Rocket software

API-first Driven ERP Transformation

A continuous modernization, business-critical imperative

Contents

Executive Summary

In the digital age, where agility, efficiency, and innovation are paramount, leveraging application programming interfaces (APIs) to transform Enterprise Resource Planning (ERP) systems has become a business-critical imperative. In addition, while the ERP market has evolved, platforms such as IBM® i and IBM zSystems® continue to be prominent in supporting ERP systems due to their stability, integration capabilities, scalability, security, legacy system support, and strong partner ecosystems. However, monolithic applications, while reliable, can become barriers to innovation and hinder cloud-centric business transformation.

As with any service or software, APIs greatly enhance a business's use of ERP software. APIs ensure that other applications and systems can easily access the ERP as an intermediary with any third parties, such as users requesting information. To stay competitive by way of modern cloud and edge software deployments, companies now execute strategies that break down applications into APIs and microservices. Organizations need to know what challenges to approach and how to do so successfully. With a strategic and consistent path toward modernization that includes process discovery and collaboration, these organizations can answer IT questions in a nuanced, calculated, and data-driven way for real change.

Organizations using IBM i and IBM zSystems can harness an API-first approach to drive ERP transformation as part of their continuous modernization plan. Specifically, this can come by accelerating automation, increasing security measures, and seamlessly integrating cloud applications. With these capabilities, enterprises unlock new levels of operational excellence, scalability, and real business value. For example, the Netflix Cosmos platform, built on an architecture that features an API Gateway, manages roughly two billion API edge requests daily. The platform also powers Netflix's 24/7 availability, capacity to scale for further growth, and maintenance of optimal speeds. The most significant consideration for anyone tasked with driving major organizational transformation is doing so in a way that delivers the most value for the company.
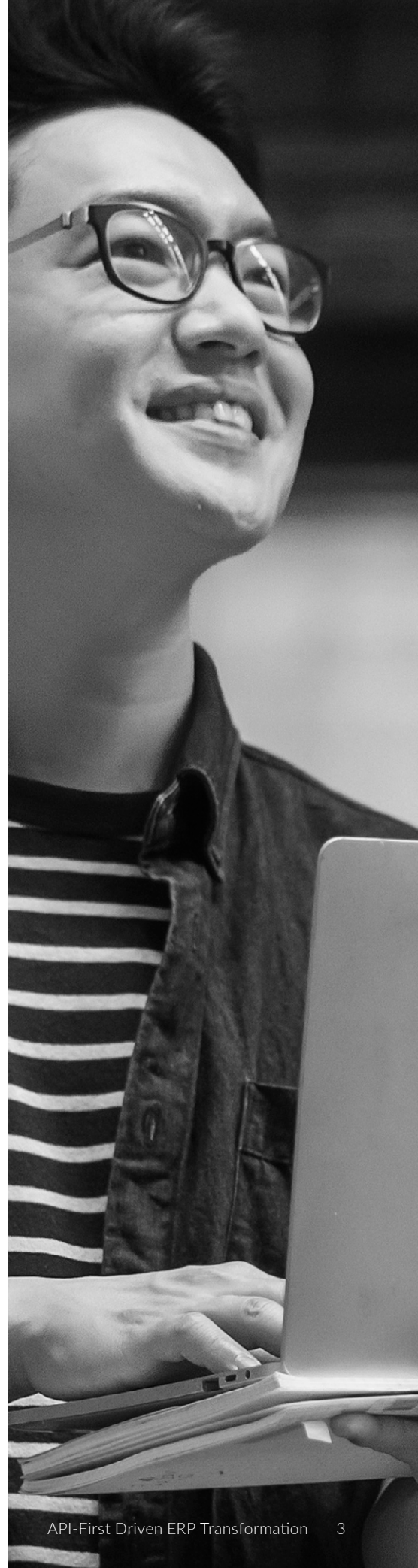
ERP advantages and challenges

ERP software emerged in the 1990s as a response to the growing complexity and inefficiencies of fragmented business systems. Traditional business software solutions were often siloed, leading to information bottlenecks, redundant processes, and difficulty obtaining a holistic view of organizational operations. ERP software aimed to address these challenges by integrating core business functions into a unified system.

ERP software is relevant because it offers numerous benefits across different industries, including finance, HR, manufacturing, etc.

Some key advantages include:

- **Streamlined operations:**
  Centralize data and processes, eliminate duplicate efforts, and improve operational efficiency.

- **Improved decision-making:**
  real-time visibility into key performance indicators (KPIs) and the generation of comprehensive reports, enabling data-driven decision-making.

- **Enhanced collaboration:**
  Facilitate cross-functional collaboration by enabling departments to access and share information seamlessly. Promote effective communication, coordination, and knowledge sharing within the organization.

Despite the benefits they offer, many businesses have traditionally struggled to make effective use of their ERPs, faced with the growing technical complexity of their IT environments — including the need for more and more internal and external integrations, thus hindering their ability to advance organizations in the face of industry shifts.
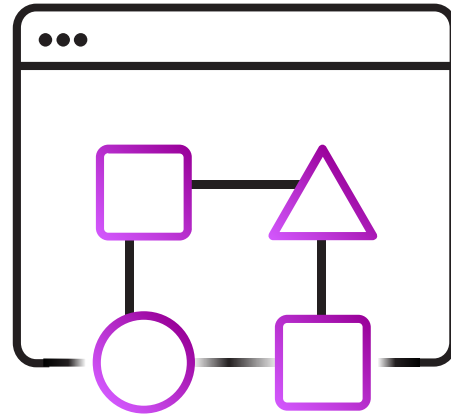
Some of these challenges include:

- **Integration complexity:**
  ERPs may face challenges integrating with external systems, third-party applications, or emerging technologies. This limits interoperability and data exchange that inhibits organizations from fully leveraging the potential of digital ecosystems.

- **Legacy architecture:**
  Many ERPs were built on older, monolithic architectures, making them complex, hard to maintain, and resistant to change. This slows down innovation and impedes organizations from adopting newer technologies.

- **Vendor lock-in:**
  Some ERP systems result in vendor lock-in, where organizations depend on a single vendor for support, updates, and enhancements. In addition, there are instances where upgrade paths to versions that would deliver new functionality are not provided.

To overcome these struggles, organizations have looked to APIs to come to the rescue. APIs provide a layer of abstraction that enables seamless integration, interoperability, and flexibility, often requiring no changes to the underlying ERP application. APIs are also incredibly helpful in integrating legacy on-premises ERP systems with cloud-based software and infrastructure.

Code-first approach to APIs

Organizations have used a code-first development approach for many years to develop their APIs. A code-first approach usually involves coding an API from business requirements and then producing a machine-readable API definition from that code. An API code-first approach means an organization has a design process distributed among various code documents. Opting for a code-first approach to build APIs for ERP transformation can present numerous challenges.

Let's explore the drawbacks of the code-first approach:

Bottlenecks:

The code-first approach often follows a waterfall model, where API development progresses sequentially. This setup can lead to bottlenecks, as different teams cannot work simultaneously on different phases, causing delays and inefficiencies.

Wasted time and money:

Modifying an already-coded API can be costly in terms of time and resources.

Ineffective APIs:

When you develop APIs through code first and receive feedback after several iterations, you may end up with unnecessarily bloated APIs. These APIs might contain redundant resources that users don't need, making them less efficient and user-friendly.
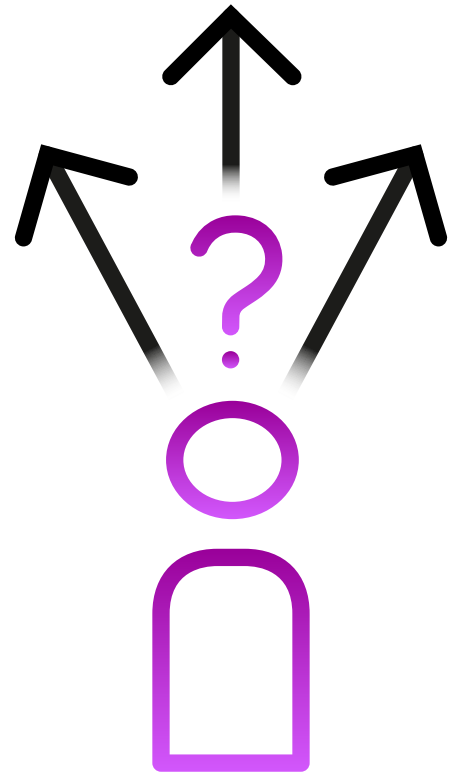
While the code-first approach might offer the allure of faster API development, the potential complications and issues that arise along the way can overshadow any initial time savings. For these reasons, companies increasingly adopt an API-first approach to transform their ERPs.

Why an API-first approach?

An API-First approach to building software products emphasizes the importance of treating APIs as reusable and easily accessible products that client applications consume. Unlike traditional methods where APIs are added later, API-first involves designing products around an API from the start. APIs are treated as critical business assets, and each API is designed around a contract written in an API description language. This approach ensures consistency, reusability, and broad interoperability. By starting with an API contract, organizations can better plan API design and gather feedback from stakeholders before writing any code.

According to the "2023 State of API Report" from Postman[1], organizations are increasingly adopting an API-first approach to integration.
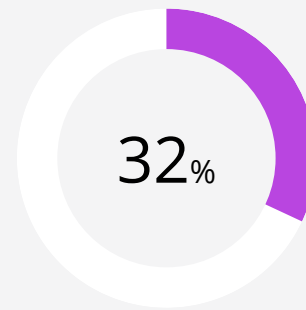
Increase in API-first organizations

2023   **2022**
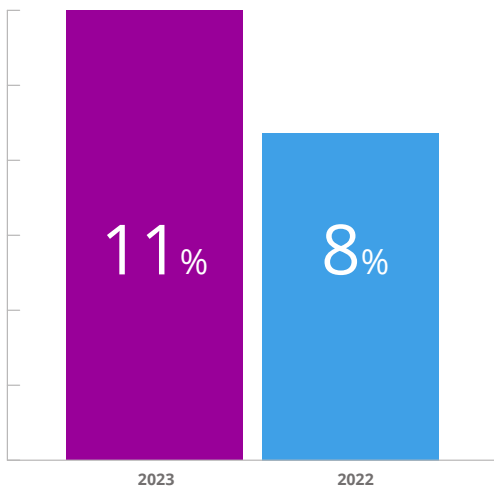
The report shows that 11% of surveyed organizations identify as API-first, a rise from 8% in the previous year. This shift signifies the growing recognition of APIs as foundational in modernizing enterprise systems and driving digital transformation.

Small companies with 100 developers or fewer have mainly observed the advantages of adopting an API-first approach.

32% of respondents from such small companies strongly agreed that API-first organizations tend to onboard faster.

This finding aligns with the agile nature of APIs, allowing for modular development and independent deployment, enabling quicker integration and time-to-market for new features or services.

As companies scale in size and reach a significant developer base, the benefits of API-first practices become even more pronounced.
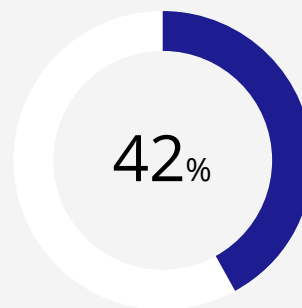
42% For organizations with 5,000 developers, 42% of respondents strongly agreed that API-first companies excel in onboarding speed.

This demonstrates that as organizations grow, the focus on API-first approaches gains importance, likely driven by the need for efficient collaboration, reduced dependencies, and streamlined development processes.

---

1   2023 state of The api report: Brought to you by postman. Postman API Platform. (n.d.). https://www.postman.com/state-of-api/

The report underscores the positive correlation between an API-first mindset and onboarding speed, particularly in small and large organizations. Embracing an API-first approach enables development teams to build front-end solutions with modern languages, then connect those solutions to applications and data via APIs without needing to touch legacy code. This empowers companies to onboard new developers swiftly, fostering an environment conducive to productivity and innovation.

An API-first mindset promotes developer satisfaction. Developers benefit from the modularity and independence of APIs, such as hiding underlying systems' complex characteristics, enabling them to work autonomously on specific services or features. This freedom accelerates development cycles, reduces bottlenecks, and fosters a culture of experimentation and continuous improvement. It should be noted that the emergence of the API-first approach aligns with the rise of microservices architecture.

Companies like Netflix and Amazon have transitioned from monolithic to microservices architectures, leading to new to successes regarding user experience and scalability. This shift has resulted in other companies exploring and adopting microservices, which heavily rely on APIs. Also, going API-first has gained traction as part of a continuous application modernization strategy.

The results are promising. But organizations with an API-first integration strategy will face challenges regarding API ownership, implementation, and delivery. According to Gartner®, "Burdening the integration team with API administration when adopting an API-first integration strategy creates unwieldy integration landscapes. To succeed with API-first integration, software engineering leaders should establish a separate API strategy that complements their integration strategy."[2] The report highlights key findings on the challenges of API ownership, implementation, and delivery.

An API-first strategy also contributes to establishing best practices for continuous modernization:

Designing with APIs in mind:
APIs are considered from the outset of ERP modernization, the focus being on designing the ERP system architecture with APIs seen as core building blocks.

Modularity and API-driven development:
The ERP system is deconstructed into smaller, self-contained services or microservices that communicate through the APIs, which also support integration and future scalability.

Data-driven insights and optimization:
APIs provide easy access to data within the ERP system, enabling organizations to extract valuable insights and make informed decisions using analytics and business intelligence tools.

Agility and flexibility:
APIs enable organizations to adapt to changing business needs, integrate with emerging technologies, and leverage best-of-breed solutions. The API layer facilitates incremental development, experimentation, and rapid iterations.

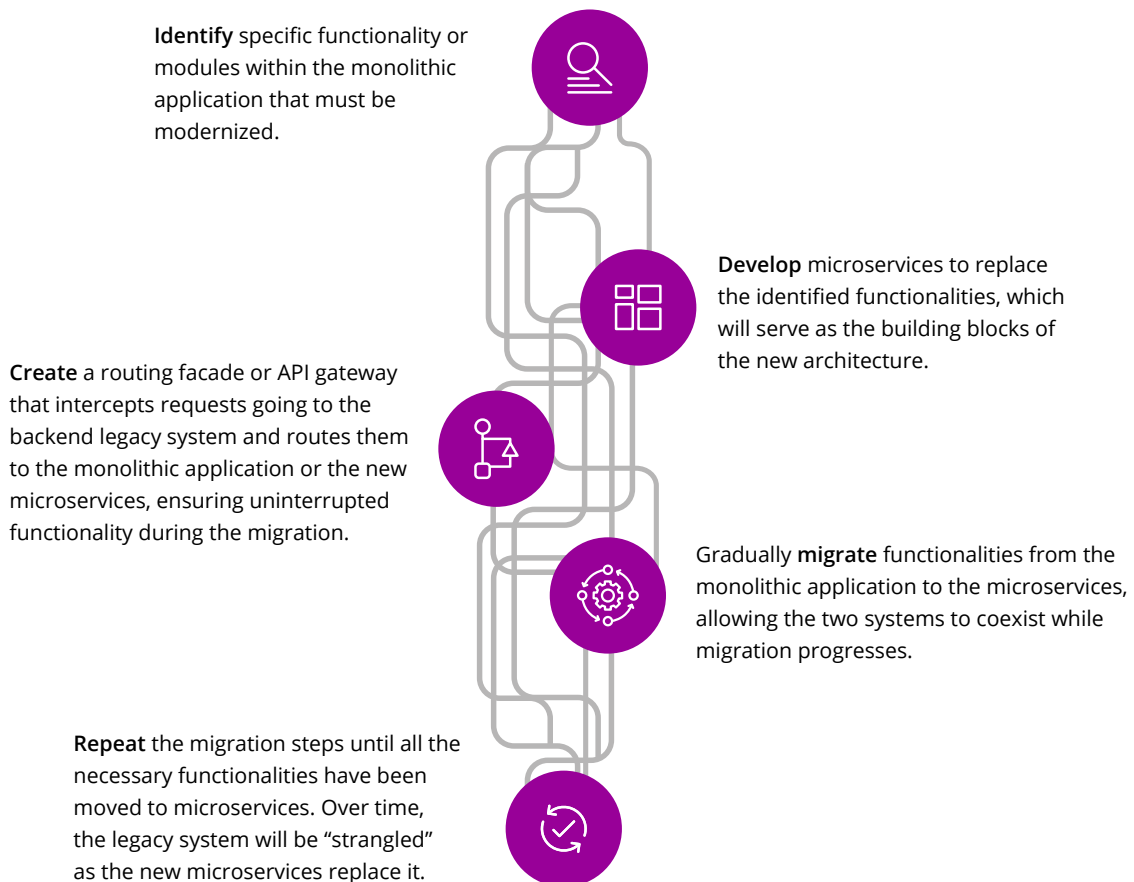**Developer experience and documentation:**
APIs are designed to focus on usability, simplicity, and consistency. Thorough documentation, SDKs (Software Development Kits), and developer resources are provided to enable developers to leverage the APIs effectively.

The Strangler Pattern:
Incremental modernization of monolithic applications

The Strangler Pattern is an architectural approach that facilitates the incremental migration from a monolithic application, including ERPs, to a microservices-based architecture. This strategy examines the business capabilities that a legacy application provides and creates small strangler services that encapsulate the logic of each capability in independent "bubbles."

Coined by Martin Fowler, this pattern derives its name from the way a vine slowly strangles a tree, replacing its growth. Similarly, the Strangler Pattern involves replacing parts of a monolithic application with microservices over time, allowing for a controlled and low-risk modernization process.

Identify specific functionality or modules within the monolithic application that must be modernized.

Develop microservices to replace the identified functionalities, which will serve as the building blocks of the new architecture.

Create a routing facade or API gateway that intercepts requests going to the backend legacy system and routes them to the monolithic application or the new microservices, ensuring uninterrupted functionality during the migration.

Gradually **migrate** functionalities from the monolithic application to the microservices, allowing the two systems to coexist while migration progresses.

Repeat the migration steps until all the necessary functionalities have been moved to microservices. Over time, the legacy system will be "strangled" as the new microservices replace it.

Harnessing the synergy:
Strangler pattern plus API-first approach

When the Strangler Pattern and API-first approach are used in tandem, they form a powerful duo for modernizing software systems, transitioning to microservices, and optimizing the development process. The API-first approach ensures a transparent and standardized interface for the microservices, while the Strangler Pattern enables the gradual, controlled replacement of monolithic components with modernized microservices.

# An API-first strategy:
Key enabler for cloud integration

As businesses ascend the ladder to the cloud, the allure of countless benefits awaits enterprises using IBM® i and IBM® zSystems. The cloud offers a compelling proposition of lower total cost of ownership (TCO) achieved through reduced footprint via logical partitions (LPARs) and cost-effective maintenance, freeing resources for strategic initiatives.

The transition from capital expenditures (CAPEX) to operational expenditures (OPEX) adds a new dimension of flexibility to IT budgets. Moreover, the cloud opens doors to innovative capabilities, such as analytics, voice-assisted artificial intelligence, and machine learning. Typically, these are beyond the reach of on-premises systems, as in the case of ERPs. The inherent scalability of cloud-based solutions further enhances their appeal and persuades businesses to explore cloud adoption. According to the 2023 Gartner CIO and Technology Executive Survey results, "46% of the organizations will increase their spend on application modernization (the top four technology area in spending) and 50% will increase their spend on cloud platforms in 2023 (the top three technology area in spending). Also, 47% will decrease investments in legacy infrastructure and data center technologies, illustrating the transition to modern technology platforms."[3]

Adopting an API-first strategy becomes instrumental in making the most of ERP transformation and seamless cloud integration. APIs are the linchpin for connecting ERP systems to diverse cloud-based applications and services, facilitating efficient data exchange and interaction. Implementing an API-first strategy streamlines integration processes, ensures standardized interfaces, and fosters interoperability between ERP systems and diverse cloud services. The API-first strategy also addresses critical challenges related to cloud integration, such as security. By embedding authentication, authorization, and encryption mechanisms directly into APIs, organizations can effectively control access to sensitive ERP data and enforce stringent security protocols for data exchange with external systems. This enhances overall data protection and safeguards critical business information.

Moreover, the API-first approach promotes flexibility, scalability, and adaptability, which are critical for successful cloud integration. By designing APIs with scalability in mind, organizations can easily accommodate future growth and expansion in the cloud environment, fostering agility in response to changing business needs.

Combining the power of the cloud with an API-first strategy unlocks new dimensions of innovation, streamlines ERP transformation, and empowers organizations to reimagine their businesses through continuous modernization and digital conversion. With the synergy of cloud capabilities and API-driven integration, companies can confidently embrace the future, knowing they are well-equipped to harness the full potential of the evolving ERP market and drive success in a rapidly changing technological landscape.
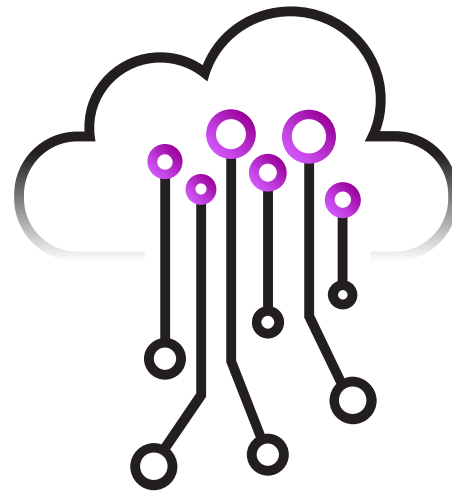
3. *How to Choose the Right Approach for Application Modernization and Cloud Migration by Stefan Van Der Zijden, Howard Dodd, Published 23 January 2023*

Continuous modernization plan:
API-first approach, modern application development, and Cloud

As businesses embark on the journey of modern application development and cloud adoption, they unlock many opportunities for growth and innovation. Understanding the advantages of cloud and application modernization lays the groundwork for this transformative process. The path to modernization is not a one-time event but a continuous journey. When facing unexpected modernization challenges, it is ideal to design your API-first strategy to execute an end-to-end, continuous modernization plan that drives automation and ongoing business results strategically.

This is a consistent, data-driven plan that includes six crucial steps:

01

Process discovery:
A detailed and data-driven understanding of IBM® i / IBM® zSystems' applications and ERP system usage is essential to kickstart the modernization journey. Process discovery helps assess workflows, estimate project timelines, and mitigate cost overruns. Collaborating with business and IT ensures comprehensive insights into application utilization, providing a solid foundation for modernization efforts.

02

Project prioritization:
Armed with a clearer view of workflows and processes, organizations can identify key areas for automation and modernization, delivering tangible value to the business. Prioritization aligns projects with success and secures buy-in from crucial stakeholders, showcasing direct links between proposed work and business impact.

03

Automation execution: An efficient way to gain quick ROI is through Robotic Process Automation (RPA). By automating mundane tasks within workflows, RPA enhances productivity and user satisfaction, freeing up resources to focus on more critical tasks.
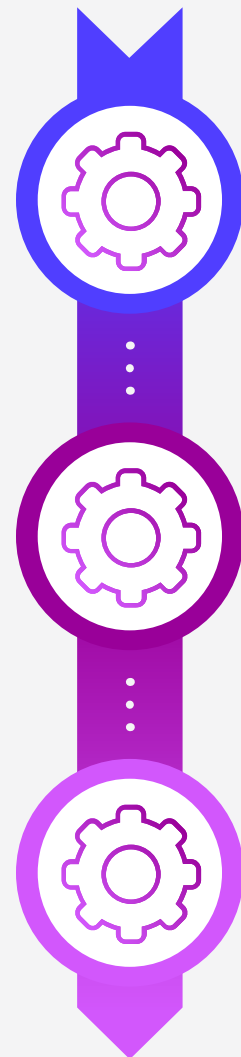
04

Modern user experience:
In pursuit of modern application development, organizations identify workflows burdened by legacy green screens slowing down user experience (UX). By creating user interfaces (UI) tailored to user workflows, featuring practical moving fields and intuitive flows, modernization efforts deliver improved user experiences.

05

Process integration:
Building APIs to integrate IBM® i / IBM® zSystems' applications with critical business applications (ERP systems) fosters seamless data exchange and collaboration. APIs bridge legacy applications and diverse cloud-based services, breaking down silos and untethering business operations from IT infrastructure constraints. Efficient API implementation optimizes response times, bypassing terminal screens or accessing data directly from storage systems.

06

Automation management:
Centralized management of robots and APIs through a DevOps-lite tool extends modernization and automation enhancements to developer environments. This automation streamlines processes enforces policies, and boosts developer productivity, paving the way for continuous improvement.
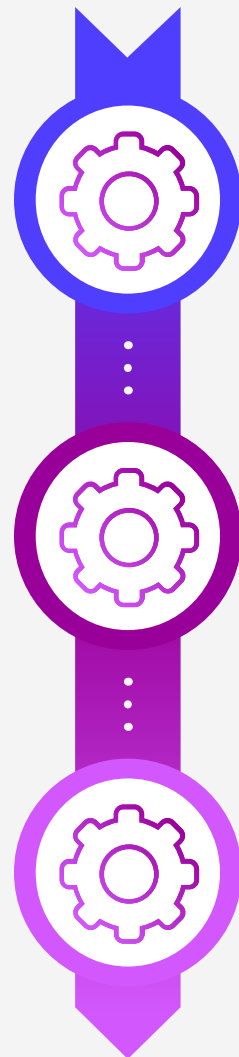
The success of an API-first-driven ERP transformation depends on how it connects the power of modern application development and the cloud with cohesive integration, improved security, and enhanced efficiency. By thoroughly understanding these applications and workflows through process discovery, organizations can confidently prioritize modernization efforts and effectively transform monoliths into microservices.

Leveraging APIs as building blocks empowers seamless integration between on-premises and cloud-based systems, which facilitate data exchange and foster adaptability in the ever-evolving ERP market. Every time new processes, technologies, or trends arise, your ERP software is affected. Industry volatility also warrants a review and update of the IBM® i and Z application workflows. Instituting a continuous modernization approach as an integral part of the daily operations best informs your API-first strategy best. It also helps you drive business resiliency through productivity improvements and higher ROI on your systems.

Process Discovery:
The crucial step

You must understand how your applications are used to build an effective API-first strategy. This includes knowing which workflows and processes are used most and which data are accessed most frequently. From there, you can build an intelligent plan that provides business results on day one.

Companies are increasingly breaking down core business capabilities into individual, reusable services. When moving to the cloud, refactoring legacy applications, implementing API/REST, or deploying microservices, process discovery helps you prioritize not only what goes first but also how to transform monoliths into services and, ultimately, microservices. For example, you can confidently know which parts of a monolith to break out and refactor to run in the cloud. Through process discovery, you can also identify how these microservices enable software applications to work dynamically and provide an enhanced user experience on all devices, platforms, and screen sizes.

It would be best to start by refactoring the most-used workflow or function available in one of the older, terminal-based applications your department supports. There are many details to consider, including whether the vendor offers a newer application version that provides APIs "out of the box." You may also need time to find the optimal integration point.

By incorporating tools that automate process discovery, developers can focus on developing great features and functionality for all who use your software — internally and externally. And developers no longer worrying about process discovery, testing builds, or scripts have more time for deeper productivity, while your organization is empowered to reallocate other resources toward other initiatives.

A Final Word

Today, IT leaders are under tremendous pressure to move to the cloud, refactor applications, and transform their ERP systems while deploying APIs and microservices to achieve business superiority. Yet, they need help finding insights into where to start, how to implement, and what to transform. Process discovery is the key with which businesses unlock fast and significant bottom-line value from their modernization and automation projects. With an Intelligent Continuous Modernization plan, built on knowledge of your company's engagement with its IBM® i and IBM® zSystems applications, you can leverage an API-first approach is paramount, empowering businesses to harness the full potential of modern application development and cloud capabilities.

As you look for tools to execute your plan, evaluate your choices based on which tools and vendors can help you automate and integrate processes, gather the necessary data to revolutionize your ERP landscape, and pave the way for continued growth and transformation.

Rocket Software is a 33-plus year partner of IBM® i and IBM® zSystems businesses looking to innovate through industry-leading modernization and automation approaches. We have a comprehensive portfolio of proven tools specifically developed to automate and support a continuous modernization plan and strategy.
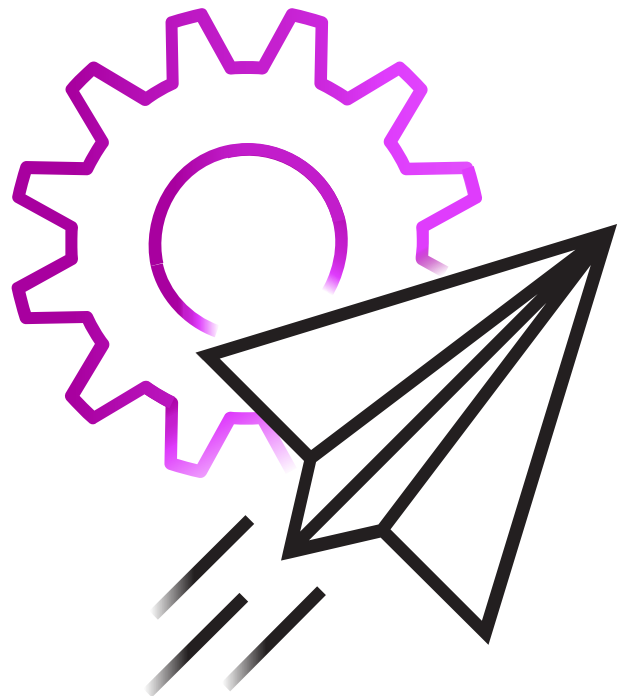
Rocket Software's Continuous Modernization Kit for IBM® i and IBM zSystems®

Rocket® Process Insights

A visual tool that lets you see your workflows in totality, giving you the information you need to build an intelligent and continuous modernization plan. It tracks how your organization engages with the data and business logic of your IBM® i and IBM® zSystems applications. It helps you build a data-driven strategy that eliminates redundancy and wasted time within your workflows. Then, you can leverage Rocket modernization solutions (integration with Rocket® API and Rocket® Modern Experience) or any other solutions of your choice to create modern user experiences and workflows that provide accurate results for your business.

Learn more about Rocket® Process Insights

Rocket® API

The only robotic process automation solution that delivers quick and significant ROI from automating your IBM® i and IBM® zSystems processes. It removes tedious, manual work that causes bottlenecks, introduces errors, and limits innovation, without needing in-house legacy development expertise. It enables businesses running IBM® i and IBM® zSystems applications to build workflows and innovative experiences that align with how customers and employees engage with your company and not how IT is built.

Learn more about Rocket® API

Rocket® Modern Experience: (UI/UX Modernization)

Quickly build modern user experiences for your mainframe and iSeries applications without relying on developers with COBOL or RPG programming expertise. Also, enhance your employees' and customers' web and mobile experiences while increasing productivity.

Learn more about Rocket® Modern Experience
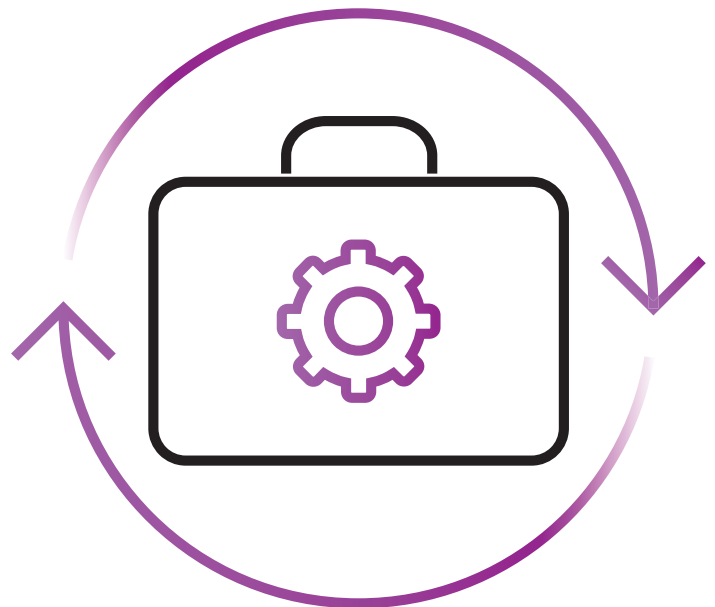
About Rocket Software

Rocket Software partners with the largest Fortune 1000 organizations to solve their most complex IT challenges across Applications, Data and Infrastructure. Rocket Software brings customers from where they are in their modernization journey to where they want to be by architecting innovative solutions that deliver next-generation experiences. Over 10 million global IT and business professionals trust Rocket Software to deliver solutions that improve responsiveness to change and optimize workloads. Rocket Software enables organizations to modernize in place with a hybrid cloud strategy to protect investment, decrease risk and reduce time to value. Rocket Software is a privately held U.S. corporation headquartered in the Boston area with centers of excellence strategically located throughout North America, Europe, Asia and Australia. Rocket Software is a portfolio company of Bain Capital Private Equity. Follow Rocket Software on LinkedIn and Twitter.

Check out Rocket Solutions for IBM® i

The future won't wait—modernize today.

Visit RocketSoftware.com >

Learn more