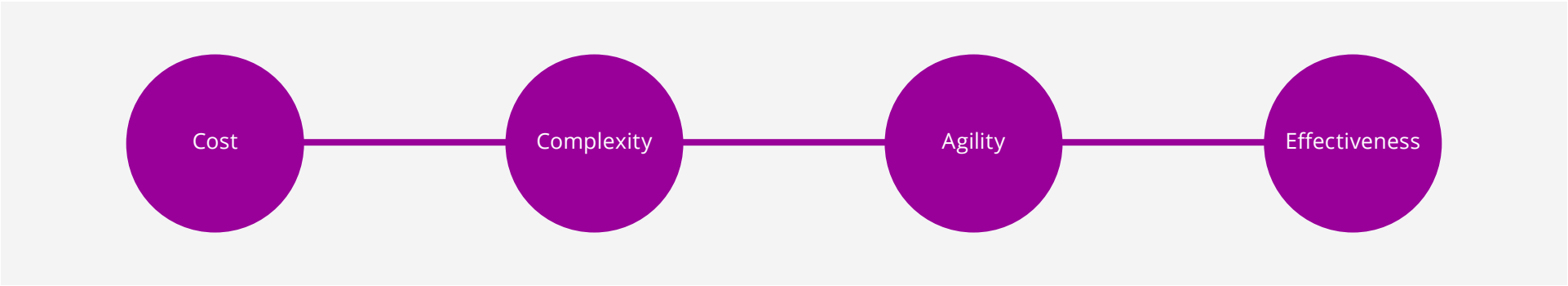
Rocket® software

# Plan Your Modernization Path

## The Modernization Maturity Model

# Contents




# The dilemma of application portfolios

**Your application portfolio represents a lot to your organization. You've built your business on it, and it's helping you be successful every day. But you can't stand still — your journey must include responding to rapidly changing requirements, competitors, new markets, and the downward cost pressures of doing business.**

### Let's set the stage.

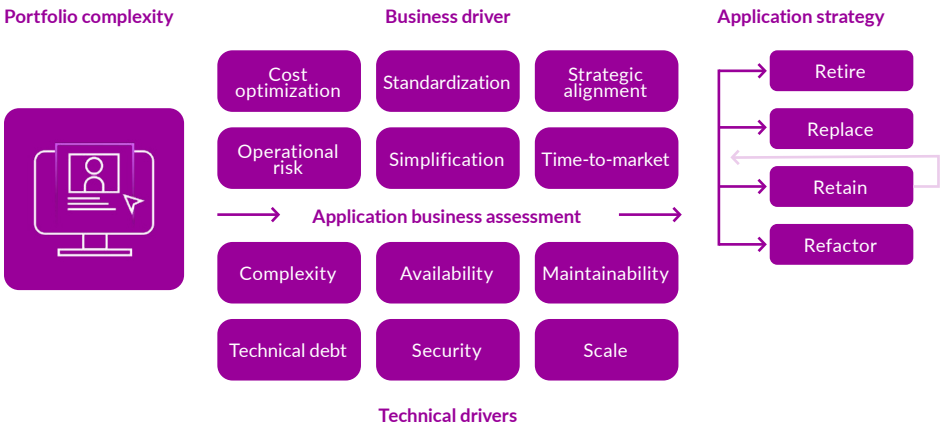
Your existing IT environment has likely evolved over years — even decades. As you've extended your core infrastructure — perhaps all the way from mainframe to containers — complexity has increased and costs have gone up. Sometimes at the expense of agility and effectiveness.

**Digital transformation is not a simple journey, but it needs to happen now.**

### Different points of view

Everyone across your organization has a different point of view, skewed by the lens through which they must deliver. There are so many options — replacement, refactoring, rearchitecting, and more. The right choice will depend on business and technical requirements. To complicate matters, the solution can be different for each application. This continuous transformation imperative must be carefully evaluated along with needs across the business. Not all solutions are right for every enterprise.

### Modernization — A pragmatic approach




Plotting the journey to change is, by its very nature, a difficult task. So how do you determine what's right for your organization?

### Start with conversations.

# The Modernization Maturity Model

## A powerful framework

Charting the path of change isn't easy. So, when holding important conversations about your digital strategy, it's helpful to have a framework to guide them.

The Rocket Software Modernization Maturity Model (MMM) was developed to help you plan and execute core business system modernization programs. It's not a roadmap or one specific destination — it's a framework to guide your exploration of the solutions that make the most of your IT investments while modernizing what's needed to compete in the digital economy.

**MMM has five lenses that help you think through the various aspects of modernization:**

Infrastructure | Application | Culture | IT Process | Management

Each lens has five stages. The stages are what's possible in modernization — but not necessarily what's right for you. MMM helps you choose your journey and your destination, based on the outcomes you determine are best for your business.

| Lens/Stage | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| Infrastructure | Mainframe | Distributed | Cloud | | |
| | | | Cloud-ready | Cloud-optimized | Cloud-native |
| Application | Monolithic<br>Proprietary<br>ACID transactions | N-tier<br>Portable<br>Virtualised | Loosely-coupled<br>Relational<br>"macro" container | Components<br>API Services<br>Containers | Microservices<br>NoSQL BASE transactions |
| IT Process | Waterfall | Iterative | Agile | DevOps | DevSecOps |
| Management | Initial | Managed | Defined | Measured | Optimizing |
| Culture | Departmental | Directed | Collaborative | Trusted | Performance |

# Personalized planning

Each decision to modernize comes with costs and risks, so the importance of having these conversations with all the affected parties is key. For some, a full-out transformation to Stage 5 is what's needed, and the risks and costs make sense. But that's not the right choice for everyone, so you need to figure out your organization's personalized journey. There will be tradeoffs — many of them — along the way. Having the right conversations about the right topics across the lenses of the entire business can reveal powerful solutions tailored for you.

**Let's look at each lens individually.**

# Infrastructure lens

**Infrastructure modernization is often a primary driver for broader modernization efforts.**

Examples include moving from a mainframe to a distributed or cloud platform or taking an existing on-premises application to the cloud. Reasons vary, including limitations of the original platform, such as cost, operator skills, or long-term availability of the hardware or operating system.

| Mainframe | Distributed | Cloud-ready |
|---|---|---|
| Single, centralized server | Network of x86 servers | Virtual machine-capable |

| Cloud optimized | Cloud native |
|---|---|
| IaaS and PaaS-capable | Written/Designed for cloud |

## Business driver example

Many modernization efforts start with a desire to move to the cloud. The organization may be interested in reducing costs, increasing agility, or addressing skills concerns by consolidating applications onto a single platform.

When moving from mainframe to cloud (infrastructure modernization), organizations should be able to reuse application logic and data platforms, and apply best practices to improve mobility and decouple dependencies.

Sometimes applications need to be rearchitected to maximize platform performance. Distributed platforms may deliver better price/performance, but deployment patterns must be updated to leverage capabilities of a distributed environment (application modernization).

The chosen deployment patterns may affect processes. Automation for environment creation, application deployment, testing, and elastic monitoring and scale can maximize productivity and efficiency(process modernization).

As these changes take place, it may require different management systems, and buy in from developers and operators (management and culture modernization). These are key areas to drive adoption.

# Application lens

**Application modernization frequently must happen to deliver new business functionality.**

This might include API enablement or providing other digital interfaces to support extension and reuse. Applications that were architected for one platform may require updating to take advantage of new capabilities as part of a changing infrastructure. Modernized applications help protect against disruption from competitors.

**Monolithic, proprietary, ACID transactions**

Performance optimized but not portable

**N-tier, portable, virtualized**

Portable, platform dependencies removed

**Loosely coupled, relational, macro container**

Application dependencies reduced, reusable interfaces

**Components, API services, containers**

Architectural constraints removed, reusable services

**Microservices, NoSQL, BASE transactions**

Independent abstracted services, scalable data access

## Business driver example

Today's customers demand digital interfaces — web and mobile access to new, transformative capabilities. The applications themselves need to be modernized in order to deliver those new, modern interfaces, with the ability to connect with external services and platforms.

Application modernization and the need to deliver new functionality to customers quickly often drives a move to Agile development (process modernization) and a need to restructure how data is stored and used.

The need to interact with external services and cloud computing resources often drives changes in the underlying infrastructure (infrastructure modernization). Restructuring the application to increase portability across platforms increases options for ongoing modernization efforts.

Moving from monolithic, proprietary applications to microservices and modern data structures opens up opportunities to move to a flexible management process (management modernization) driven by small, self-directed teams (culture modernization).

# IT process lens

**Delivering applications more quickly and with higher quality requires modernization of development processes.**

Improving the quality of code and speed of delivery requires a process that supports automation, continuous integration, and continuous deployment. Treating the infrastructure as you would code development with automation of the build, test, and platform deployment requires repeatability, and helps yield greater productivity. This can happen when applications are deployed in containers or public cloud environments.

| Waterfall | Iterative | Agile |
|---|---|---|
| Linear sequential phases of development | Cyclical phases of development | Development methodology, including requirements gathering and more |

| DevOps | DevSecOps |
|---|---|
| Combines software development and IT operations | Security practices integrated into DevOps |

## Business driver example

To respond to changing regulations, market conditions, or competitive pressures, organizations must quickly deliver new capabilities to maintain a competitive edge. Waterfall delivery processes are well known for taking a long time between requirements gathering and the first delivery to customers.

Modernizing the development process often requires access to automation and modern tooling, which may require a change in infrastructure (infrastructure modernization). Distributed/cloud platforms provide more cost-effective, dynamic, and scalable options, with access to new tools and services to accelerate delivery. It is easier to setup automated testing — that can be run at any time — at predictable, lower cost.

A move to DevOps is often paired with a move away from a monolithic application structure (application modernization), enabling focused development efforts led by smaller, Agile teams. Smaller, focused development efforts may require a move to a flexible management process (management modernization) and organizational changes (culture modernization).

# Management and culture lenses

**Modernization of infrastructure, application, and development processes to deliver value requires that people and organizations change.**

Change can often be uncomfortable, so it is common to find resistance. Getting an organizational sponsor(s) as well as buy-in from the team(s) for updating both systems and processes that support news ways of working is key.

## Management

**Initial**

Unpredictable processes

**Managed**

Project-driven, reactive processes

**Defined**

Organization-wide standard processes

**Measured**

Measured, controlled processes

**Optimizing**

Stable, flexible processes

## Culture

**Departmental**

Team-defined practices

**Directed**

Change program-driven

**Collaborative**

Coaching, not directing

**Trusted**

Self-directed

**Performance**

Team-driven change

## Business driver example

The desire to create an organization that is more responsive to customer requirements and more appealing to workers drives management and culture changes. Providing modern tooling (infrastructure modernization) and modern development practices (process modernization) support the broader organizational changes. Changes to the application structure (application modernization) can remove barriers to creating small, self-directed teams and a structured, flexible process.

# Key questions and considerations

**Now that you have seen some examples of the ways the different lenses and stages interact, here are some key questions to consider as you begin your conversations.**

Portfolio assessment and analysis:

What is the application composed of and how it is connected?

Transition to Agile/DevOps:

What is the delivery model for the application? Does it need to be accelerated and streamlined including automated testing?

Application rejuvenation and currency:

Does the application conform to the latest on technology support, regulatory and internal guidelines?

Remove platform dependencies:

Does the application include technologies that are unique to the current deployment platform?

Decouple application dependencies:

Does the application rely too heavily upon adjacent applications, data, or other technologies?

Operational process modernization:

Who manages and controls the application from inception to products? Are separate online and batch cycles required and can activities run in parallel?

API/SOA enablement:

How can we reuse important business functionality by exposing it in new ways?

Review operational requirements:

Are the current architecture and operational parameters based on limits of the current platform? What were the original requirements?

Data modernization:

Can the application data be leveraged as part of business intelligence or machine learning by extending access?

People and culture:

Who is responsible for the application? Are their skills, processes, and IT tools and technology sufficient to support change?

# Most importantly

## It's an ongoing process

**As you progress in your modernization journey, leveraging the MMM, your needs will change—so these conversations will be ongoing.**

Sometimes you'll need to make changes in more than one area at the same time, or some prerequisites to change may become apparent. Keep in mind — your desired business outcome is the key.

Making changes in your current maturity, such as automating testing or switching to Agile techniques, will help make the transition to the next maturity level easier — including reducing risk and increasing speed. But also note that the effort to transition between levels varies, as some transitions are more complicated than others.

## How do you get started?

**Getting started really is the important part.**

And remember, the goal is to meet your business needs, not necessarily to achieve Stage 5 of each lens. There's one more question that cuts across all discussions around modernization:

### What is the application composed of, and how is it connected?

Understand your business and technical drivers for each application in your portfolio so you can determine what is required to achieve your business goals. The majority of applications still represent considerable value to your organization and can be modernized to deliver on current and future requirements.

The Modernization Maturity Model helps you understand what is required to progress along each stage and the dependencies and considerations that also need to be addressed to deliver successful outcomes. By using the Modernization Maturity Model to define your current state and your desired future state, you can communicate the strategy more clearly across the organization.

And there are things you can do now to get started, even if your final destination is the cloud.

Sometimes you'll need to make changes in more than one area at the same time, or some prerequisites to change may become apparent.

Understand both your business and technical needs for each application in your portfolio to align with your business goals effectively.

# About Rocket Software

Rocket Software is the global technology leader in modernization and partner of choice that empowers the world's leading businesses on their modernization journeys, spanning core systems to the cloud. Trusted by over 12,500 customers and 750 partners, and with more than 3,000 global employees, Rocket Software enables customers to maximize their data, applications, and infrastructure to deliver critical services that power our modern world. Rocket Software is a privately held U.S. corporation headquartered in the Boston area with centers of excellence strategically located around the world. Rocket Software is a portfolio company of Bain Capital Private Equity. Follow Rocket Software on LinkedIn and X.

**Modernization.** Without Disruption.™

Visit RocketSoftware.com ›

Rocket software



MAR-10416_eBook_PlanYourModPathMaturityModel_V5