esynergy™

Application modernization:

What it is, why it matters, and how to get started

Index

Index	1
Introduction	2
What is application modernization?	3
Why does application modernization matter?	5
The benefits of application modernization	7
How to modernize your applications	9
How to prepare for application modernization	16
How to implement application modernization	20
Application modernization best practices	22
How to measure success and ROI	27
Key challenges and barriers	29
We are here to help	31

Introduction

Thanks to powerful cloud-native technologies like serverless and microservices, huge strides have been made in how applications are built and maintained.

As modern application development advances in leaps and bounds, pre-existing applications that have been built in more traditional ways start to create problems.

Firstly, performance deteriorates while the costs of maintaining them and the risk they represent increases.

Secondly, as they struggle to integrate with new technologies you can't leverage the benefits of your modern technology investments (e.g. cloud). You're stuck in the past, effectively!

This is why businesses set out to bring these applications in alignment with newer technology standards in an attempt to mitigate these costs.

But any enterprise application landscape is incredibly complex and there are difficult trade-offs and considerations to navigate in this process. And small differences in approach and strategy can have wildly different results.

In this playbook, we will outline everything you need to know to successfully approach a major application modernization project, helping to turn these pain points into business value.

This is intended for IT and business decision-makers who know they need to modernize but want to make sure that they maximize the business value of their app modernization program by taking the right approach.

What is application modernization?

Application modernization is the process of upgrading and transforming legacy systems and outdated application architectures, technologies, or processes to align with the latest technology standards.

The goal is to ensure that software remains agile, scalable, maintainable, secure, and responsive in the face of changing requirements and emerging technologies.

Key concepts

Modern application development revolves around numerous new technologies and approaches, forming the core of ongoing modernization initiatives.

Here are some of the most important concepts:

- Cloud
- Microservices
- Containerization
- Serverless
- DevOps

Cloud

One of the major facets of application modernization is moving legacy systems to cloud platforms, leveraging cloud-native features and managed services.

Microservices

Breaking down monolithic applications into smaller, independent services (‘microservices’), providing better scalability, maintainability, and flexibility.

Containerization

Using containers, like Docker, to package up an application with all of its dependencies and libraries, ensuring consistency across multiple environments for better portability.

Serverless

A cloud-native development model that allows developers to build and run applications without having to manage servers.

DevOps

Emphasizing collaboration between development and IT operations, and automating the application delivery process to improve release frequency and quality.

Why does application modernization matter?

Why would you want to modernize your applications?

The stand-out reason for modernizing an application is that not doing so leaves you unable to leverage the cloud, which is the biggest single enabler of enterprise innovation that exists.

The cloud is highly automatable, scalable, flexible, and resilient. It helps to accelerate time-to-market while reducing the cost and labor intensity of innovative development work.

The cloud is powerful because it enables you to abstract away decades of legacy technology and ways of working in one fell swoop. It lays the groundwork for a cultural shift towards modern ways of working based on automation, experimentation, and innovation.

Trying to rapidly innovate without the public cloud (and modern cloud-native applications) is like trying to fly a car to the moon. It's just not built for that kind of travel. You need a spaceship!

There is no way to be a competitive, modern enterprise business without fully leveraging the cloud and its capabilities.

By failing to modernize your applications and bring them in line with modern (i.e. cloud-native) standards, you are setting yourself up with a competitive disadvantage.

While this is the main issue, there are other important concerns, which can be divided into four main categories:

Operational and performance limitations:

- **Performance:**

Outdated architecture and inefficient code mean slower response times that lead to a poorer customer experience.

- **Scalability:**

As a business grows, on-premises legacy systems might struggle to scale with customer demand.

Security and compliance risk:

- **Security vulnerabilities:**

Older systems may not have the latest security measures, making them vulnerable to attacks.

- **Regulatory and compliance issues:**

Outdated systems might not be compliant (for example, adequate protection of personal data).

Economic challenges:

- **High maintenance costs:**

Working with outdated/unsupported programming languages or hardware, software licenses, and so on can result in maintenance costs skyrocketing over time.

- **Talent trouble:**

It can be harder to find the skills for older technologies and newer tech talent will want to work with newer tools and frameworks.

Integration challenges:

- **Lack of integration:**

Modern applications might have compatibility issues with other legacy systems, third-party applications, or specific hardware.

- **Inflexibility:**

Legacy systems might not be flexible enough to adapt to changing business requirements, making it hard for enterprises to pivot or innovate.

The benefits of application modernization

By modernizing your applications and bringing them in line with cloud-native ways of working you open the door to a range of benefits, both technical and commercially- orientated.

Here's a breakdown of each:

Technical benefits:

- **Optimization and efficiency**

With modern approaches like containers and serverless, infrastructure resources are used more efficiently.

- **Flexibility and future-readiness**

Modern architectures, especially microservices, allow for quick changes and deployments, making it easier to make new changes in response to shifting requirements and future-proofing your application.

- **Leverage the benefits of the cloud**

Apps can be scaled automatically up and down in response to demand, take advantage of managed services as well as the resilience of the cloud.

- **Integration**

Modern applications are typically easier to integrate with other systems and services using APIs and standardized protocols.

Commercial benefits:

- **Increase innovation**

With a modern foundation, organizations can more swiftly roll out new features, services, or products.

- **Improve customer experience and lower churn**

Faster, more responsive apps with better UI lead to higher customer satisfaction and reduce customer churn.

- **Attract tech talent**

Tech professionals generally prefer to work with modern tools and technologies, making it easier to attract and retain top talent.

- **Lower costs**

Reduced maintenance costs decrease the need for specialized legacy skills and make more efficient use of resources.

The ultimate benefit

The ultimate benefit of application modernization is that you get a chance to break away from legacy ways of operating: on-premises, waterfall projects, monolithic codebases, and so on. Replacing them with modern approaches that are much more conducive to rapid innovation and experimentation: cloud-native, agile product mindset, microservices, etc.

By just lifting and shifting your applications, you keep all of your existing legacy processes, bugs, vulnerabilities, and so on, along with the related operational costs and time-to-market delays.

And you miss out on the massive competitive advantage that comes with being able to rapidly iterate on your software in response to changing market demands and customer preferences.

How to modernize your applications

Application modernization is a very complex process that can involve hundreds of applications as well as countless different people, teams, departments, processes, tools, data, and so on from across your whole business.

These threads come together to form a vast tapestry that includes every corner of your business. And you can't 'modernize' one corner of this tapestry without considering the rest of it.

That's why this section sets out the following key topics:

- The different approaches to application modernization
- How to choose which approach is right for your business
- Best practices across people, processes, and technology

Approaches to application modernization

There are many different approaches you can take (and depths to which you take them) when it comes to modernizing your applications.

Each approach has advantages and disadvantages that must be balanced in light of your overall business objectives and unique context.

These approaches are often summarized under the '6 Rs', which are as follows:

1. Rehost (“lift and shift”):

Move applications to the cloud without any modification.

This is often the quickest way to migrate but may not deliver the most business value. For example, a monolithic legacy application can be hosted on the cloud but will not be able to scale up and down in response to customer demand. This is a ‘quick and dirty’ approach that gets the job done but adds very little business value.

Advantages

- Rapid approach
- Minimal disruption
- Minimal cost

Trade-offs:

- May not fully leverage cloud-native features
- Potential missed opportunities for optimization
- Legacy system issues might persist

2. Refactor:

Migrate applications to the cloud with some modifications, often to leverage cloud-native capabilities.

This involves restructuring certain portions of the application code to better fit the new environment. For example, reworking some sections into microservices or using APIs to improve communication between services.

Advantages

- Takes advantage of some cloud-native capabilities
- Balanced approach with moderate modifications

Trade-offs:

- Requires some re-engineering, which can be time-consuming
- Not as optimized as fully rearchitected solutions
- May involve some level of compromise between old and new architectures

3. Rearchitect:

Redesign the application from the ground up to fully leverage cloud-native features.

For example, fully transforming a monolithic app into microservices. This can be time-consuming but yields applications that are optimized for performance, resilience, and scalability in the cloud.

Advantages

- Fully optimized for the cloud, leading to better performance, scalability, and resilience
- Allows for a more future-proof design

Trade-offs:

- More resource-intensive and time-consuming
- Requires significant changes or a complete redesign
- This might necessitate retraining of staff

4. Rebuild:

Discard the current application and rebuild it from scratch using cloud-native technologies, e.g. serverless.

While this can be resource-intensive, it results in a fully optimized, modern application that can fully leverage the benefits of the public cloud.

Advantages

- Leverages the latest technologies and cloud-native features
- A clean slate allows for best practices in design from the outset

Trade-offs:

- High upfront cost, effort, and potential disruption
- Potential challenges in data migration

5. Replace:

Replace the legacy application entirely with a commercial Software-as-a-Service (SaaS) product.

This approach is suitable when off-the-shelf solutions meet business needs better than existing applications, for example, why build your own payments gateway when so many are available on the market?

Advantages

- Immediate modernization using established solutions
- Often comes with support and updates from vendors

Trade-offs:

- Potential loss of custom features unique to the original application
- Recurring costs for SaaS licenses
- Dependence on external vendors

6. Retire:

Identify and eliminate applications that are no longer useful or redundant.

This declutters the IT landscape and eliminates maintenance costs.

Advantages

- Reduces IT complexity
- Cost savings from eliminating unnecessary maintenance

Trade-offs:

- Potential loss of data or functionality
- Need to ensure that the retired application doesn't contain any critical business logic

Which approach should you take?

There are several key considerations to take into account when choosing between them, each of which has its trade-offs.

For example, the quickest and cheapest approaches generally give you the least in terms of business benefits and future readiness. While efforts to completely revamp your apps from the ground up can massively increase their performance, scalability, and user-friendliness, they represent a bigger investment of time and effort.

The question is how far do you want to go? Do you just want to revamp the front end? Or rebuild the architecture from the ground up? Do your applications need full cloud-native capabilities, or just become more scalable?

Generally speaking, the desired outcomes and objectives of a business, paired with its constraints and limitations, inform the most suitable approach to app modernization.

Below is a selection of the most common aspects that enterprise organizations weigh up against each other ahead of deciding on the best approach.

Business objectives:

When charting the course for modernization, aligning with your organization's long-term business objectives is crucial. Reflect on what your technology needs are to these goals. Ask yourself whether your applications need to be fully cloud-native or if just making them minimally cloud-compatible is adequate. By clearly understanding both your immediate and future aspirations, you can create a strategic roadmap. This approach not only helps in achieving early success but also ensures sustained progress and adaptability in your technological journey.

Time:

Prioritizing your requirements and aligning with your time and resource constraints will help ensure that value is delivered at every project stage. By engaging your stakeholders and understanding what your Minimal Viable Product (MVP) requires, you can set realistic expectations and align your development team for success.

Disruption:

Modernizing applications can be disruptive in the short term but offer significant long-term advantages. It's crucial to gauge your business's tolerance for disruption. Opting for a full-scale modernization might initially seem overwhelming but could lead to greater efficiencies. On the other hand, less comprehensive approaches might be less disruptive now but could require additional work in the future.

Cost:

When considering the budget for your modernization project, it's important to weigh the current maintenance costs of your existing apps against the potential savings from modernization. Think about the expenses related to downtime or poor performance and how modernization can reduce these costs. It's not just about the immediate expenses; consider the long-term financial benefits that modernizing your apps will bring.

Off-the-shelf software:

Before diving into custom development, consider if there's existing off-the-shelf software that meets most of your needs. Sometimes, tailoring a pre-existing solution can be a cost-effective and time-saving option. It's about finding the right balance between customization and leveraging what's already available.

Effort:

Assess the resources you have for this transformation. Do you have the necessary development team in place, or will you need third-party support or retraining? It's a balancing act between investing resources now and the ongoing effort needed to maintain legacy apps. Sometimes, the initial investment can free up resources and reduce long-term maintenance demands.

Security:

Legacy applications, especially those on the cloud, can pose significant security risks. Modernizing your applications can be a crucial step in reducing these vulnerabilities. Evaluate how much modernization can fortify your applications against potential security threats, ensuring your data and operations are safeguarded.

Future-readiness:

Think about your long-term goals and requirements. Are you planning to use the application for a short period or over many years? Fully modernizing your applications not only aligns them with current needs but also prepares them for future changes. This approach minimizes the need for frequent overhauls and can be more cost-effective in the long run.

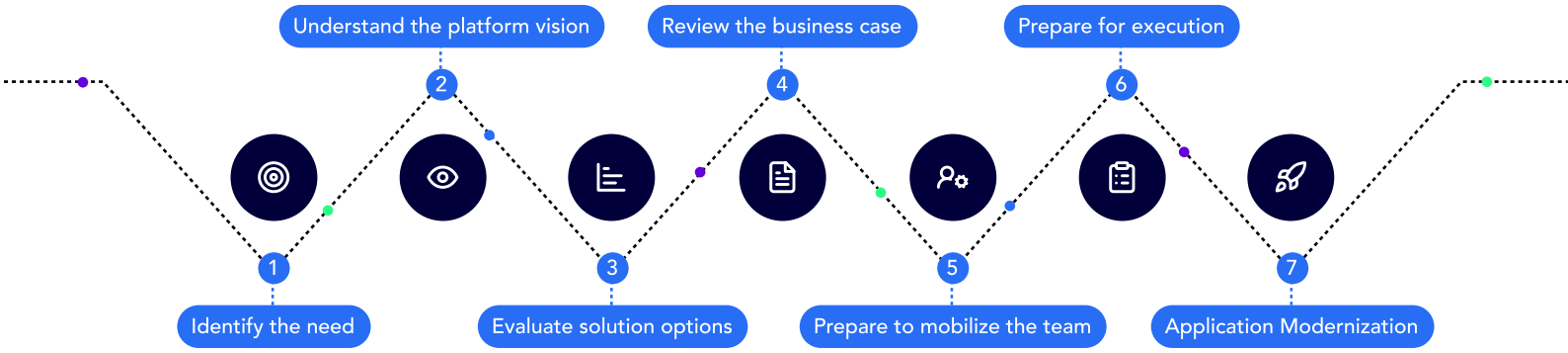
"In the realm of platform development, aligning closely with business needs is paramount. As an API platform product manager, I stress the importance of APIs as they speak the native digital language of transformation. They are the building blocks that allow us to seamlessly integrate and extend our platforms, bridging disparate systems and unlocking new capabilities. In this way, we ensure that our development efforts are not only technically sound but also deeply rooted in driving business efficiency and customer satisfaction, reflecting the true spirit of digital transformation."

David Holliday
Senior Product Manager APIs

Munich RE

How to prepare for application modernization

esynergy has created a seven-step process for application modernization. Following each step will take an organization from the initial stages of starting to grapple with the problem, all the way to executing their application modernization strategy and continuous improvement.



Stage 1: Identify the need

This stage is twofold, on the one hand, you must consider the needs of your customers, which can be both internal and external. On the other, you should consider your own business operations. This stage consists of identifying the needs and determining the requirements of your users.

This is the time to ask questions, run workshops, conduct research, user surveys, user interviews & market analysis to ensure that you fully understand your customer's needs and their 'jobs to be done'.

Secondly, you must assess your current business operations to identify inefficiencies and areas for technological improvement. Consider leveraging data analytics to obtain insights into operational challenges, and deliver assessments This is to get a clearer picture of the problem and the kinds of business outcomes that are desired.

Questions to ask:

- Who are your users?
- What are their pain points?
- What are your pain points?
- What are your business objectives?
- What changes in the market will impact your business?
- Is there an opportunity to improve your efficiency or reduce costs?
- What are your key operational in-efficiencies?

Stage 2: Understand the platform vision

This stage is to craft a vision for the new technology platform and get the rest of the business to back it.

Here the user's pain points are translated into concrete features and alignment is established on a clear end state for the platform. Features can be product features or related to the infrastructure. You should conduct focused workshops to define and refine your vision for your new technology platform. You should include a diverse stakeholder group to ensure that you receive comprehensive input.

Use methods like MoSCoW (Must have, Should have, Could have, Won't have) to prioritize features that align with user pain points and business objectives.

Questions to ask:

- Do you have an understanding of the scope/features and functionality of what you want?
- Do you understand the business value?
- Do you have stakeholder buy-in?
- What does good look like?
- What is the priority?

Stage 3: Evaluate solution options (buy vs build)

In this stage, the value, costs, and risks of different solutions are considered and managed.

In the context of platform integrations, this would be the model that you would choose such as using iPaaS, custom APIs, or a completely new platform.

Perform a detailed cost-benefit analysis of each of the solutions you review, including a view of long-term maintenance and scalability. Evaluate your current technology stacks and infrastructure to determine how compatible they are with new technology.

Questions to ask:

- What type of architecture are you looking at?
- Is there an existing solution that could address the needs of your platform?
- Do you understand your total cost of ownership?
- Will you be able to maintain this in the future?

Stage 4: Reviewing the business case

With strategic choices made, it's time to make a business case for the project to secure the required budget and development resources.

Develop an ROI (Return on investment) analysis for each selected solution to justify your business case. Identify the potential risks and challenges and how they can be mitigated.

Questions to ask:

- Have you got a resource and program plan?
- Have you got a prototype?
- Have you got a budget sign-off?
- Have you validated there is a need for these changes?
- How long will it take to generate a return on the investment?

Stage 5: Preparing to mobilize the team

In stage five, the teams are mobilized in preparation for execution.
You'll need clear roles and responsibilities as well as incentives and metrics that are aligned with business objectives. You'll need to know if your current team has the skills required for the project and identify any training needs. If you don't have the skills then you'll need a plan for hiring.

Finally, you'll need a change management plan to ensure a smooth transition and adoption of your new platform.

Questions to ask:

- Have you decided if you're delivering this internally (review internal capacity vs external resources)?
- Have you established project governance?
- Have you identified internal systems that will be integrated into the platform?
- Have you got the 'right' team?

Stage 6: Preparing for execution

In this stage, the final preparations are made for organizing team structure, allocating development resources, and aligning behind a methodology.

Start by choosing a methodology such as agile that aligns with the project scope and the dynamics of your team. Create a plan for your resources, who will work where, and what technology resources will be used.

Questions to ask:

- Have you decided on your ways of working?
- Have you identified your software engineering capabilities?
- Do you know what resources are available?

Stage 7: Application modernization

Launch the project! Start modernizing your applications, building continuous feedback into your approach, and integrating lessons learned over time.

Implement strong CI/CD (Continuous Integration / Continuous Deployment) practices to efficiently and reliably release code. Establish robust feedback loops which include talking regularly to customers.

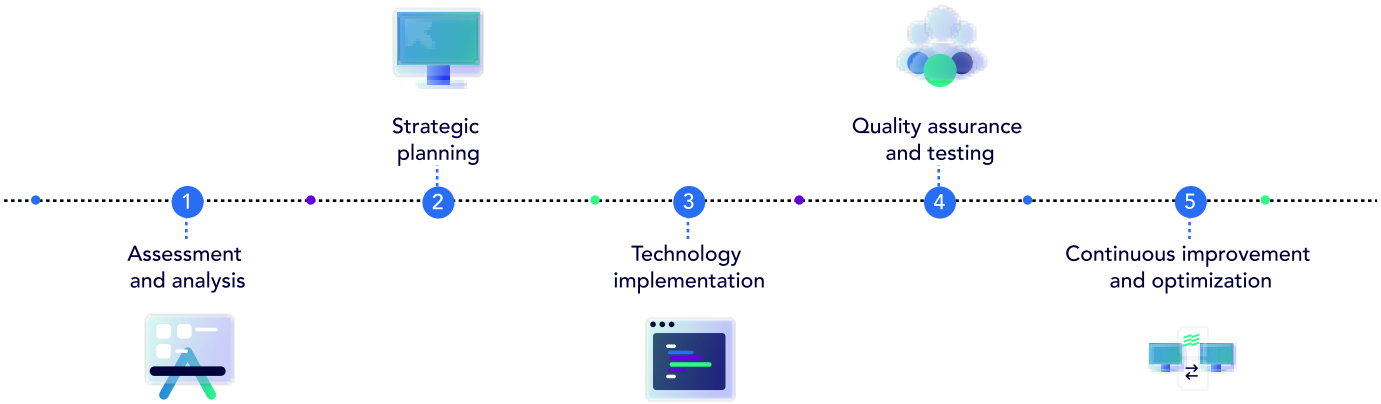
This project doesn’t have a discrete endpoint. You can keep evaluating your application landscape and adding new applications to the modernization queue indefinitely.

Questions to ask:

- How can we continually improve?
- How can we get user feedback?

How to implement application modernization

You've identified you need to undertake an app modernization project, but what's next? At esynergy, we have developed a comprehensive 5-step plan to guide organizations through the transformation of their legacy systems into modern, efficient, and scalable applications.



1: Assessment and analysis

Objective:

Evaluate the current state of your application portfolio.

Activities:

- Conduct a thorough audit of existing applications.
- Identify technical debts, dependencies, and potential risks.
- Analyze the business impact and usage of each application.
- Prioritize applications for modernization based on business value and technical feasibility.

2: Strategic planning

Objective:

Develop a tailored modernization strategy.

Activities:

- Define modernization objectives aligning with business goals.
- Choose the appropriate modernization approach (Re-hosting, Re-platforming, Refactoring, Rebuilding, or Replacing).
- Develop a roadmap with clear milestones and timelines.
- Consider compliance, security, and data migration needs.

3: Technology implementation

Objective:

Execute the modernization process.

Activities:

- Implement chosen modernization strategies for each application.
- Adopt cloud-native technologies and microservices architectures.
- Ensure integration with existing infrastructure and data systems.
- Apply DevOps practices for efficient deployment and continuous integration/ continuous deployment (CI/CD).

4: Quality assurance and testing

Objective:

Validate the modernized applications.

Activities:

- Conduct comprehensive testing (functional, performance, security).
- Ensure compatibility with existing systems and data.
- Gather user feedback for iterative improvements.
- Monitor performance and scalability.

5: Continuous improvement and optimization

Objective:

Validate the modernized applications.

Activities:

- Implement monitoring tools for ongoing performance tracking.
- Regularly review and update applications based on emerging technologies and business needs.
- Foster a culture of continuous feedback and iterative development.
- Ensure ongoing training and development for staff to keep up with modern technologies.

Application modernization best practices

Here are twelve best practices to help you approach this project in a way that will ensure that you take your people and the rest of your business with you on the journey.

They are categorized across people, processes, and technology.

People

Application modernization is a process of change management as much as it is rearchitecting code or building new platforms.

If you don't help your people understand what's happening, get them aligned to the project, and enable them to grow and develop along with your applications portfolio, then you are much more likely to fail!

1: Know your 'why'

Understanding why you're modernizing your applications is imperative for the project to succeed and has consequences for design and implementation decisions, such as timelines, roadmaps, and team structures.

For example, is your number one priority to drive greater efficiencies, increase digital innovation, or improve the customer experience?

How you approach this project will depend entirely on the answer to this question!

2: Stakeholder management

If the business and the key stakeholders aren't with you, your modernization project will grind to a halt. Ensure that you include stakeholders from the business at every stage of the transformation.

This can even include people outside your organization. For example, you will need to liaise with third-party vendors if you are using their software and there may be a substantial lead time that you need to factor into your project plan.

3: Communication is key

The balance of resources is complicated and needs to be carefully managed, which is the reason why constant communication is foundational.

Updating key stakeholders on a monthly or quarterly basis provides the business and service owners the ability to forecast when resources are required to modernize an application.

It gives them the ability to feedback on changes to the roadmap due to SME availability or other factors, such as technical requirement changes and modernization requirements.

4: Training

Modernizing applications will often involve upskilling the relevant teams that maintain or support those apps. For example, they might need skills in the area of containers, cloud-based managed services, or development methodologies such as DevOps or Agile.

You need to provide training and support to enable your teams to work with the new technologies, tools, and processes that you will be rolling out.

This includes considering how you support your teams that were supporting an old platform or now-retired application. Do you upskill them in the new platform/tooling? How do you properly communicate the changes that are coming to them?

Process

Old processes cannot be 'lifted and shifted' over to new technologies and platforms. They require new processes that are suited to their architecture and characteristics.

Here are some key best practices to help you work smoothly with your new, modern applications.

5: Modernize your operating model

If you change the architecture and/or re-platform your applications you will need to modernize your operating model to match.

Typically, this means a cloud-native operating model that caters to the various phases of cloud adoption and includes relevant business domains (e.g. finance, security, data, legal, etc.).

6: Work in iterative waves

You don't need to update everything at once. Work instead in iterative, organic waves.

In this way, each wave of app modernization can learn the lessons from the previous one to lower risk and maximize the chances of success.

7: Product—not a project—mindset

A project mindset tends to have a defined budget and lifecycle. Once a project is 'done' any future changes need to be planned, approved, allocated budget, etc.

A product mindset is more flexible, taking a longer-term, continuous improvement view. This kind of thinking is much more aligned with organic waves of improvement than the more rigid project approach.

8: Take an agile approach

You want to work in such a way that—should approval or budget be withdrawn—you aren't left hanging with hundreds of unusable half-upgraded applications.

Take a modular/agile approach that focuses on delivering small chunks of functionality so that applications are in a near-deployable state at all times. In this way whatever improvements you make can still deliver value if the project is curtailed or interrupted.

Technology

Many technology estates are a tangled web of legacy systems that have built up over decades. Ensuring that you properly understand and handle this tangled web—while keeping the business ticking—is difficult but imperative.

9: Develop a single source of truth

Enterprise businesses can have hundreds or even thousands of applications.

However, often there is no single source of truth for what applications exist, where they're hosted, who is maintaining them, what they cost, and so on. Before you can get started, you need to conduct a thorough assessment of your application landscape.

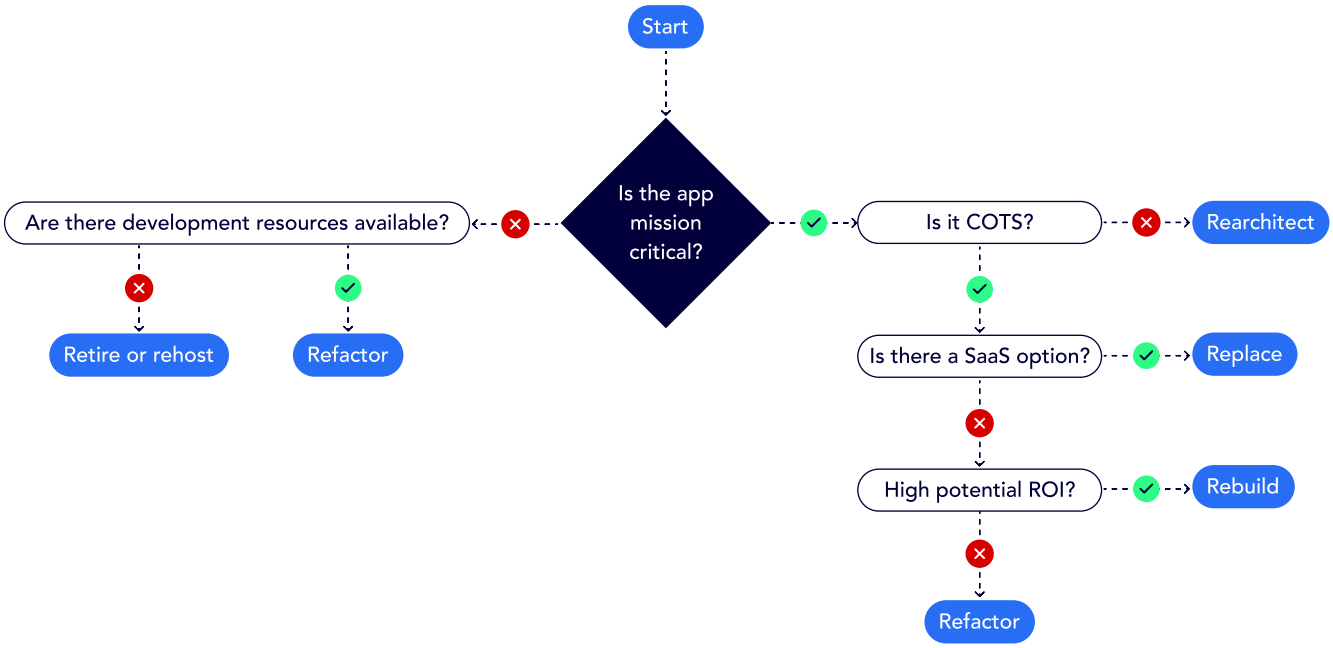
You can use a combination of configuration management databases, cloud provider records, and interviews with your teams and experts to end up with an accurate map of your current app landscape.

10: Have a system for prioritizing your applications

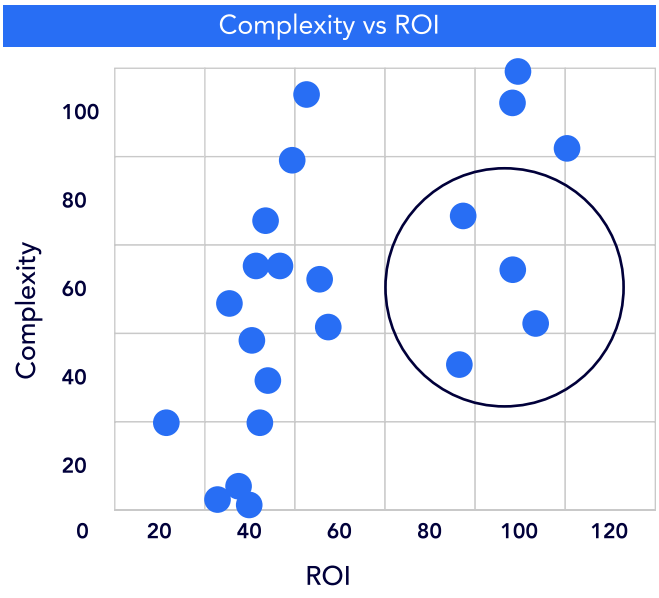
Build a decision matrix and/or flowchart that determines which approach (i.e. which of the 6Rs) you use for different applications based on your requirements, the state of the application as well as the capacities/structure of your people and processes.

You will need to develop a system for evaluating your applications based on your unique situation, risks, objectives, and so on.

A flowchart helps you to determine which approach to take based on the characteristics of the app and your business objectives.



Once you've established how to approach each application, map these on a matrix to help you prioritize the order in which you should approach them based on the complexity and the potential business value (ROI).



○ Mitigate these applications

11: Build a permanent application modernization team

This team will be responsible for mapping, evaluating, and tracking your application landscape. This will be an ongoing task that should produce a continuous backlog of applications to modernize or retire, allowing for continuous improvement.

Their task is to:

- Identify applications and details like ownership, stakeholders, related vendors, teams, etc.
- Record total cost of ownership
- Evaluate each application using your matrix/flowchart
- Define an approach for each application
- Put the app in the modernization roadmap
- Assign a team to execute
- Update the business with progress

12: Develop repeatable blueprints

To minimize duplication of work it can be helpful to develop infrastructure-as-code blueprints for things like development pipelines, containers, and microservices modules to be used across the organization.

By developing these repeatable assets early, you can save your teams a lot of time and effort and save them from reinventing the wheel.

"In the fast-paced world of tech platform development, the key to swift success lies in harnessing the power of experienced teams and deep customer involvement. By focusing our efforts on lean, agile practices and maintaining a relentless customer-centric approach, we can rapidly translate complex technical challenges into tangible business solutions. This streamlined synergy between expertise and customer insights ensures not just speed, but also precision in delivering impactful results."

Matt Lockyer

Matt Lockyer - Platform Practice Lead

esynergy™

How to measure success and ROI

How you measure success and track the ROI of your application modernization campaign will depend on your initial business goals and objectives.

In this section, however, we will give an overview of the five main areas where businesses typically develop key performance indicators (KPIs), along with example metrics.

1: Digital innovation

How fast can you develop and release new features to market? Modern cloud-native apps can leverage the cloud, which enables much faster release cycles.

- Time-to-market
- Deployment frequency
- Change failure rate

2: Financial benefits and resource utilization

Would the cost of maintaining your legacy applications have outweighed the upfront cost of modernization?

- Change in support and maintenance costs
- Resource efficiency (before and after modernization)
- Daily operations savings

3: User satisfaction

Are the modernized applications delivering a better user experience?

- User satisfaction
- User behavior
  - Bounce rates
  - Dwell time

4: Application performance

Is your application more available, performing better, and more resilient?

- Response time
- Availability
- Scalability

5: Security and compliance

Are your applications more secure and can vulnerabilities be remedied effectively?

- Security incidents over time
- Mean-time-to-recovery (MTTR)
- Cost per incident

"When building out platforms, especially service platforms, I've found it critical for success to balance out the technical capabilities, with a strong business value proposition and an efficient go-to-market engine to deliver return on investment. In today's modern world, platforms require understanding and investments from cross-functional teams, usually being led by an aspirational product and technology vision."

Ryan Clifford
Platform Group Product Manager

SPENDESK

Key challenges and barriers

Application modernization is a complex process, and enterprises often face a myriad of challenges and barriers during the journey.

Here's a list of some of the key challenges:

Technical challenges

- **Data migration:**

Modern applications are built completely around data; ensuring that they have access to the right data (and that it is migrated properly, if appropriate) is paramount.

Employing robust data migration tools and strategies can ensure seamless, secure, and accurate transfer of data to new systems.

- **Technical debt:**

Over time, systems accumulate patches, workarounds, and quick fixes, which can make modernization efforts more complicated.

A systematic approach to refactor and update systems incrementally, prioritizing areas that hinder modernization the most, can be implemented to lessen the impact of technical debt.

- **Skillset gaps:**

You may lack in-house expertise in modern technologies or methodologies required for the modernization process, which can be resolved either by calling on third parties or an internal training program.

Investing in comprehensive training programs and bridging skills gaps by partnering with technology experts can help to upskill and strengthen technical capabilities and skills.

Business challenges

- **Balancing speed and value:**

Finding the sweet spot between minimizing disruption and time-to-value versus ensuring you are delivering real value is difficult. Proceeding in iterative waves can help to adjust your approach over time.

Agile methodologies enable quick, iterative releases, allowing for regular assessment and adjustment to align with business value.

- **Business disruption:**

Critical applications need to be continuously available, and the prospect of downtime during modernization can be a significant barrier to major changes.

Careful planning for phased rollouts, and scheduling key activities during off-peak periods can minimize business disruption during upgrades and enhancements. With redundancy built into the technical architecture, and well-defined processes for identifying and pre-empting potential outages, the availability of systems can be best in class.

- **Resistance to change:**

Concerns over their new role and resistance to new ways of working are common, and proper communication and training are necessary.

Fostering a culture of change through continuous engagement, clear communication, and training programs will ease the transition and uncertainty for employees and stakeholders.

- **Regulation and compliance:**

Enterprises in regulated industries (e.g. finance, healthcare) need to ensure that any changes do not breach compliance mandates.

All modernization plans should be reviewed from a compliance and regulatory perspective at all stages of the design and delivery of the modernization project. The core team should include key members who can bring requirements and insights to shape the product.

We are here to help

At esynergy, our approach to application modernization is designed to be comprehensive, yet flexible, allowing for customization based on specific organizational needs. We aim to ensure that your modernized applications are not just compliant with current standards but are also scalable, efficient, and prepared to meet future challenges and opportunities.

esynergy has helped many organizations transform their applications to achieve better performance, scalability, and resilience, view our application modernization case studies:

- CPS modernizes Case Management System
- Leading fintech company integrates payments system
- Clúid Housing's digital transformation

If you are struggling with any of the key challenges of application modernization:

- Making a business case
- Winning over key stakeholders
- Deciding what approach to take
- Determining the most appropriate tools and technologies
- Measuring success and key metrics
- Continuous improvements

Get in touch with our Head of Platforms and App Modernisation:

Matt Lockyer
matt.lockyer@esynergy.co.uk

esynergy™

New London House, 6 London Street, London, EC3R 7LP

info@esynergy.co.uk

0207 444 4080