

Bring your legacy software back from the brink of extinction

A practical guide to modernising systems for a digital future.





Contents



A different approach to the challenge of legacy software	4
What exactly do we mean by legacy?.....	5
When does legacy become a pressing challenge?	6
Hidden Costs of Legacy	6
Risks of Legacy	7
Addressing legacy challenges for AI success	8
What are your options when it's time to move on?	9
A cost-effective approach that puts your business on the right path?	10
Encapsulation - Enabling Easier Integration	11
Replatforming: A Strategic Move	13
Legacy Modernisation in action: Rearo.....	14
Refactoring: Enhancing Agility and Maintainability	15
Rearchitecture: Empowering Innovation and Scalability	16
Legacy Modernisation in action: Stanley Black & Decker	17
Legacy Modernisation in action: Crown Commercial Service	18
But what about the drawbacks?	19
How do you decide which approach works best for your business?	20
Choosing the right partner for your business system modernisation journey	22
About PDMS.....	24



Embrace, Evolve & Excel

A different approach to the challenge of legacy software.

Digital transformation is a top priority for businesses across industries and sectors, and a key challenge lies in modernising the legacy software that supports essential operations. As AI becomes more integral to innovation, outdated systems have also come under scrutiny for blocking AI adoption and limiting the potential business gains it offers.

Should you replace legacy software with an entirely new system? Or would it be better to adopt a gradual migration approach to incrementally upgrade the legacy application?

In this paper, we delve into the challenges of dealing with legacy or heritage applications and the different approaches you can take to modernisation.

We take a deeper dive into the benefits of the different migration options as a more manageable alternative to the 'rip out and replace' approach. We also include some practical examples of organisations, both large and small, who have tackled the legacy challenge with different technical modernisation strategies.

What exactly do we mean by legacy?



Many organisations run on applications and systems that were built many years ago – in some instances more than 20 years ago!

These systems are vital to the company, as they contain a wealth of valuable data and business logic, and touch on many different elements of the organisation's operations. Staff are familiar with them, use them daily and they are deeply ingrained in the business' processes. In many cases, they have been enhanced and upgraded over the years and represent a long-standing investment for the company both in terms of time and money.

The term 'legacy' implies that the software is built on older technologies, architectures, or programming languages that may not align with current industry standards or best practice. You may also hear legacy referred to as 'heritage' software, which acknowledges the software's contributions – recognising its value and the investment made in developing and maintaining it over time. It also reflects the idea that the software has a rich legacy and holds institutional knowledge and expertise that you may want to preserve.

Legacy software is an issue across many different sectors from financial services and retail through to manufacturing and the public sector. According to the National Audit Office more than 25% of the UK government's total digital budget is being spent on reducing legacy IT and cybersecurity.

When we refer to legacy systems they often have some of the following characteristics:

- End of life with vendors having discontinued support and development
- Outdated architecture
- Reliance on expensive hardware e.g., old mainframe computers
- Custom built solutions which have become outdated over time
- Shortage of professionals who can support the outdated technology.

Importantly, these systems are still used as part of an organisation's daily operations and in most cases are business critical systems. They have stood the test of time and work well enough. An organisation's long-term employees are familiar with them and the fear of disruption has delayed the decision to tackle the legacy problem. Some companies are still running systems written in COBOL which debuted in 1959!

More than half of legacy applications are core to the business



Source: Infosys Knowledge Institute

When does legacy become a pressing challenge?

There's a whole host of reasons why legacy technology - whether software or hardware - is not only holding businesses back but why it is fast becoming an increasing business risk. It hampers compatibility and integration, poses security risks, restricts scalability, increases maintenance costs, reduces efficiency, and can put businesses at a competitive disadvantage.



Hidden Costs of Legacy

Lack of integration and compatibility

Legacy software often lacks the necessary integration capabilities, resulting in information silos and fragmented data. They may lack the ability to integrate with newer technologies, such as AI, cloud computing, Internet of Things (IoT), or advanced analytics platforms. This hampers efficient decision-making, process automation, and real-time visibility across the organisation.

Impact on productivity

Legacy systems frequently require significant manual data entry, so processes can become more time consuming and prone to errors. These systems may lack modern user interfaces and intuitive workflows, resulting in reduced user productivity. In addition, there is often a lack of documentation for systems that have evolved over many years which means a steep learning curve for new users. Outdated software with slow response times and frequent system crashes can disrupt processes, leading to production delays, quality issues, and customer dissatisfaction.

Maintenance and support

Often legacy systems are 'end of life' meaning they are not always supported by the original supplier, making it difficult to get updates, patches and technical support. Additionally, the cost of maintaining legacy systems can be high, as they may require specialised hardware or software that is no longer readily available. Finding skilled professionals with expertise in maintaining and troubleshooting these systems becomes more difficult and expensive. Companies may need to invest significant financial and human resources to ensure the continued operation of legacy technology, diverting funds and focus away from strategic initiatives and innovation.

On-premise vs cloud

Some legacy systems run on-premise and struggle to scale to meet the growing business demands. The systems are not architected to take advantage of a distributed approach and modern cloud services, and the costly exercise of updating or adding more hardware only gives diminishing returns. It is hindering the ability to move to the cloud to take advantage of more cost-effective data storage and the advanced features, security and reliability offered by cloud providers.

Risks of Legacy

Security and compliance

Legacy technology often lacks the robust security measures required to protect sensitive data from evolving cyber threats. Outdated software and hardware may have vulnerabilities that make them more susceptible to breaches, putting valuable intellectual property, customer information, and trade secrets at risk. Additionally, compliance with industry regulations and data protection standards becomes more challenging with legacy systems, potentially leading to legal and reputational consequences.

Performance

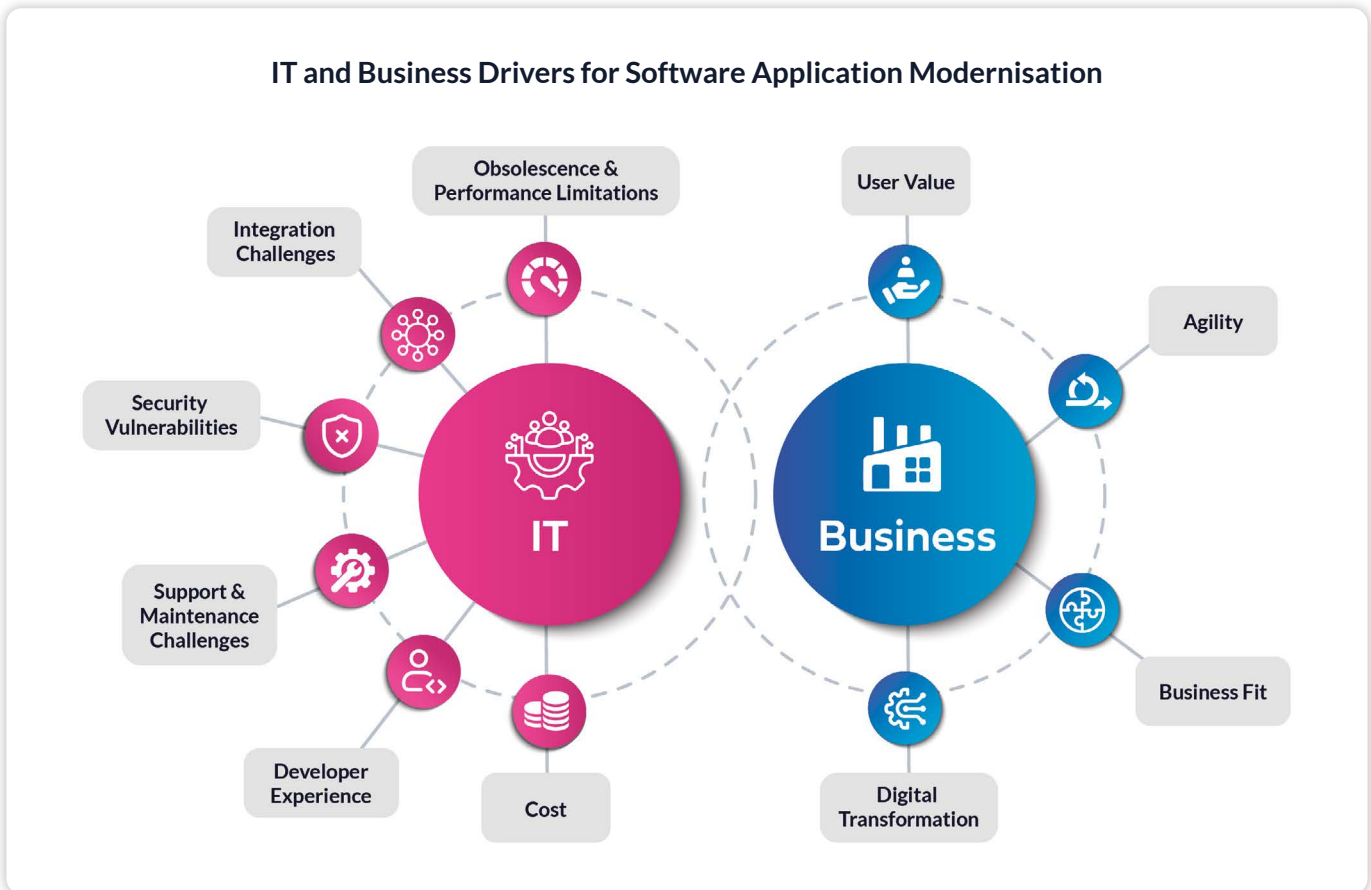
Older technologies, programming languages and frameworks may not be optimised for modern hardware and software environments. Consequently, they may run slower and be less efficient compared to contemporary applications.

Access to skills

As technology evolves rapidly, the pool of professionals with expertise in older programming languages, outdated frameworks, or legacy systems diminishes over time. As new technologies gain prominence, there is a shortage in the market of the specific skills required to understand, maintain and enhance legacy software systems.

Agility

Companies must be able to adapt to changing market demands, scale their operations, and explore new business opportunities. Legacy technology can impede these endeavours by limiting scalability and flexibility. Outdated systems may struggle to handle increased data volumes, support growing user bases, or accommodate new functionalities. This lack of scalability and flexibility hampers agility and competitiveness, making it challenging for companies to respond swiftly to evolving market dynamics.





Addressing legacy challenges for AI success

Most organisations are looking into how they can unlock the potential of AI to automate tasks, provide insights and enhance decision-making. However, legacy systems and infrastructure can significantly slow down the adoption of AI.

One of the biggest challenges with legacy systems is that they weren't designed to handle the vast amounts of data or complex calculations that AI relies on. AI thrives on real-time data, advanced algorithms, and the ability to process huge datasets quickly. Unfortunately, legacy systems tend to be slower, less flexible, and harder to integrate with new technologies. Data is often stored in outdated formats and non-standard structures making it difficult to extract, transform and load data into AI applications.

Additionally, legacy systems often lack the computing power and scalability that AI requires. AI systems need to adapt and grow as they process more data, something older systems weren't built for.

Security is another roadblock to AI adoption. Legacy systems are more vulnerable to security threats because they may not receive regular security updates or patches. This can pose significant risks when integrating AI, which could exacerbate security and data privacy risks.

Many of these factors will also be roadblocks for the adoption of other disruptive or transformative technologies such as the Internet of Things (IoT), big data analytics and Automation and Robotics.

What are your options when it's time to move on?

The approaches you can take to dealing with the modernisation of legacy applications were nicely summed up by *Gartner* who also ranked them by ease of implementation.



(Gartner: 7 Options to Modernize Legacy Systems - 2019)

A cost-effective approach that puts your business on the right path

In some instances, rebuilding or replacing will be the best option, but migrating to new systems, often in a 'big-bang' approach when a business switches from their old system to a brand new system at a single point in time, can be daunting, and very disruptive.

In a recent survey, Infosys found that 68% of respondents who had a higher-than-average number of big-bang projects (39% or more) experienced more frequent crippling disruption. The frequency of crippling disruption for a phased modernisation method was far lower.

In this paper, we are focusing on key areas we believe offer a balanced approach between benefit, risk and cost for many organisations – encapsulate / replatform / refactor / rearchitect. These approaches offer a compromise and often a cost-effective approach to dealing with the issues of legacy software as part of a longer-term strategy to reap the benefits of digital transformation.

They enable companies to maintain business continuity whilst simultaneously upgrading software platforms to improve productivity. By migrating applications and systems in stages, downtime is avoided and disruption to users, frequently created by a 'big bang' approach, is minimised. This is especially important in sectors with mission-critical software that has a significant impact on production, or service delivery, if they are taken offline.

In addition, by gradually transitioning your software, the impact on your staff, customers and business performance will be easier to manage.

It is also worth highlighting that encapsulation can be a useful first step in the modernisation process, especially when it comes to re-platforming. It provides a low barrier of entry approach – encapsulating the reliable, line of business functional blocks and exposing a well-defined and documented API for integration with wider systems. These functional blocks can then be modernised by refactoring, rearchitecting, rebuilding or replacement. We are going to describe these key approaches to modernisation in the following section - considering the advantages of each of them in a little more detail.



Encapsulation - Enabling Easier Integration

Encapsulation involves leveraging and extending the features of an existing application by encapsulating its data and functions and making them available as services through an Application Programming Interface (API).

Encapsulation is like designing a smart home system for an old house. You want to enhance its functionality and ease of use without changing the house's exterior or its overall purpose. Just as you organise and streamline the house's functions with a centralised control system, encapsulation organises and centralises data and functions in software without altering its visible behaviour.

This approach allows organisations to update and enhance their legacy software systems without completely rewriting them, enabling easier integration with new technologies and platforms. It can also help address poor user experience (UX) by providing an improved user interface. Encapsulation is often a low barrier of entry approach to modernisation and a useful first step in a wider replatforming strategy.



Technical soundbite

An Application Programming Interface (API) is a set of rules and tools that lets different software applications communicate. It defines how developers can request and exchange information between software components, simplifying development and enabling integration between diverse systems without worrying about your underlying technologies. An API serves as the bridge between the encapsulated features of the legacy application and the outside world.



The advantages of encapsulation are:

Isolation of Functionality

Encapsulation allows you to isolate specific functionalities or components within a larger system. This makes it easier to identify, modify, or replace individual parts without disrupting the entire system.

Security and Governance

Encapsulation and API development provide opportunities to implement security measures and access controls. You can enforce authentication, authorisation, and data validation mechanisms at the API level, ensuring that only authorised users or systems can access the encapsulated features.

Integration with Modern Technologies

Encapsulating legacy features and providing them through an API enables seamless integration with modern technologies and platforms. This is especially useful when you want to integrate the legacy system with new web and mobile applications, cloud services, or third-party tools. The API acts as an abstraction layer that shields the modern components from the complexities of the legacy codebase.

Modularity and Maintainability

Encapsulation promotes modularisation by breaking down the software into smaller, self-contained units. This modular structure simplifies maintenance efforts, as each encapsulated module can be updated or replaced independently. This makes it easier to identify and fix issues, improve code quality, and ensure changes in one module don't affect others.

Parallel Development

Encapsulation allows multiple teams to work on different encapsulated units concurrently. This speeds up the modernisation process as teams can focus on specific parts without interfering with each other's work.

Replatforming: A Strategic Move

Sometimes referred to as ‘progressive replatforming’, this approach involves gradually migrating a software solution from one platform to another, usually in a modular fashion. The core functionality and data are usually preserved. It’s similar to relocating a house from one neighbourhood to another while keeping the house intact.

The advantages of replatforming are:

Cost-effectiveness

Replatforming allows companies to leverage modern technologies and infrastructure while minimising the cost and effort associated with building an entirely new application from scratch. By reusing existing code and assets, companies may achieve significant cost savings.

Reduced Disruption

Replatforming allows for a seamless transition as it maintains the existing application’s core functionalities and data structures. This approach ensures that day-to-day operations continue without major interruptions, reducing the risk of downtime and productivity loss.

Scalability and Flexibility

Legacy software often lacks the scalability and flexibility needed to meet evolving business requirements. Replatforming enables companies to migrate to cloud-based or modern infrastructure platforms, unlocking new capabilities for scaling, integration with other systems, and adaptability to future needs.

Risk Mitigation

Replacing an entire legacy software system carries inherent risks, including project delays, data loss, and operational disruptions. Replatforming mitigates these risks by taking an incremental approach, ensuring that critical functionalities remain intact throughout the modernisation process.



Technical soundbite

Often replatforming involves moving from a monolith to a microservices approach. Microservices are decoupled and discrete pieces of functionality. Each service can leverage different programming languages, OS, and best-of-breed technologies. The modularity of each service eliminates the need to migrate an entire application when changing or replacing each microservice.

Legacy Modernisation in action

Glasgow based Rearo Laminates is renowned as one of the leading names for commercial laminate surfaces in the UK, with over 30 years' experience of distributing laminate goods across the nation.

Rearo was looking for a local based technology partner to help both support its existing legacy CRM and ERP product and investigate the options for modernisation or replacement.

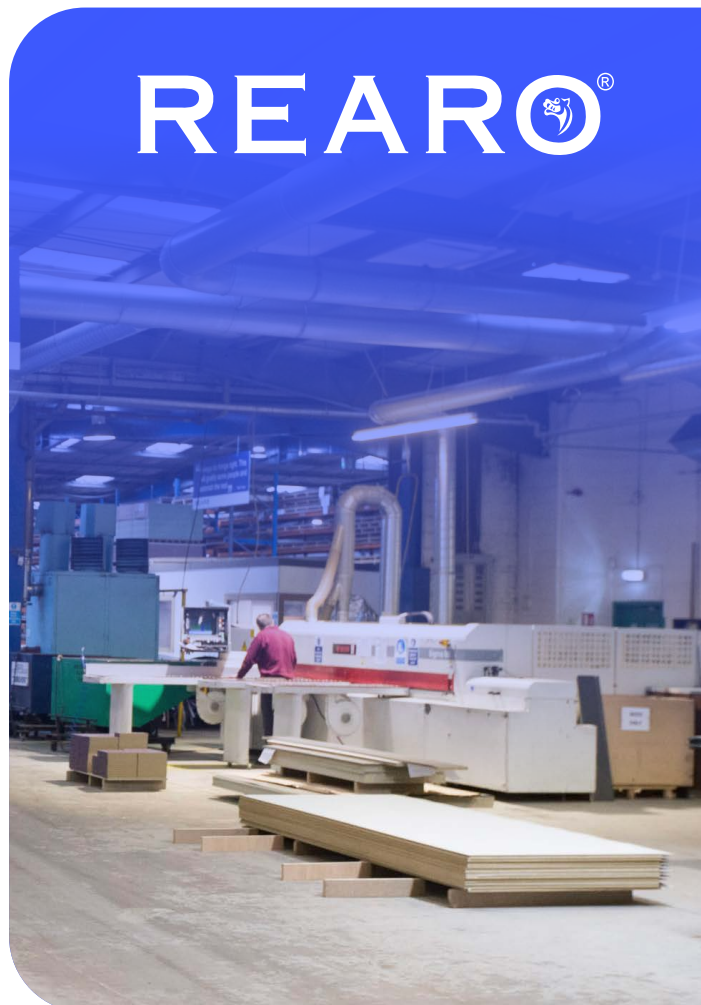
The original product was written in Visual Basic and was on an old unsupported version of Microsoft .NET. Therefore, the PDMS project team decided to carry out a pilot to migrate some of the application to a supported .NET and translate the code from Visual Basic to C#.

Following the successful completion of the pilot and after reviewing the costs and risks associated with a full replacement, Rearo decided to opt for a comprehensive replatform approach. The significant ongoing licensing costs of replacement commercial-off-the-shelf (COTS) ERP systems operating under software-as-a-service (SAAS) models was a major factor in this decision.

By moving to a new modern platform, we have also been able to integrate with third-party software including a new e-commerce website to allow for seamless stock updates and order processing. In addition, we've also helped to improve the customer experience by building click-and-collect functionality.



REARO®



Refactoring: Enhancing Agility and Maintainability

Refactoring involves making improvements to the existing software's internal structure and design without altering its external functionality. It's like renovating a house to make it more organised, efficient, and easier to maintain, without altering its appearance or the function of each room.

In the context of software, refactoring involves making changes to the codebase to enhance its readability, maintainability, and performance. It focuses on improving the underlying structure and organisation of the code without altering what the software does for its users.



The key advantages of adopting a refactoring approach:

Enhanced Performance

Legacy software applications often suffer from performance issues due to outdated code and architectural deficiencies. Refactoring enables the elimination of redundant code, streamlining algorithms, and optimising database queries, resulting in improved speed and efficiency.

Increased Maintainability

Legacy software can become challenging to maintain, with limited documentation and a lack of modular design. Refactoring introduces well-structured code, improved documentation, and modular components, making the software easier to understand, update, and maintain over time.

Agility and Innovation

Refactoring enables the integration of new technologies and features into the existing application, promoting agility and innovation. By breaking down monolithic architectures and embracing microservices or containerisation, companies can iterate quickly, experiment with new functionalities, and respond rapidly to market demands.

Reduced Technical Debt

Legacy software often accumulates technical debt over time, making it harder to implement future upgrades and enhancements. Refactoring allows organisations to address technical debt, improving code quality and reducing the risk of performance bottlenecks, security vulnerabilities, and system failures.

Automated Testing

As part of a refactoring exercise, it is important that some key functional behaviour remains externally unaltered. Automated unit and integration testing can be introduced as a first step to a refactoring project to ensure that the desired behaviour and functionality is maintained. This is particularly important where the module to be refactored integrates with other systems. This brings great advantages for any future development, as the resulting refactored code will have a good level of test coverage, minimising the effort required for manual testing, and reducing the risk of unforeseen issues. A comprehensive, modern automated testing framework also minimises the risk of introducing bugs and errors.

Rearchitecture: Empowering Innovation and Scalability

Rearchitecting a software solution involves making significant changes to its underlying structure and design to take advantage of new technologies, capabilities, and architectural patterns. It is like renovating a house, where you modify the existing structure to make it more modern, functional, and adaptable.

In the context of software, rearchitecting refers to the process of redesigning the software's foundation, often the code and infrastructure, while preserving its core functionality and business logic. It goes beyond simply adding new features or making small improvements. Rearchitecting aims to transform the software from its current state into a more flexible, scalable, and efficient solution. Rearchitecting also involves integrating the software with new technologies or platforms, such as cloud computing, Internet of Things (IoT), artificial intelligence (AI), or machine learning.

The advantages of rearchitecture are:

Preserving Core Business Logic and Knowledge

Companies have built their software applications over years, embedding valuable domain-specific knowledge and business rules into the legacy system. With a rearchitecting approach, the core business logic and knowledge can be retained and seamlessly integrated into the modernised software. Rearchitecting retains valuable domain-specific knowledge and business rules, avoiding the need to redefine and retest core processes. This speeds up time-to-market and reduces disruption.

Enhanced Customisation and Fit-for-Purpose Solutions

One of the primary advantages of rearchitecting legacy software is the ability to customise and tailor the solution to meet specific business requirements. While off-the-shelf software may provide certain features, it often lacks the precise functionalities needed for intricate processes. By rearchitecting, businesses can address specific pain points, optimise workflows, and introduce new capabilities that align with their unique operational needs. This customisation leads to increased efficiency, improved productivity, and a competitive edge in the market.

Integration with Modern Technologies

Rearchitecting enables seamless integration with technologies like cloud computing and AI, providing real-time data analysis, predictive maintenance, and smart capabilities ready for Industry 4.0.

Cost Savings and Risk Mitigation

Rearchitecting is a cost-effective alternative to replacing software, minimising the risk of introducing bugs and errors. It also reduces the need for extensive user training and system revalidation.

Incremental Modernisation and Agile Adoption

Rearchitecting supports an incremental approach, minimising disruption and enabling companies to prioritise critical components. Agile methodology allows for quick responses to business needs and continuous improvement throughout the process.

Long-term Scalability and Flexibility

Legacy software systems often lack the scalability and flexibility needed to adapt to changing business requirements and future growth. Rearchitecting addresses these limitations by leveraging modern application architectures. These architectural patterns enable modular design, easier maintenance, and the ability to scale specific components independently. Companies can seamlessly integrate new technologies, extend functionalities, and adapt to future market demands, ensuring the longevity and relevance of the software solution.



Legacy Modernisation in action

We've worked with Stanley Black & Decker (SBD) for over 20 years and have recently completed a major rearchitecture project with them to deliver a Global Service Information System (GSIS).

This involved modernising their existing Bill of Materials (BOM) system and bringing their European, Middle East and African and USA operations together under one central system. The GSIS supports the collaboration and management of multiple projects for new product launches, automated parts in warehousing and new workflows for the Supply Chain team. It also provided the opportunity to integrate with several other key systems including their SAP ERP software for pricing and accounting of inventory.

The new web-based system provides a futureproof platform to build upon, increased speed and performance, enhanced security and delivered significant productivity benefits.

StanleyBlack&Decker





Crown
Commercial
Service



Legacy Modernisation in action

PDMS first developed the fleet portal twenty years ago. Based on spreadsheets and manual, office-based processes, the portal was originally set up from scratch as a pilot for the NHS Purchasing and Supply agency (PASA).

Now run by the UK's biggest procurement organisation, Crown Commercial Service (CCS), the car leasing portal allows customers across the UK public sector to compare car options and get the best possible deals through live competitive quoting across all suppliers and vehicle types.

There had been no significant upgrades or enhancements to the fleet portal for some time. Despite the functionality still proving valuable to their operations, the interface looked dated and more sophisticated users requested additional functionality for search, layout and reporting in line with recent technological developments.

To retain the core functionality while enhancing the portal's look and feel, we embarked on a refactoring project. PDMS provided a major enhancement of the platform, introducing new features and improvements including:

- Enhanced search functionality to include the most common criteria people search for and allow up to three vehicles to be selected and compared side by side
- More contemporary, refreshed website appearance with mobile responsiveness
- Improved functionality to feature special offers and news from CCS and suppliers
- A modern website front-end where CCS can maintain content
- Improved reporting and features to comply with GDPR requirements

Following the launch of the portal, users found it was easier to navigate with more vehicle information available to allow for improved comparisons. Within the first two years alone, the organisation's registrations quadrupled to circa 1600 site users, more than doubling CCS's initial target.

In addition, the new search and filter functionality helped realise some key sustainability benefits, with ultra-low emission vehicle (ULEV) registrations increasing from 7% to 30%, a figure which continues to rise as more ULEVs are available within the market.



But what about the drawbacks?

As with any major technology project there are implications and drawbacks to all of these approaches which need to be evaluated carefully against the specific circumstances.

Time and Effort

The process of replatforming, rearchitecting, or refactoring legacy software can be time-consuming and complex. It involves understanding the existing system, designing the desired architecture, implementing the changes, and thoroughly testing the new solution.

Technical Complexity

Legacy systems often have intricate dependencies, tightly coupled components, and outdated technologies. Replatforming, rearchitecting, or refactoring such systems can be technically challenging. It requires a deep understanding of the legacy codebase and may involve dealing with outdated or poorly documented code, making the process more difficult and prone to error.

Poor System Architecture

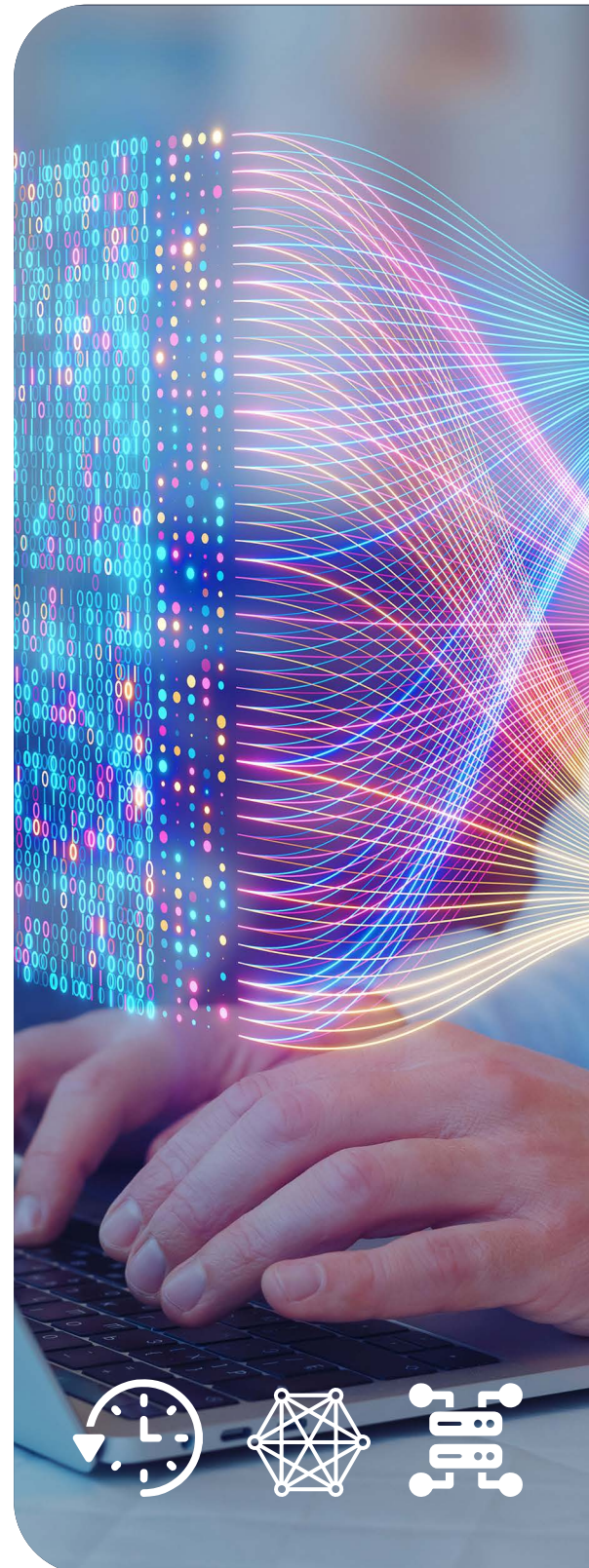
If the existing software has fundamental architectural flaws, such as serious scalability limitations, poor performance, or maintenance challenges, a rebuild or replacement strategy can provide an opportunity to start fresh with a well-designed and scalable architecture.

Encapsulation can sometimes provide a useful first step to help mitigate the technical complexity and system architecture challenges.

In our experience, when all the various factors are considered, a continuous modernisation approach is the preferred option compared to a 'rip-and-replace' approach.

“ Legacy applications often impede continuous delivery of value, but rip-and-replace efforts are costly, risky and time consuming. Applications and software engineering leaders should instead use continuous modernisation to minimise the cost, risk and impact of legacy-applications optimization.

Gartner - Use Continuous Modernisation to Optimize Legacy Applications - published 20 July 2022.



How do you decide which approach works best for your business?

So, you have a legacy software issue that needs addressing, the first step is to decide whether to rebuild or replace. This requires an assessment that takes into account various technical, business and strategic factors. The impact on end-users and stakeholders should also not be underestimated. If after evaluation, you decide that a more gradual and modular modernisation method would work best for your organisation – how do you then decide which technical route to go down?

This requires careful consideration of your organisation's unique needs, constraints and long-term goals. Factors such as budget, timeline, existing system complexity, and business priorities should guide the decision-making process. Depending on the level of technical expertise within your business, you may also need to seek advice from a technology partner who will look at both your business and technical challenges.

We've outlined some of the steps that we would typically undertake to help our customers decide on the best approach if they were looking to modernise a key legacy system such as their ERP.



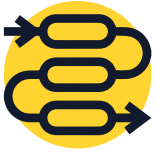
Understand the Business Objectives and Challenges

Begin by gaining a thorough understanding of the company's overall business objectives, pain points, and challenges. This includes identifying specific goals for the modernisation project, such as improving operational efficiency, increasing scalability, enhancing data visibility, or enabling faster innovation. We would typically conduct interviews and run workshops with key stakeholders and departments to gather insights into the current software limitations and the desired outcomes.



Assess the Existing Software Landscape

Perform a comprehensive assessment of the company's existing software landscape, including legacy systems, custom applications, and third-party solutions. Evaluate the functionality, performance, integration capabilities, security vulnerabilities, and scalability of each system. Identify the dependencies and interconnections between different software components to understand the potential impact of modernisation efforts.



Analyse Business Processes and Requirements

Analyse the company’s core business processes and specific software requirements. Identify critical workflows, data flows, and integration points across departments, such as inventory management, production planning, supply chain, quality control, and customer relationship management. Determine the specific functionalities and features needed in the modernised software to meet the company’s business requirements effectively.



Define the Modernisation Approach Options

Based on the analysis conducted in the previous steps, evaluate which of the approaches aligns with the company’s objectives and constraints.



Risk Assessment and Mitigation

Identify the risks associated with each approach such as potential data loss, system downtime, integration challenges or employee resistance. Assess the impact of these risks on the company’s operations and develop mitigation strategies. Consider factors such as the project timelines, budget, resource availability, and the company’s risk appetite.



Cost-Benefit Analysis

Perform a comprehensive cost-benefit analysis for each modernisation approach considering factors such as initial investment, ongoing maintenance, licensing costs, training, and potential productivity gains. Evaluate the return on investment (ROI) and assess the long-term benefits and strategic value of each approach in alignment with the company’s goals.



Present Recommendations and Roadmap

Based on the assessment, provide clear and actionable recommendations on the most suitable modernisation approach. This would include developing a detailed roadmap that outlines the implementation plan, milestones, resource requirements, estimated timelines and clear deliverables.





Navigating success

Choosing the right partner for your system modernisation journey

Bringing in external expertise at the evaluation stage can help you obtain an unbiased and objective assessment of your current software systems. They can identify areas for improvement and pinpoint inefficiencies that might be overlooked by an internal team that is accustomed to the status quo. You should benefit from fresh perspectives and ideas and insights from their experiences with other clients and industries – many of whom are facing the same challenges.

External input can also help speed up the process and make sure the process is efficient as possible, with the right people involved at the right time.

Choosing the right partner is a crucial decision that can significantly impact the success of your project. Here are some key factors to consider when evaluating potential technology partners:



Industry Expertise

Look for a partner with a deep understanding of modern technology trends, with a track record of successfully implementing software solutions.



Proven Track Record

Review the partner's portfolio and case studies. Have they successfully executed similar projects? Do they have references from satisfied clients? A proven track record indicates reliability and competence.



Technical Proficiency

Assess the partner's technical skills and capabilities. Are they proficient in the technologies and platforms relevant to your modernisation need?



Cultural Fit

A good cultural fit between your organisation and the technology partner is vital for a productive and enjoyable working relationship. Meet the people you will be working with (not just their sales person!) – the analysts and project managers as well as solution architects and developers.



Customisation and Flexibility

Business systems vary widely, and a one-size-fits-all approach may not work. A good partner should be willing and able to customise solutions to meet your specific requirements and workflows.



Scalability and Future-Readiness

Your modernisation efforts should provide long-term value. Ensure the partner can design scalable solutions that can accommodate future growth and technological advancements.



Data Security and Compliance

Business systems handle sensitive data. Your partner should have a strong focus on data security and compliance with relevant industry regulations.



Project Management

Effective project management is crucial to keep the project on track and within budget. The partner should have a structured approach to project delivery and project management with the ability to adapt to changes.



Innovation and Creativity

Modernising business systems often requires innovative thinking. Choose a partner that can bring creative solutions to the table, helping you leverage technology to drive operational excellence.



Collaboration

Ensure that your partner has factored in how they are going to involve the system's internal users, so they are bought into the project from the outset and don't feel like they are having a new way of working imposed on them. Experience in change management will help a smooth transition to new improved systems.



Gap Analysis

If you choose the 'replace' option, ensure that both you and your technology partner assess the gaps between what the new proposed system can do and what your requirements are before you sign on the dotted line. This is particularly important if it's an off the shelf ERP system.



About PDMS

With over 30 years of development experience behind us, we deliver successful outcomes for our clients through close collaboration, long term commitment and great software.


As software engineering specialists, our approach isn't based on selling you a product. Instead, we help you identify the best approach and solution for your business. Our experienced team work with a range of clients in a variety of sectors and we can apply our extensive knowledge to help ensure you get the most out of your investment in technology.

Our team would be happy to have an initial chat to find out more about your business and your technology changes.



Get in touch with Richard Graham at Richard.Graham@pdms.com or call us on 0141 880 1000



-  pdms.com
-  facebook.com/pdmsgroup
-  linkedin.com/company/pdms
-  GetInTouch@pdms.com
-  +44 1624 664000

Isle of Man Office
Global House
Isle of Man Business Park
Cooil Road
Isle of Man
IM2 2QZ
British Isles

Scotland Office
Unit 5000, Academy Park
Gower Street
Glasgow
Scotland
G51 1PR
United Kingdom