

# The **application modernisation challenge**

Key factors and considerations when  
modernising business critical systems



[nashtechglobal.com](http://nashtechglobal.com)

Delivering  
technology  
**excellence**

**Nash  
Tech.**

# Table of **contents**

Introduction .....	3
Why modernise? .....	4
Why not go for a big bang replacement? .....	5
The phased approach – following the ‘strangler pattern’ .....	6
Key advantages of the phased / strangler approach .....	9
How NashTech can help .....	10
Client story .....	11

# Introduction

Many businesses become dependent on IT systems or applications that later need to be modernised. With technology moving and developing so quickly, and business needs growing with it, this is inevitable.

This may happen because an application that an organisation depends on becomes unsupported by the software provider. Or it may happen because a business buys an off-the-shelf application and then customises it to such an extent that it becomes effectively a bespoke system and requires updating.

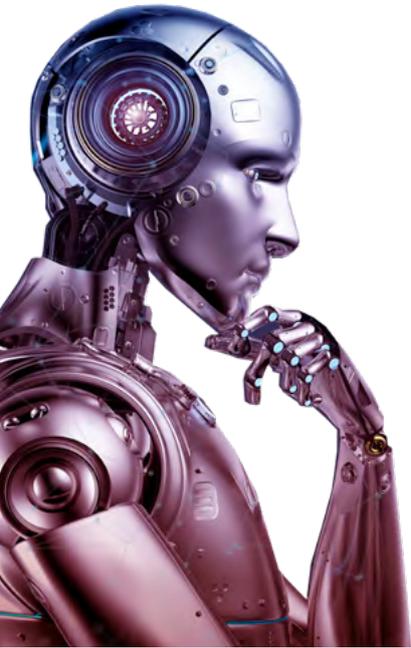
A business may have created its own unique system from the outset to run business critical applications. This system may have become steadily more complex over time as new functions and components have been added, with increasing numbers of applications integrating into the central system. Eventually, the point is reached where this system needs to be overhauled and modernised.

For some organisations, the experience of operating through the Covid-19 pandemic may have resulted in a closer working relationship between IT and the business and a stronger partnership culture may have formed. At the same time, businesses have learned new degrees of adaptability and agility. These are all factors that may lead the organisation to believe the time is right to take action and address their system needs.

In this paper, we discuss the common considerations facing businesses in such a position, the alternatives open to them, and recommended approaches to achieving successful modernisation.

**We do this from NashTech's experience of working with organisations across sectors and of all sizes – modernising systems and applications, developing new software through agile and DevOps approaches, and rigorously testing throughout the journey.**

# Why modernise?



Before we turn to the detail of modernisation, it is worth discussing why it is necessary at all. Previously organisations may have had more time to respond to changes, but today changes to business processes, and the systems that support them, happen more frequently and require changes to be implemented much more quickly. Solving issues as they arise, and so avoiding the need for long, complex and costly upgrade or replacement projects, is something that is of the utmost importance.

This may be effective for a certain period of time – but it is almost certain that old and outdated systems will come to a crisis point eventually. This becomes increasingly likely the more that business demands on the organisation's IT infrastructure grow. Even if an eventual 'meltdown' scenario is avoided, it is likely that old systems will significantly hold the business back due to several potential issues:

- the system becomes a risk to the business. Changes in one area can cause cascading faults across the application
- changes are taking excessive amounts of time to plan, deliver and test - and in some cases change not being attempted at all
- because the system has been developed over a number of years, in an ad-hoc fashion, there may be limited documentation, which further hinders changes and improvements
- changes to the system are not always documented, resulting in no one having a complete picture, and the continued operation of the system being dependent on the knowledge of a few key individuals
- the limitations of the system start to constrain the business, making responding to changes or seizing new opportunities more difficult



**For these reasons, it is likely to become unavoidable that the system needs to be extensively modernised or replaced.**

# Why not go for a **big bang** replacement?

Once the decision has been taken to replace the the existing system, the first question to arise is whether to overhaul it in phases, or go for the 'big bang' approach.

Depending on the nature of the system or application, an off-the-shelf system such as an ERP application from one of the big vendors, such as Oracle or SAP, may fulfil all the needs of the business.

So, why not simply go down that route? Speak to a big vendor, discuss any customisation requirements, agree a price, and then work with an IT consultancy on the implementation plan?

There are a number of reasons why this may not be the best solution for some businesses:



**cost** – an off-the-shelf systems implementation is likely to carry a hefty price tag



**customisation** – if the business has become accustomed to a series of very specific functionalities through its existing application, the implementation may require a number of customisations, adding to both the complexity and cost of the project



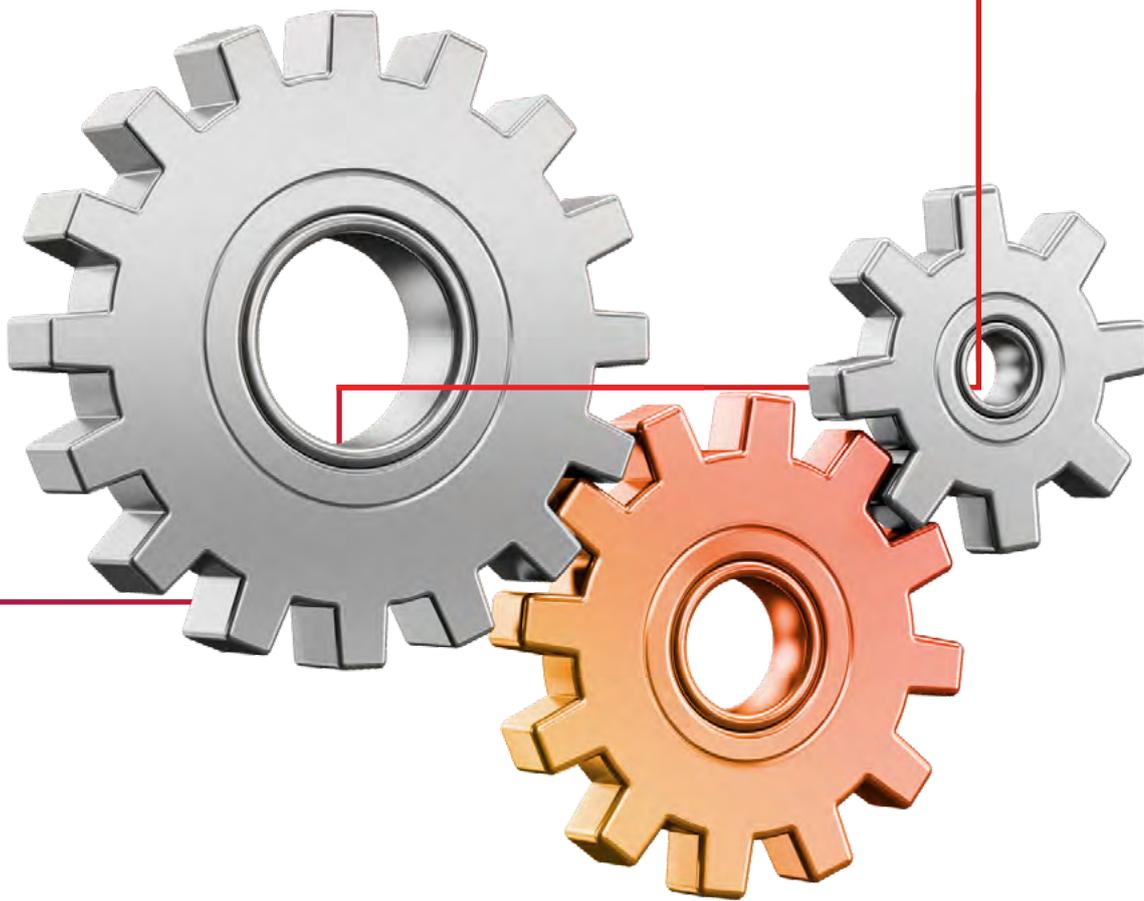
**unused functionality** – big off-the-shelf systems, such as ERP systems, are likely to include significant amounts of functionality that a business may never use – effectively wasting money on unutilised benefits



**time scale** – an implementation is likely to take a number of years, meaning that the business will need to carry on with and maintain its existing system



**cliff edge transition** – no matter how much planning and simulation is built in, eventually there will be a 'moment of truth' when the switch over happens to the new system



## The **phased approach** – following the ‘strangler pattern’

It is for the reasons above that many organisations choose instead to modernise their systems or applications over time. However, this raises a different but inevitable challenge: how to overhaul a system while it is still in use? Clearly, there is a danger that in modernising parts of the system, technical issues could arise that cause disruptions and outages that no business can afford to countenance.

**In our experience, there is a clear roadmap to follow for a successful modernisation programme.**

---

## Enterprise architecture review

# 1

Firstly, the organisation should conduct a comprehensive review of its complete enterprise architecture. What does the application to be modernised consist of, and what key functions does it perform? What other applications integrate or talk to it?

This mapping process will provide clarity and may also help identify certain applications that simply aren't used any more that can be switched off at the outset, simplifying the landscape.

Once a clear view of the enterprise architecture has been obtained, planning can begin for the process of upgrading it.

---

## Buy or build?

# 2

Another key early question is to consider which parts of the application should be reconfigured by the internal or consultancy team – and which parts can be replaced wholesale with a bought-in solution?

### Create the roadmap

The information on the 'as is' and 'to be' states captured in the review should be used to create a roadmap for systems modernisation. This will set the priorities for extending the life of, replacing or decommissioning the applications and services currently in use. It will also create a timeline for completing the process, showing the quick wins that are possible and the longer-term strategic priorities.

As part of this process, decisions will need to be made on whether to buy or build. In many cases the answer will be a combination of off-the-shelf software, bespoke and software-as-a-service. Breaking the system up in to component services offers the flexibility to choose the best solution for a specific purpose.

---

## The strangler pattern

# 3

For the parts of the application to be reconfigured or modernised internally, our recommendation is to follow an established methodology known as the strangler pattern.

The strangler pattern is an approach to application modernisation that seeks to de-risk the project by freezing changes to the existing application and replacing key functionality, rather than the whole application. The legacy application can co-exist with the new functionality, with the application being replaced over time. This leads to smaller projects which are less likely to fail, and which cause fewer issues if they do.

It is called 'strangler' because once the new functionality has been built, the old component is 'strangled', the new service is activated, and the old component is decommissioned.

A key feature of the approach is that an abstraction, or API layer, is built which sits above the old component to be replaced, meaning that when changes are made to it, there will be no impact on the rest of the system. This way the application can be overhauled piece by piece with no danger of the whole system falling over.



# Key advantages of the phased / strangler approach

In our view and experience, there are a number of clear advantages to the process described above including:



**risk is managed all the way through** – it's a managed process that keeps risk below an agreed acceptable level



**earlier value is achieved throughout the project** – instead of having to wait until the whole project has been completed as with an off-the-shelf big bang implementation, the business can receive the benefits of improved functionality with each stage. Benefits build up incrementally through the process



**all functionality becomes fully documented** – at each stage of the modernisation, technical specifications and configurations are fully documented so that the business has a complete open-source technical guide that everyone can access. The problem of knowledge residing in specific individuals is overcome



**costs are likely to be lower than a big bang off-the-shelf implementation**



**timescales should be no longer than a big bang implementation** – with changes and benefits being unlocked progressively throughout

This is not to say that the strangler approach will be the most suitable methodology in all cases. Some businesses may decide that an off-the-shelf implementation better suits their needs. This is something that needs to be assessed and discussed with each organisation individually with a decision made according to its own unique profile, priorities and IT objectives.

# How NashTech can help



We have extensive experience of large and small scale modernisations.

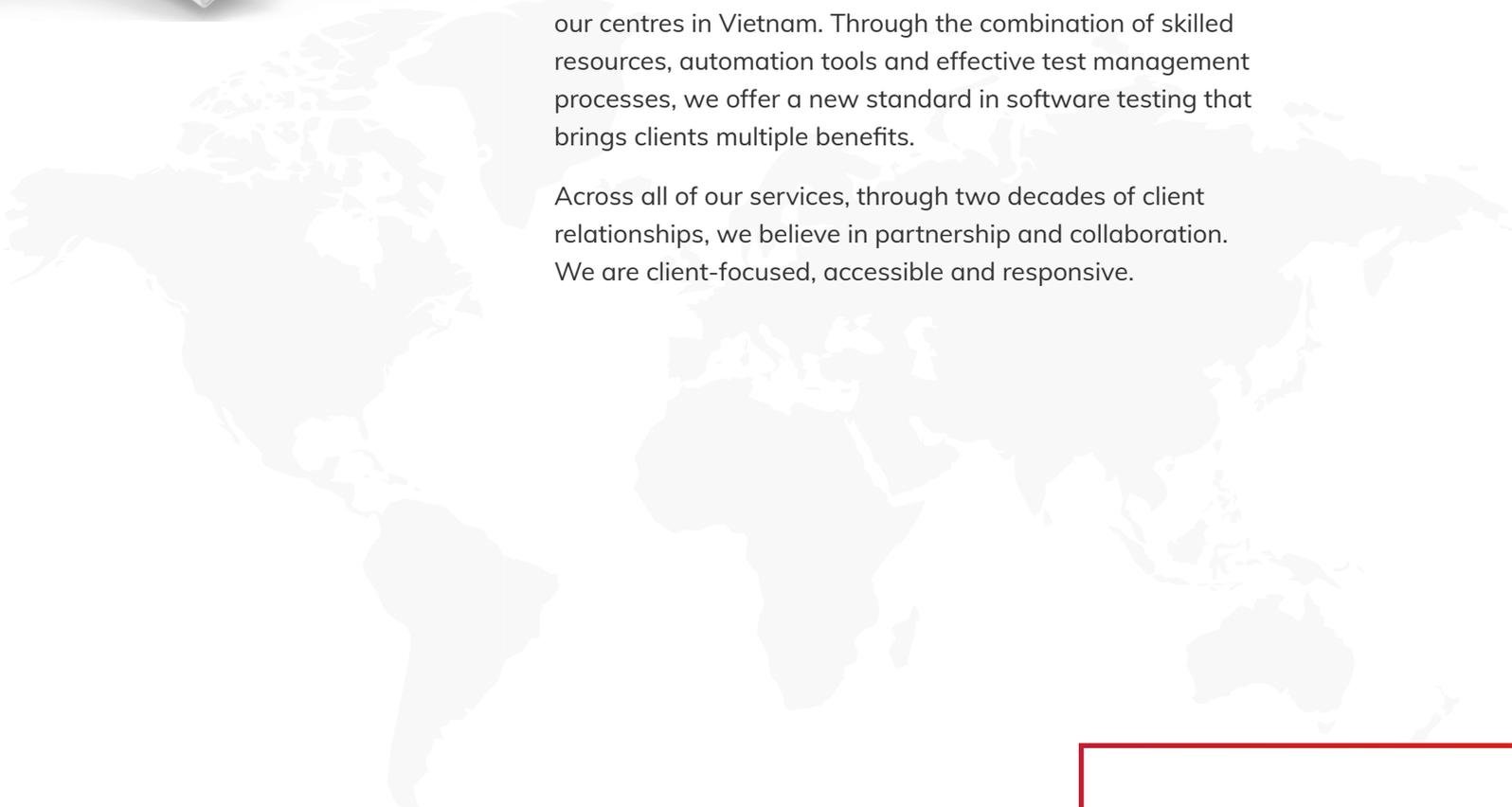
**Software development is one of our core competencies** – having been designing, testing and implementing bespoke software solutions for businesses across sectors for over 20 years. We help clients across the complete software lifecycle, from business vision and development through to maintenance and enhancement.

**We embrace lean and agile principles that are iterative, interactive and collaborative** – fostering a ‘start-up’ mindset that encourages innovation and creativity and which is not afraid to pivot the approach.

We are UK-headquartered, with a highly experienced, motivated and qualified team of software developers based offshore in Vietnam. Backed up by a team of local expertise who liaise directly with clients and direct the offshore development work.

**Testing is also at the core of our business.** We have a highly skilled and experienced team of over 300 testers based in our centres in Vietnam. Through the combination of skilled resources, automation tools and effective test management processes, we offer a new standard in software testing that brings clients multiple benefits.

Across all of our services, through two decades of client relationships, we believe in partnership and collaboration. We are client-focused, accessible and responsive.



# Legacy system upgrade to support international growth



**We helped this global business services organisation to deliver its major applications modernisation by replacing legacy systems with a single technology platform.**

## Company overview

Our client collects and validates supplier data and mitigates risks globally. This data-driven insight builds more secure, sustainable, better performing supply chains.

-  **Client name:** Confidential
-  **Service:** End to end software development
-  **Technology:** .Net
-  **Industry:** Supply chain management
-  **Location:** Global

## Outcome

Over two years NashTech's scrum teams have completed some +50 sprints to deliver the solution over 20 increments.

The key metrics have demonstrated a continuous improvement in team sprint velocities as well as consistently high quality supported by test automation.

NashTech's teams have worked highly collaboratively with both the client's global teams and other delivery partners. Excellent communications and team morale has supported very low turnover during the entire project.

The new platform has already demonstrated significant operational costs savings as well as providing the flexibility to support international growth and new business models.

## The challenge

This global business services organisation had grown through acquisition and found itself with a business critical applications portfolio that was costing a lot and leaving few resources to focus on innovation and growth. The organisation had experienced traditional offshore delivery and needed a partner who they could depend upon and trust to deliver.

## The approach

A scaled agile framework was used together with a continuous value delivery chain (DevOps). A particularly complex code branch and release environment was required due to the landscape. A robust governance model was established with comprehensive metrics and strict quality criteria.

## The solution

The objective was to replace around 30 legacy applications with a single technology platform with interfaces into the systems of record.

We are experts in technology, delivering smart solutions that solve business challenges and create value. Our award-winning teams apply deep expertise and passion to deliver complex IT projects globally.

 @NashTechHN

 NashTech Limited

For more details, please send your enquiry to [info@nashtechglobal.com](mailto:info@nashtechglobal.com) or visit our website [www.nashtechglobal.com](http://www.nashtechglobal.com)

**Nash  
Tech.**