esynergy™

New platform build

What it is, why it matters, how to do it

Index

Introduction

Over the years and decades, enterprise businesses accrued layer after layer of tools, technologies, and processes. These are often incorporated with little thought as to how they will function together over the long term. As the layers accumulate, they become less harmonious, with different systems struggling to integrate.

Workarounds and quick hacks are then deployed that keep the business going in the short term but just make the legacy problem worse over the long term. At the same time, new technologies are emerging (e.g. cloud-native, machine learning, artificial intelligence, advanced data science) that possess incredible potential to open up new streams of business value and revenue.

Crucially, there comes a point where the disconnect and separation between different legacy systems starts to put limits on a firm's ability to leverage these new technologies. They simply don't easily integrate with complex legacy systems, siloed data, and inflexible infrastructure.

To break out of this legacy prison and wipe their technology slate clean, businesses often consider building an entirely new technology platform for their business.This has the potential to abstract away decades of legacy and offer incredible technological freedom but is also a complex and costly endeavor, with many challenges and pitfalls.

This playbook is an introduction to the topic of new platform builds. It is intended for those IT and business decision-makers who know they need to overcome the limitations of their legacy systems to unleash new business value and are wondering whether and how building an entirely new platform might help them meet that goal.

New platform build: a shared definition

A platform is a foundation for building and running business applications. It allows users to deploy and leverage software without having to worry about the underlying technology, which is abstracted away.

A platform typically consists of integrated access to core components such as infrastructure, databases, application development frameworks, UI/UX tools, API tooling, security (including identity and access management), development tools (DevOps, automation, testing, monitoring, analytics, etc.) as well as collaboration and project management tooling.

A new platform build refers to building such a software platform from scratch, to overcome the limitations of existing legacy systems and open the door to new strategic capabilities. This stands in contrast to other approaches to working with legacy systems such as integrating them (platform integration playbook) or modernizing them (application modernization playbook).

Here are some practical use cases that demonstrate the different ways in which businesses can deploy new platforms.

Digital banking platform

Developing a modern digital banking platform that allows customers to perform banking activities online, including account management, fund transfers, bill payments, and mobile check deposits.

It can also integrate features like personal financial management tools and digital wallets.

Payment processing platform

Creating a secure and efficient payment processing platform for online and mobile transactions.

This platform may support various payment methods, including credit cards, mobile payments, and cryptocurrency.

Wealth management and investment platform

Building a platform that provides wealth management services, including portfolio management, financial planning, and investment advisory.

Citizen services portal

Developing a centralized online portal that offers citizens easy access to a wide range of government services, including tax filing, permit applications, public records requests, and social services applications.

Process automation

Streamline customer requests through a single portal where your customers have a single account and can access all of your services digitally.

Digital identity and authentication platform

Building a platform that provides wealth management services, including portfolio management, financial planning, and investment advisory.

Insurance policy management system

Developing a platform that enables insurance companies to manage policies, claims, and customer information more efficiently.

This can include automation of underwriting, claims processing, and policy administration.

What is platform thinking?

At the highest level, building a new platform is not just limited to creating a place to run new applications and build separate self-contained products.

True platform thinking goes a step further. It's about connecting all the different parts of your business—channels, data, tools, people—to create an adaptable ecosystem of business capabilities and building blocks, on which products are built and through which they are connected.

It's like a stable center of gravity for your technology stack.

This kind of deeply interconnected, flexible, modular technology platform helps to create innovative customer experiences and bring them to production faster and more frequently.

Importantly, you can also harmonize your existing applications and data, while leaving room to add or subtract new applications, tools, and data as your business grows and changes, without having to start from scratch, integrate a bunch of complex systems, or bother with messy migrations.

The two kinds of platform

Broadly speaking there are two kinds of platforms: engineering and service.

Engineering platforms:

These provide developers and engineers with everything they need to build further platforms and products, including:

- **Cloud resources:**

Cloud infrastructure, environments, and managed services (including automated onboarding)

- **Developer resources:**

Development frameworks, component building blocks, documentation, etc.

- **Developer tools:**

Integrated development environments, version control, CI/CD pipelines, tools for automated testing, security, monitoring, collaboration, etc.

Note: this can include low- or no code capabilities for faster application development and to open product-building possibilities to so-called 'citizen developers' (i.e. business users who leverage pre-approved tools to build application capabilities without the need for code).

A high-quality engineering platform is an accelerator (a meta-platform, in a sense) for the production of service platforms (see to the right).

Service platforms:

A service platform is designed for customers (either internal or external) to fulfill a particular use case, emphasizing ease of use and service delivery.

Internal examples would include document management systems or customer data platforms, external examples would include payment processing platforms or customer portals.

They typically include components like:

- **Application services**

Specific functionalities that deliver the core services to users

- **User interfaces**

Intuitive user-friendly interfaces.

- **User resources**

Account management, support, and help desks.

- **Business tooling**

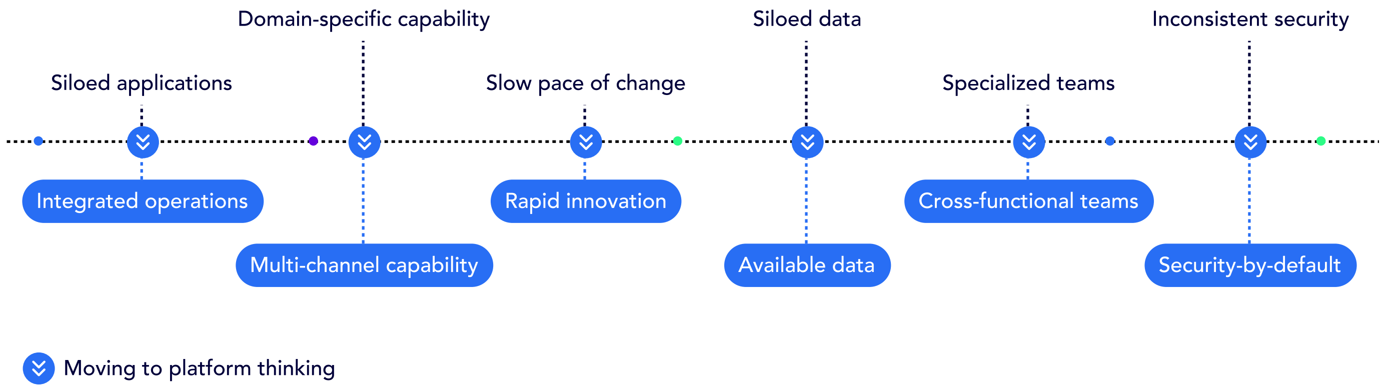
Analytics, reporting, notification systems etc.

Key concepts of platform thinking

The reason platforms are so powerful is that they provide a stable developer ecosystem to create new products and services at speed and scale as well as add additional value by bringing together different business elements (teams, data, processes, etc.).

This has a profound effect on the structure of your business, your processes, and your people.

Here are some key differences that emerge when you move towards platform thinking:



Integrated operations

The narrow focus on individual applications is left behind. You can seamlessly integrate applications and data from across your business by leveraging APIs.

Multi-channel capability

By integrating disparate apps, tools, and data, you harmonize domain-specific channels into a seamless multi-channel capability.

Rapid innovation

The developer ecosystem makes frequent releases (and roll-back!) of small chunks of code straightforward, enabling rapid experimentation and innovation.

Available data

By connecting your entire business (or at least parts) via your platform, data no longer needs to be centralized, which is a bottleneck. Instead, valuable data products can be created anywhere and consumed from anywhere.

Cross-functional teams

Rather than teams operating in specialized verticals, cross-functional teams can take an idea to production without handovers.

Security-by-default

Common platform security standards can be automatically embedded by default into your platform infrastructure as well as the code that is deployed on it.

Why new platforms matter

A new platform is a valuable opportunity to break free from years of legacy technology, process, thinking, and culture.

Done well, it offers a generational leap ahead for your business in terms of innovation capability. Rather than struggling to integrate, modernize, and balance a tangle of systems built at different times, to different standards, and with different technologies, you can abstract away the whole thing and start afresh.

You get an almost complete reset on the organizational scar tissue that has built up around your technology over the years.

And this freedom from legacy systems is fertile ground for the growth of genuine digital innovation.

This freedom creates space, for example, for advanced data science and analytics, machine learning and artificial intelligence, building a 360° view of your customer, building powerful business-wide reporting capabilities, and so on.

And, even better, this innovation isn't limited to just the technologists. Low- and no-code platforms make it possible for non-specialist business users to run analytics, build products, etc. By democratizing technology capabilities through your business in this way, you open up the power of your technology to business experts who can utilize their knowledge of the business and the power of the technology to be truly innovative.

When should you build a new platform?

There are other useful approaches for overcoming the limitations of legacy platforms. Which one is right for your business depends on your circumstances and needs. So what are the business problems and pain points that make building a new platform from scratch the best option?

Firstly, when current systems are unable to keep up with emerging business requirements for scalability, growth, cost-efficiency, and integration with new technologies.

Secondly, when modernizing or integrating existing systems to achieve this is too complex, too costly, or simply not possible.

Here are some of the main business problems and pains that are compelling reasons to embark on a new platform build:

Limits of legacy technologies

- **Outdated legacy systems:**

Existing systems are unable to keep up with business requirements and modernizing them will be too complex and/or costly.

- **Integration challenges:**

Existing systems are siloed and disconnected but integrating them will prove too complex, and costly or will result in a platform that is still unable to meet modern demands, e.g. AI and ML.

Leveraging innovation

- **Not embracing innovation:**

New technologies like AI, IoT, and big data analytics require flexible, efficient, and data-driven platforms.

- **Incorporate modern features**

Features like mobile access, real-time analytics, or cloud-based services that existing legacy systems cannot support.

Growth and strategy misalignment

- **Rapid business growth:**

If existing systems are rigid and difficult to adapt, exponential scaling requirements can necessitate the building of a new platform with a much more scalable foundation.

- **Change in business strategy/direction**

New platforms are needed to support new business strategies, such as targeting new markets or changing core offerings.

High costs and lower revenues

- **Maintenance costs:**

When the cost of maintaining existing systems becomes prohibitive due to inefficiencies, downtime, or expensive licenses, building a new, more efficient platform might be more cost-effective in the long run.

- **User churn:**

If existing systems provide a poor user experience this can be costly in terms of user retention and, ultimately, revenue.

The benefits of a new platform build

When your existing systems are proving a major bottleneck to achieving your business goals and strategic objectives (and integration/modernization will prove too complex and/or expensive), building a new platform can represent a massive leap forward for the business.

The biggest benefit is the chance to bypass the complex spaghetti monster of legacy systems and start fresh.

From here there are a variety of technical and commercial benefits that cascade downwards:

Technical benefits

- **Customize anything:**

New platforms in enterprise businesses can customize anything to precisely meet business needs, offering significant benefits in terms of overcoming legacy system limitations and gaining a competitive edge.

- **Streamline operational efficiency:**

Reduce the cost of siloed, separate systems (maintenance, delays, handovers, etc.) and improve efficiency.

- **Scale at ease:**

Enable increased traffic volumes without complex restructuring or quick fixes.

- **Democratize critical technology capabilities:**

Empower commercial SMEs to use technology platforms to build new products, run analytics, etc.

- **Accelerate delivery:**

Get to market faster, without compromising on security, cost, or quality.

Commercial benefits

- **Generate new revenue streams:**

Open the door to strategic new customer experiences and products that can create new sources of revenue.

- **Accelerate experimentation & innovation:**

New ideas can be more rapidly developed, tested, brought to market, and iterated.

- **Gain competitive advantage:**

Differentiate your business through innovative features and services that were previously impossible.

- **Align your strategy:**

Your technology platform mirrors the strategic goals of your business, helping to achieve them more effectively.

- **Improve customer experience:**

A single platform for customer data, touchpoints, and interfaces makes for a vastly superior customer experience that can be highly personalized and responsive to their needs.

How to successfully build a new platform

For all the benefits of building a new platform, it also requires careful planning, considerable investment as well as the right skills to manage the inherent risks and complexities.

In this section, we will cover a few core issues:

- Should you build or buy?
- 5 principles of platform thinking

Framework: build or buy

The biggest decision to make when considering building a new platform is whether you build it yourself or buy off-the-shelf and customize to better meet your needs.

The key factors to take into account when making your decision are the following:

- Customizability
- Total cost of ownership
- Time-to-market
- Skills

Customizability:

How tailored the platform is to your exact business requirements (now and in the future)? Do you need an all-singing-all-dancing platform to meet extremely specific technical use cases? Or will something generic fit the bill perfectly?

Build:

You should build your own platform when off-the-shelf products would have to be fundamentally altered to meet your needs.

Buy:

If you can get 80% of the functionality you need from off-the-shelf, then do so and tweak it to your needs as much as time/budget restrictions allow.

- How specific and niche are your requirements?
- What future use cases do you envision?
- How much would you need to customize a COTS product to meet your needs?

Total cost of ownership:

What is the total cost of building/buying and then maintaining the platform over time, versus the business value that is created?

Build:

Tends to be much more expensive upfront but has the potential to deliver greater business value down the line by opening new strategic possibilities and delivering a competitive advantage.

Buy:

Tends to be more affordable in the short-term, enabling quick wins but offering less flexibility over the long term (and those licensing costs can rack up quickly…!)

- How much budget is available?
- What is the projected ROI of a more bespoke platform?
- What are the maintenance/license costs associated?

Time-to-market

How quickly can you deploy the new platform? Depending on your business circumstances there may be more or less urgency to deliver value, versus invest in a more flexible, longer-term solution.

Generally, the longer it takes to build, the more customized and bespoke it will be to your requirements. Where the sweet spot lies in terms of this trade-off will vary from business to business.

Build:
Building your own platform from scratch is a bigger investment time-wise, but offers greater depth and flexibility.

Buy:
This is a much quicker option (particularly if you use it straight out of the box with no tweaks) but is less flexible.

- How fast do we need to move?
- Are there particular product roadmap or transformation deadlines we need to hit?
- How flexible do we need the platform to be?

Skills

What does your existing pool of skills (including resources available to get third-party support) enable you to do? And how available are these resources for the project?

Build:
This has the highest skills outlay, requiring high-quality, cloud-native development skills (including agile/DevOps ways of working).

Buy:
Lower barrier to entry in terms of skills, but does require maintenance over time as well as development skills for any customization that is required.

- Do you have the requisite development skills?
- How stretched are those resources?
- What third-party support is available?

6 principles of platform thinking

Once you've decided on the approach you want to take, how do you go about making it a reality? Platform thinking involves moving from vertical, siloed, separate structures and processes towards horizontal, integrated, connected structures and processes that facilitate end-to-end accountability and ownership.

Here are 6 core principles that will lay a successful foundation for your platform project.

- Focus on business value (i.e. use cases)
- Build the engineering foundations first
- Prioritize the user experience
- Build an MVP and iterate
- Leave slack for adaptation and learning
- Treat your platform like a product

1: Focus on business value (i.e. use cases)

Ensure that every aspect of your platform can be linked back to the creation of business value and fulfillment of business objectives in a way that is specific and measurable. You need to first consider what use cases you want to enable, determine the capabilities required, and then start building out the technology from there.

Building an all-singing-all-dancing platform before you think about use cases is an effective way to waste a lot of time and effort on technology capabilities that won't be fully utilized.

2: Build the engineering foundations first

The foundation has to be a core engineering platform. This is the key enabler for the effective development of further service platforms and/or products. This means creating an ecosystem of developer capabilities and building blocks that make it as easy as possible for developers to innovate.

If you get this right, you can develop any number of further platforms, products, and services much faster (and to higher standards) than would otherwise have been the case.

A modular design means that you can add new capabilities so the platform can continuously evolve. These building blocks include things like:

- Cloud services and infrastructure
- Identity management
- Version control and source code management
- Shared developer resources (documentation, code libraries, Terraform modules, etc.)
- Automation (CI/CD, testing, orchestration)
- API development and management tools
- Data accessibility and availability
- Collaboration and project management tools

3: Prioritize the user experience

Resist the temptation to put the technology at the heart of your platform project; instead, put the user front and center and let the right tech choices emerge from there.

The user interface and user experience are foundational to the effectiveness of your platform. Users must be onboarded very smoothly and be able to get what they need quickly and intuitively.

This will necessitate user and requirements research upfront. Don't be shy here, put as much effort into understanding your users as you do into thinking about the technology (if not more!). The investment will pay for itself many times over.

Don't make assumptions about how your users (engineers or business users) interact with the platform. Just because a piece of functionality exists doesn't mean that it's presented in a way that makes sense to people and that they can use.

4: Build an MVP and iterate

Paradoxically, although the vision for a core platform is a large one, the best way to proceed is to deliver it in small chunks of functionality (i.e. as a minimum viable product [MVP]). Create first the most basic functionality that you need to start creating value. And then iterate from there based on user feedback and responsiveness to changing business requirements.

Resist the temptation of the ‘big bang’, trying to do everything at once upfront. This is too inflexible to changing requirements, too distant from user feedback and so complex to deliver that it often collapses under its weight.

5: Leave slack for adaptation and learning

When your plan is too detailed and overwrought and too focused on the outcome, there’s no room left to be flexible and adaptable in response to feedback from the business or the environment.

In any large IT project, you aren’t going to be able to predict everything upfront. There needs to be slack in the system so there’s space to gather feedback, learn, and adapt.

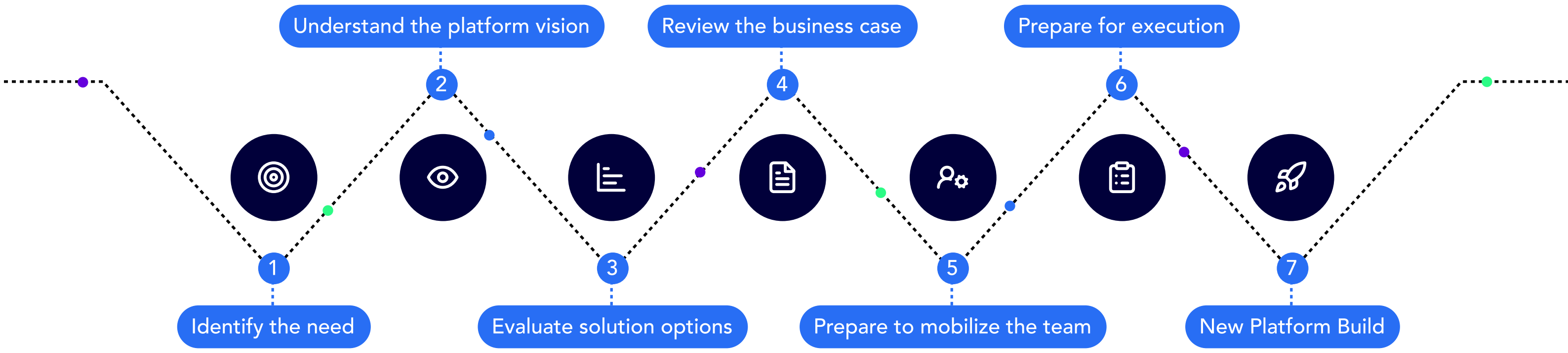
6: Treat your platform like a product

The best way of ensuring that you continue evolving your platform ecosystem is to treat it as a product and your employees, board members, leadership teams, etc. as the customers.

Identify the customers, what their needs are, and how the features of the operating model can meet those needs based on metrics that can be linked back to overall business goals.

Define a product owner for your operating model who builds relationships with the other parts of your business, manages requirements, and oversees the platform roadmap and rollout.

How to prepare for a new platform build



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?

"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

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: New Platform Build

Launch the platform, 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.

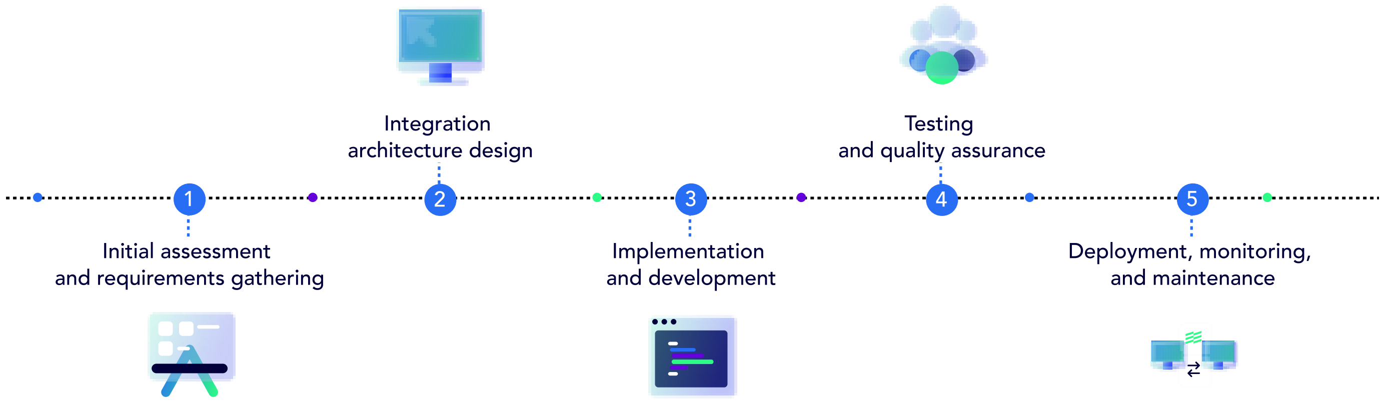
This project doesn’t have a discrete endpoint. You can keep evaluating your application landscape and adding new applications to the modernization queue indefinitely.

Questions to ask:

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

How to implement a new platform build

In the era of digital transformation, deploying new platforms is a critical aspect for any organization seeking to streamline operations, enhance efficiency, and drive innovation. esynergy's strategic approach to implementing a new platform build involves a comprehensive 5-step plan that ensures seamless integration of various systems and platforms, aligning them with your business objectives.



1 Initial assessment and requirements gathering

2 Integration architecture design

3 Implementation and development

4 Testing and quality assurance

5 Deployment, monitoring, and maintenance

1: Initial assessment and requirements gathering

Objective:

To evaluate the existing IT infrastructure and gather detailed requirements.

Activities:

- Conduct an audit of existing platforms, systems, and data sources.
- Identify business processes that will be impacted by the integration.
- Gather technical and business requirements from stakeholders.
- Establish clear integration goals and success metrics.

2: Integration architecture design

Objective:

To design an optimal integration architecture that meets business needs.

Activities:

- Develop an integration architecture that supports scalability, security, and performance.
- Select the appropriate integration tools and platforms.
- Define data flow and integration points between different systems.
- Ensure compliance with data privacy and security standards.

3: Implementation and development

Objective:

To develop and implement the integration solution.

Activities:

- Develop custom integrations or configure existing integration platforms.
- Integrate disparate systems, databases, and applications.
- Implement API-led connectivity for seamless data exchange.
- Test integrations in a controlled environment before going live.

4: Testing and quality assurance

Objective:

To ensure the integration solution works as intended.

Activities:

- Conduct thorough testing including functional, performance, and security tests.
- Validate data integrity and consistency across integrated platforms.
- Resolve any issues or bugs identified during testing.
- Involve end-users in user acceptance testing (UAT).

5: Deployment, monitoring, and maintenance

Objective:

To deploy the integration solution and ensure its ongoing effectiveness.

Activities:

- Deploy the integration solution in a live environment.
- Monitor the integration for performance and issues.
- Provide ongoing maintenance and support.
- Continuously optimize and update the integration as needed.

Best practices for platform thinking

How do you actually enact the above principles? They need to be mirrored in the structure of your teams and processes.

Here are 5 key best practices to bear in mind when seeking to embody the principles of platform thinking.

1: Adopt a business-wide operating model

A new operating model is required to move your business from siloed, vertical operations to integrated, horizontal networks that mirror the structure of your technology platform.

One that can enable a continuously evolving, business-wide ecosystem of capabilities and that goes beyond the IT department to include other relevant business teams: compliance teams, legal teams, business operations teams, data teams, etc.

These horizontal cross-cutting functions need to be included equally as peers alongside IT in the operating model.

Shift left the consultation with these supporting functions before you start the initial build. The operating model should enable continuous collaboration between IT and non-IT supporting functions at each key stage of delivery thereafter.

2: Create cross-functional teams

Horizontally organize your people, rather than vertically.

Rather than individual functions looking after their stage of the delivery cycle (e.g. product, development, etc.), which results in multiple handovers, miscommunications, and delays, you want to restructure your people into independent, cross-functional agile teams that combine business, engineering and product expertise with end-to-end accountability.

3: Adopt DevOps/Agile ways of working

The purpose of a powerful platform is to be able to deliver new features rapidly. Accordingly, your teams need to adopt ways of working that emphasize the continuous delivery of small chunks of functional code at speed, rather than slower delivery of big pieces of software.

DevOps entails building a collaborative and delivery-focused culture, highly automated delivery pipelines, and cross-functional teams of developers and operations engineers.

4: Democratize your data

Data is the lifeblood of the modern customer experience and effective digital innovation.

A major benefit of building a new platform would be to streamline your dataflows, reducing the number of data sources and making it higher-quality and more accessible to your product teams.

You will need to develop a data architecture, governance framework, and processes that enable the democratization of your data, i.e. the free flow of high-quality, trustworthy, and highly available datasets across the business.

A federated approach consisting of a decentralized data mesh alongside centralized governance and standards, for example, would be one way of doing this.

5: Make careful technology choices

You don't want to invest everything in a flexible, scalable platform, only to be constrained and boxed in by rigid, inflexible technology choices.

The technology choices should come late in the game, emerging as a result of your overarching business objectives and the user experience.

The most important consideration is whether or not a piece of tech can do what you need it to. But beyond that there are other important considerations against which this must be balanced:

- How easily can you hire or acquire the skills needed to maintain the tech?
- Will you get locked into a particular vendor?
- Will it leave you reliant on a small number of people?
- What is the long-term total cost of ownership?
- How flexibly does this technology integrate with other systems?

"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 platform integration program 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: Operational efficiency and agility

How has the integration impacted your business's operational efficiency and agility?

- Time-to-market
- Deployment frequency of integrated systems
- Reduction in manual processes

2: Costs and resource utilization

Did the cost savings and improved utilization from the integration outweigh the initial investment?

- Change in support and maintenance costs post-integration
- Resource efficiency
- Operational cost savings

3: User and customer satisfaction

Has the platform integration improved the experience for users and customers?

- User and customer satisfaction ratings
- User and customer engagement metrics
- Reduction in customer service issues

4: System performance

How has the integration affected the performance and reliability of your systems?

- System response times
- System availability and uptime
- Scalability improvements

5: Data and technology democratization

How has the integration improved your people's access to key data, tools, and technology?

- Data quality and accessibility
- Tool adoption rates
- Reduction in customer service issues

> *"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

At esynergy, we've seen many businesses attempt to build new core platforms, but few end up being as successful as they envisioned.

The following are some of the major barriers that we see businesses struggling with:

Winning hearts and minds

Your people need to be convinced that it's worth their time to change from old, comfortable to new, less-familiar ways of working. Unless you win the hearts and minds of your people you will struggle to move the needle.

Foster a culture of change through clear communication of benefits, involving employees in the process, and recognizing their efforts in embracing new technologies.

Making a strong business case

To win support, you need to make a powerful business case for change. Try a lighthouse project to demonstrate the value of platform approaches and get a mandate to further investment.

Building the technology capabilities

Platform approaches demand substantial technological sophistication that many firms lack: mature cloud infrastructure, very high levels of automation, democratized data, excellent data/security governance, and so on. These are the heart of the modular platform architecture that enables rapid code releases at scale.

Gradually build or upgrade to a robust cloud infrastructure, automate processes, and implement strong data and security governance to support a modular platform architecture.

Lack of skills

These technology capabilities require in-demand skill sets that can be difficult to find. Businesses need to invest in a powerful employee value proposition to attract talent and/or find partners that can fill some of those skill gaps.

Develop an attractive employee value proposition and training programs to attract and upskill talent, or partner with external experts to bridge skill gaps.

Security, regulation, and compliance

Ensuring your new platform is secure by default and in line with the relevant legal regulations is a major challenge. You need to be able to determine what standards are required, translate them into technical language, and embed them in every stage of platform development.

Engage key stakeholders within the design and delivery of the platform, to ensure that regulatory and compliance requirements are understood and incorporated, with security embedded at every stage.

We are here to help

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

We've helped many organizations integrate their platforms to achieve better performance, scalability, and resilience, view our platform integration case studies:

- HMRC builds new cloud PaaS
- Taxually's state-of-the-art architecture
- Mettle builds a challenger bank
- Building a wealth platform to revolutionize platform analytics

If you are struggling with any of the key challenges of a new platform build:

- 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 New Platform Builds:

Matt Lockyer

matt.lockyer@esynergy.co.uk

esynergy™

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

info@esynergy.co.uk

0207 444 4080