

Collaboration:

A Recipe For Success In Global Markets Digital Transformation

🕒 Estimated reading time **8 minutes**

Executive Summary

The decision to buy or build is no longer a binary one.

Asking the right questions at the start of a project helps business and IT teams determine how much, if any, of the build needs to be retained in house for reasons of competitive advantage.

Collaborating with the right technology partner, who takes the time to understand how your project fits into your overall strategy, whether for the whole build or simply part of it, can shorten time to market, focus any internal team on what they do best, and reduce delivery risk substantially.

Winston Churchill is believed to have said, 'Success consists of going from failure to failure without loss of enthusiasm.' That might be true in politics, but in the context of managing a global markets digital transformation project, that's hardly going to delight your clients, and even less so your bank's management. This article looks at the challenges encountered by a typical bank faced with the need to update its technology offering, and, based on Caplin's experience of over twenty years, what leads to the most positive outcomes whilst avoiding some of the more common pitfalls.

Asking the right questions

Every project has to start somewhere. Unless generated by a single catastrophic event, most of these projects are not cliff-fall moments, but rather a process of gradual erosion to the point where it's increasingly dangerous to continue along the same path. Different individuals and different teams have different concepts of danger!

- How does an organisation looking to improve its clients' electronic experience maximise its chances of success?
- How do the many talented individuals and teams working in Global Markets at banks, in management, technology, business process and transformation, coalesce around a single project and manage the differing priorities, time-scales and objectives?



RIP RFP?

In order to provide a reasonably accurate quote for a new project, a technology vendor has to understand what it is they are being asked to build. Quite often, the information conveyed to the vendor is less than complete, and focuses exclusively on the 'what' of the project, with little or no background on the 'why', even if this process is conducted under NDA. However, whilst most RFPs are unloved by both creators and respondents, they can serve a useful purpose as a valuable circuit breaker, and encourage a pause in the relentless 'what' and 'when' timetable to consider the 'why' and 'how' of the project. In our experience, superior outcomes and on-going relationships happen with banks that have honestly answered those questions, and shared those answers with us.

Tell us what you want, what you really, really want...

Banks, even just the divisions dedicated to Global Markets, are complex institutions, often built around fiercely defended silos, generating multiple revenue streams, focused on individual and group performance, measured by quarterly targets rolling up into shareholder returns generated by their customer base. The rewards for perceived success are great as is the disapproval of perceived failure. Time is short, people are busy, and collaboration between departments that are all too often competing for the same scarce resources is sometimes not as great as senior management would imagine it to be. Yet with care, and thoughtful leadership, it's entirely possible to achieve a win/win situation not only across departments, but in striking the right balance between internal and external development.

Building a vision

For a technology partner to give a more considered and hence accurate response, it would be comforting to know that the business management team of the prospective client had asked themselves and formulated responses to the following questions:

- **Goal orientation:** Do we have a clear understanding of the developing needs of our target audience/client base? Have we conducted any independent unbiased surveys to validate this? How do we respond to those needs in a way that engages our clients and continues to generate revenue, creates client stickiness and improves our productivity?
- **Context clarity:** Do we understand our own company/department without unrealistic wishful thinking, or a nostalgic view of past greatness? Who are our current and future competitors and what would we do to capture our market share and client base if we were in their place?
- **Understanding trade offs:** What/who will benefit/suffer if we do/don't complete this project? Do we fully understand the interdependencies of what we are suggesting with other systems in the bank? In a perfect/non-political world, which other departments could benefit from what we are planning? What future proofing framework do we need to have in place to strike the right balance between action now and analysis/paralysis?
- **Impact:** How will the changes we are considering impact the bottom line and, as importantly, how will not deploying them affect the business? What might the long term impact be on client retention/acquisition and perception of our organisation?
- **Commitment:** We know that investment and budget cycles should consider the long term, but that external and internal events, and the constraints of quarterly reporting encourage short termism. What steps do we take to ensure that we are not left with the worst of both worlds i.e. sunk cost and an uncompleted project?

Communicating goals

Assuming that the factors above have been considered, it's unlikely that all of this information would find its way into an RFP for the obvious reason that it's commercially highly sensitive. Yet even some of the information would make a huge difference to a technology vendor compiling a response and would

also enable the bank, on receiving that response, to assess more effectively whether that vendor demonstrated an understanding of the challenge or was simply trying to shoehorn in an unsuitable product.

A minimum of useful relevant information, provided under NDA, might look as follows:

- A vision statement that explains how the replacement system will help satisfy clients and improve internal productivity alongside a realistic roadmap of future enhancements. This addresses the 'why?' of the challenge.
- A detailed description of several typical users of different elements of the proposed solution ; these 'personas' could be external clients, internal sales/traders/administrators, and other potential users within the bank.
- A description of the current system used - what it does well and its limitations, positive and negative feedback from users. What has changed in the way the business operates since the existing system was built/deployed? (Many systems we have replaced were conceived of/built before the advent of the smartphone!)
- Some context of existing or identified future systems with which the solution will have to integrate.
- The extent of any in-house development team and its relative strengths.

Furnished with such information, the next stages of how to build the solution, who to provide which elements and hence the overall cost of the project become much easier to consider and the results correspondingly more meaningful.

Avoiding the Prisoner's Dilemma

The prisoner's dilemma is a game theory experiment involving two people, each of whom can cooperate for mutual benefit or act in their own self interest, believing it will result in an optimal solution for them personally. It rarely does. Its relevance here is that, as well as having to negotiate different agendas within

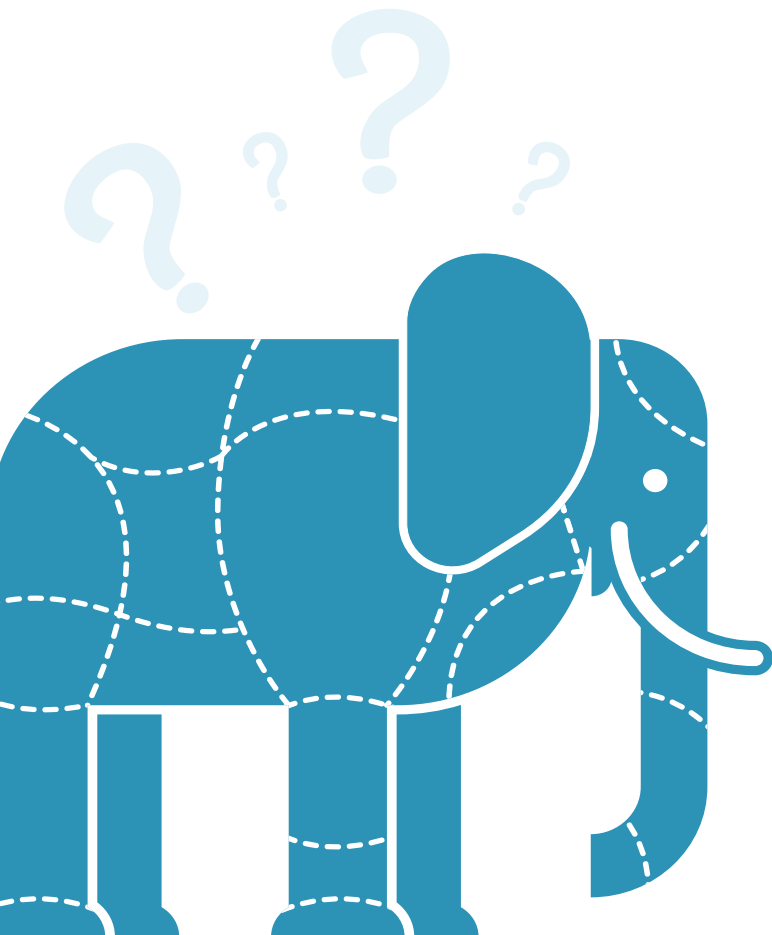
be seen to complicate matters further. It doesn't have to be that way.

Traditional thinking when it comes to build vs buy have tended to focus on three alternatives; build in-house, buy an off-the-shelf system from a vendor, or hire a third party to build the system for you. Each has strengths and weaknesses and presents the chooser with opportunities and threats specifically:

- Level of control over the final offering
- Competence and availability of in-house expertise
- Proven reliability of solution
- Ability to integrate with existing systems
- Future-proof
- Cost - including full internal costs
- Likelihood of derailment of project

Few banks or brokers that are not technology challenger newcomers have the luxury of starting a build totally from scratch. There are always systems with which to integrate; Core Banking Systems, rate engines, credit modules, trade booking systems and often all of these. Many can be years if not decades old and only tenuously supported.

What if there was a third way that combined the benefits of internal capabilities with external expertise? Caplin has spent years developing flexible, well-documented APIs, toolkits and components to work with clients to deliver projects and reduce uncertainty. But it's not just the solutions themselves, or the well-documented toolkits and components that make the difference. A big part of getting to the point of comfort with an external technology provider is mindset - do these people, from C-Suite, sales management, product owners and developers share our values, and really understand what we are trying to do? Can I trust them to be honest on what's software vs slideware, practically realisable within the constraints of the project, or simply pipedreams?



How to eat an elephant

It's an old joke, but the answer is 'One bite at a time!' To make sure no one bites off more than they can chew, one of the first steps we like to take with a client is an **in-depth analysis** of what we are being asked to provide. It's normal for us to be able to gather trading, sales, ecom, IT and management together at the start of the project to make sure all parties understand the rationale behind, and the scope of the project.

Once we have a **clear idea of the objectives**, we can break down the project into phases, making sure that at each stage, all parties know what is expected of them, not only on our side, but also on the client side with regard to exposing APIs, providing test environments for integration and throughout the process, using an Agile methodology to ensure we are always on an agreed right track. The unexpected does happen and sometimes for reasons outside the control of any of the parties involved. Good levels of **regular communication** and a willingness to be flexible on both sides make for a winning **partnership** and a successful delivery.

2 become 1

The concept of partnership lies behind the word 'collaborate' as an alternative to the simple buy or build equation. There may well be components that even the best resourced bank IT department would rather not spend time developing, whether they are core elements of the platform back end, or specialised GUIs, preferring to focus on what they feel gives them a competitive advantage. In that case, it's important that the vendor with whom they choose to partner is **open and transparent** as to how their technology can fit into their framework. Are the modules or components **well documented**? Are there clear instructions on how to use them, with code examples? Is the installation and upgrade process clearly explained? We like to think that our [developer site](#) is a good example of openness and comprehensive documentation, built specifically for developers. One of the drawbacks of a traditional external build is the 'black box' concept. Knowing which features can be customised and how, and understanding the **potential impact of customisation** on the upgrade cycle right at the start is a way to avoid expensive headaches further down the line.

Deciding **how and where to deploy** the system, whether locally or cloud based is essential in the short and long term. If the bank is in the process of slowly moving to a cloud based model within the next few years, it's important to contemplate now how any locally deployed solution can be lifted and shifted at a later date.

The screenshot displays the Caplin Documentation website. On the left is a navigation menu with sections: Home, Downloads, API Reference, Archive, Blog, Training, PRODUCTS (listing Caplin Platform 7, Caplin FX Professional, Caplin FX Sales, Caplin FX Mobile, and Caplin FX Corporate), INTEGRATION (listing FX Integration API, Caplin RET Adapter Toolkit, Barracuda Order Adapter, WallStreet Adapter, and Caplin Director), and CAPLIN TRADER (listing Caplin Trader 5 and Caplin Trader 4). The main content area is titled 'Caplin Documentation' and features a grid of nine product cards, each with a title and a brief description:

- Caplin Platform 7**: A suite of low-latency servers and libraries to support online trading applications.
- Caplin FX Professional**: Desktop web application for bank customers.
- Caplin FX Sales**: Desktop web application for sales teams trading on behalf of bank customers.
- Caplin FX Mobile**: Mobile web application for bank customers.
- FX Integration API**: Java library used by integration adapters that provide data to Caplin FX Suite applications.
- RET Adapter Toolkit**: A developer framework for integrating the Caplin FX Suite with Refinitiv Electronic Trading.
- Barracuda Order Adapter**: An adapter for integrating the Caplin FX Suite with the Barracuda FX order management system.
- Wallstreet Adapter**: An adapter for integrating the Caplin FX Suite with Wallstreet Systems WSS Core.
- Caplin Trader 5**: SDK for developing online trading applications. The foundation of Caplin FX Suite.

The footer contains links for Home, APIs, Archive, Contact Us, Cookies, Third-party Licences, Terms & Conditions, Privacy, and Accessibility.

Conclusion

For many banks, embarking on a digital transformation project with the aim of improving the lives of their clients, sales team and operations department can be a daunting process, fraught with uncertainty. Making the decision to collaborate with a specialist provider as a trusted partner can de-risk the project, improve the ROI and help provide a cohesive customer experience. At Caplin, we firmly believe that asking the right questions at the start can result in an optimal experience for all parties. For further information please contact us [here](#).