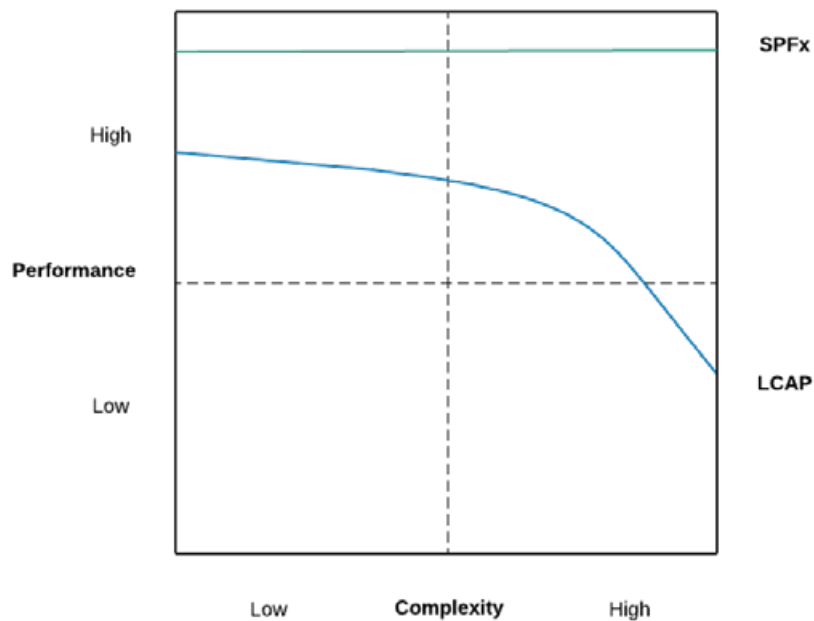


Figure 3 - Performance vs Complexity

This graph demonstrates how performance changes as the complexity of the business application increases. The key message is that the performance of LCAP based applications tend to drop off as complexity increases. Performance includes factors such as:

- Time to load user interfaces (e.g. forms)
- Ability to execute high transactional volumes
- Time to process business logic
- User interface responsiveness.

⁴ The graph is based on the observed behaviour of several Deltascheme applications built over the past few years – both using LCAP technology and bespoke development. One explanation of the degradation of performance with some LCAP solutions is the abstraction of the data layer.

Appendix 3

A model for selecting technology to build a business application

The Venn diagram below considers applications with differing levels of complexity (C1) for simple, (C2) for medium, (C3) for high complexity.

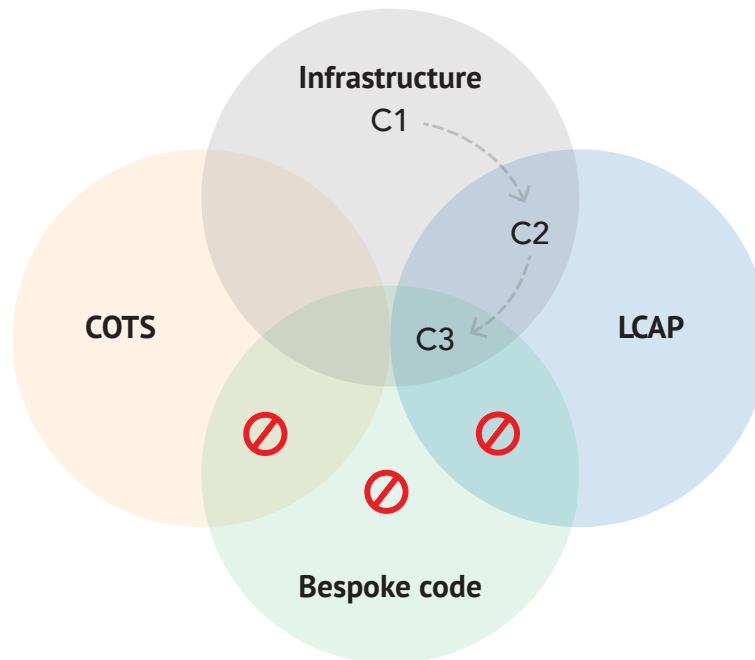


Figure 4 - Recommended choices as complexity increases

Figure 4 notes:

1. C1 marks a complexity level 1, which is a simple application which can be built with existing infrastructure, maybe an excel spreadsheet, a SharePoint list in combination with a Planner and email alerts.
2. C2 marks a medium level of complexity, which can be delivered using an LCAP in conjunction with existing Office 365 infrastructure, e.g. using PowerApps.
3. C3 marks a higher level of complexity, which can be delivered using SPFx and leverages existing infrastructure in Office 365 and Azure. The application may also use an LCAP for a discrete part of the overall application; in this way, it makes uses of all three resources. These applications can interact with a 3rd party COTS package via it's APIs.
4. The stop circles indicate areas where development is likely to be inefficient.

Appendix 4

Case study - Delivering a complex business solution using SPFx

The benefits of using SPFx are illustrated nicely by the example of a substantial public sector customer of Office 365 with a sizable workforce spread over a wide geographic area. Due to various reforms, they were planning for considerable restructuring activity and workforce churn requiring significant management effort. Coupled with this, the organisation had to adhere to strict procedures while implementing this upcoming change.

Their existing IT solution for managing this activity was a set of non-standardised MS Excel spreadsheets. They quickly recognised this was both unscalable and presented several additional problems:

1. The use of multiple spreadsheets did not support an accurate set of reports through which managers could glean the vital insight they needed to ensure a successful outcome for all the stakeholders.
2. It would be very difficult to ensure that they were following the strict change implementation procedures to mitigate the risk of being in breach and liable to a trade union claim.
3. The management resource available was insufficient; they needed to be empowered to work more efficiently to deal with the workload.

Fortunately, they prepared well in-advance and sought to migrate from an excel-based solution to a better-engineered solution. The client went about this by configuring a low code platform which allowed them to store the data more securely, design custom forms, and trigger automated processes to do things like sending notifications based on various conditions.

They had successfully configured this low code platform previously for smaller, simpler business applications, but in this case, the project ultimately failed because the chosen tech stack was too constraining and was not able to:

1. Deliver the required complex user interface
2. Handle relational data structures.
3. Quickly adapt to the high levels of changes to the business logic (since the processes were being frequently refined and needed the changes to be reflected in the UI).
4. Robustly scale to meet the high level of transaction rates generated by a large end-user base.
5. Easily integrate with third-party systems.

Trying to make the low code platform work was leading to progressively slower development and a highly compromised design which eventually led to a difficult decision to can the project. The deadline for the upcoming change programme was now looming. They were back at square one, except now with tight timescales.

The client decided to try again. Based on their frustrating experience, they decided they needed an approach that would not constrain them, one that they could deliver against a complex set of requirements, but which would also allow them to build at speed to meet their deadline (and budget!). Historically this approach was an oxymoron, but after learning about SPFx as a way of taking advantage of recent innovations across all of Office 365, open source tooling and the Azure cloud, they decided to proceed with this as their new tech stack.

This new approach delivered the project successfully. They went live with the solution on schedule and met their tight deadline. By using the existing Office 365 and Azure infrastructure, the client realised wider benefits unobtainable from the low code platform. These were:

1. Their wider IT strategy has already adopted Office 365 and Azure, so they were not increasing the size of the organisation's IT footprint.
2. They had already accounted for the licence costs.
3. The end-users were already authenticated and familiar with the Office 365 user experience, which greatly benefited training and end-user adoption.
4. There was a competitive supply of skilled resources for Office 365, Azure and the selected open source tooling they used.

Deltascheme Limited

Old Estate Office
Harleyford Estate
Marlow
Bucks
SL7 2DX
United Kingdom

Tel: +44 (0)1628 401800
Web: www.deltascheme.com

