

We need to talk about legacy IT architectures

16/04/2019

By Cliff Moyce, DataArt Global Head of Financial Services Practice



Cliff Moyce, DataArt Global Head of Financial Services Practice

Unlike their challenger bank siblings and fintech cousins, incumbent banks have one particular problem to solve if they are to remain viable and competitive. That problem is high cost to income ratios of (typically) around 60%. Compare that figure to fintech firms engaged in 'unbundling the bank' who even when fully operational are operating at ratios as low as 20%.

A large part of the difference in costs is the cost of operating and supporting legacy system architectures. Other factors include the cost of branch networks, and the (over)staffing implications of functionally divided organisations. High IT infrastructures costs in large banks arise from significant duplication and hidden redundancy; poor integration; high complexity; poor systems documentation and knowledge; a lack of agility/ flexibility/ adaptability; old fashioned interfaces and

reporting capabilities; difficulties integrating with newer models such as cloud computing and mobile devices; being difficult to monitor, control and recover; and, susceptible to security problems.

Getting old and new applications, systems and data sources to work seamlessly can be difficult, verging on impossible. This lack of agility means that legacy systems in their existing configuration can be barriers to improved customer service, satisfaction and retention. In regulated sectors they can also be a barrier to achieving statutory compliance. Pressure to replace these systems can be intensified by new competitors who are able to deploy more modern technologies from day one.

One radical approach to solving the infrastructure issue is to design and implement a new, more modern architecture using a radical clean-slate or blueprint-driven approach. Amusing analogies have often been used to encourage audiences to take such an approach, including the analogy of legacy infrastructures resembling an unplanned house that has been extended many times. But how easy is it to design and implement a new IT architecture in a large mature organisation with an extensive IT systems estate?

Rather than the unplanned house analogy, a better analogy might be a ship at sea involved in a battle. Imagine if you were the captain of such a ship and someone came onto the bridge to suggest that everyone stop taking action to evade the enemy and instead draw up a new design for the ship that would make evasion easier once implemented. You might be forced to be uncharacteristically impolite for a moment before getting back to the job at hand.

The temptation to start again is enormous, but big-bang approaches to legacy IT systems replacement can be naive, expensive and fraught with risk. At some point, many large organisations have attempted the enterprise-wide re-design approach to resolving their legacy systems problems. Yet so many initiatives have been abandoned when the scale of the challenge or the impossibility of delivering against a moving target become clear. Time has a nasty habit of refusing to stand still while you draw up your new blueprint. Re-designing an entire architecture is not a trivial undertaking, and building / buying and implementing replacement systems will take a long time. Long before a new architecture could ever be implemented the organisation will have launched new products and services; changed existing business processes; experienced changes to regulations; witnessed the birth of a disruptive technology; encountered new competitors; exited a particular business sector and entered others.

All of these things conspire to make the redesign invalid even before it's live. If you are lucky, you may realise the futility of the approach before too much money has been spent. Furthermore, the sort of major projects required to achieve the transformation are the sorts of projects that run notoriously high failure rates. A 2005 KPGM report showed that in just a twelve month period 49% of organizations had suffered a recent project failure, with IBM later reporting in 2008 that only 40% of the projects met their schedule, budget and quality goals. And as recently as 2012, a McKinsey and Company report identified that 17% of large IT projects fail so critically as to threaten the very existence of the company.

So if wholesale blueprinting and re-engineering is impractical, what options are left to solve the problem? Luckily there are some practical and cost effective approaches that can mitigate many of the problems with legacy systems while obviating the immediate need to replace systems (though eventual systems replacement should be an objective). Two viable alternative approaches are service-oriented architecture (SOA) and web services. Used in combination, they offer an effective solution to legacy systems problem.

SOA refers to an architectural pattern in which application components talk to each other via interfaces. Rather than replacing multiple legacy systems, it provides a messaging layer between components that allows them to co-operate at a level you would expect if everything had been designed at the same time and was running on much newer technologies. These components not only include applications and databases, but can also be the different layers of applications. For example, multiple presentation layers talk to SOA and SOA talks to multiple business logic layers – and thus an individual presentation layer that previously could not talk easily (if at all) to the business logic layer of another application can now do so.

Web services aims to deliver everything over web protocols so that every service can talk to every other service using various types of web communications (WSDL, XML, SOAP etc.). Rather than relying on proprietary APIs to allow architectural components to communicate, SOA achieved through web services provides a truly open interoperable environment for co-operation between components.

The improvements that can be achieved in an existing legacy systems architecture using SOA through web services can be immense, and there is no need for major high risk replacement projects and significant re-engineering. Instead organisations can focus on improving cost efficiency by removing duplication and redundancy through a process of continuous

improvement, knowing that their major operations and support issues have been addressed by SOA and web services. Another benefit is that the operations of the organisation can start to be viewed as a collection of components that can be configured quickly to provide new services even though the components were not built with the new service in mind. This principle is known as the composable enterprise.

But addressing the issue of legacy systems in a way that makes good sense is not just an IT issue; it is also a people issue. It requires people to resist their natural inclination to get rid of old things and build new things in the mistaken assumption that new is always better than old. It requires people to resist the temptation to launch 'big deal projects', for all of the reasons that people launch big deal projects – from genuine belief that they are required (or the only way), to it being a way of self-promotion, and everything in-between. It requires people to take a genuinely objective view of the business case for change, while operating in a subjective environment. It requires people to prioritise customer service over the compulsion to tidy up internally. And, it requires the default method of change to be continuous improvement rather than step change projects – which can be counter intuitive in cultures where many employees have the words 'project' or 'programme' in their job titles.

So, to summarise, of course legacy enterprise IT architectures can feel like barriers to efficiency, agility and customer satisfaction and making even the smallest change can often feel like it takes too long and costs too much money. The overwhelming temptation to throw the legacy architecture away and start again is understandable, but succumbing to that temptation can be a mistake. Luckily we now have technical tools and approaches available to affect radical improvements without having to incur the expense, effort and risk of major replacement projects. But using these tools comes with a change of mindset and approach that may be counter-cultural in some organisations. It can mean a move away from step-change and 'long-march' projects, and a move towards continuous improvement. Education and engagement will be one of the keys to making it happen.

Original publication: <https://www.globalbankingandfinance.com/we-need-to-talk-about-legacy-it-architectures/>