Rocket® software

Business case for modernization

Contents

Future proof your core business application

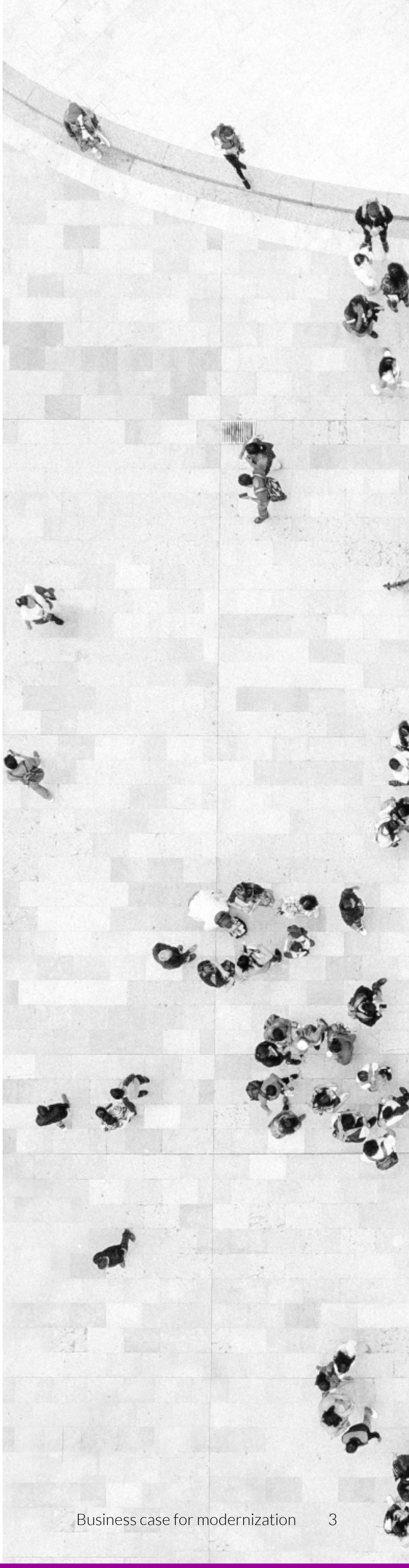
So, you've got (or inherited) a powerful MultiValue (aka PICK®) application that encapsulates the "secret sauce" for your business. Now what?

At the core of your business, your MultiValue (MV) application includes the business logic and rules to effortlessly handle your organization's unique workflows and automation. Originally popular because MV enabled citizen developers to quickly build applications, today those developers are retiring, and many haven't learned the ins and outs of modern user experience (UX) and Application Programming Interfaces (API) interconnections. While MV applications have stood the test of time, it's now time to modernize these applications.

Most MV applications have moved beyond the green screen. These days, MV applications have GUI front ends (maybe your app has even been through more than one iteration of a graphical front end).

Keep in mind that your MV application's User Interface (UI) is never done; the UI is what makes the first impression on your customers, prospects, and new hires. For ISVs selling their applications to new accounts, you don't want a dated GUI, the face of your application, to be what keeps your feature-rich application from being competitive.

And increasingly, you're being asked to refresh your application with a modern web front end and integrate it with other systems, including third-party apps and modern hardware like tablets and smartphones, but you don't know where to start.



← Back to contents

Industry best practices suggest using standard APIs to connect your MV application to other software solutions. And you've probably heard that RESTful Services simplify the development and deployment of web service clients.

Typical integration into a larger microservices architecture includes:

- Connecting with mainstream applications like Salesforce™, Workday™, and NetSuite™.
- Providing access to your application from third parties (internal and external) via APIs.
- Replacing and integrating parts of your application with third parties.
- Allowing users and customers to use tablets and smartphones to interact with your application.

The goal of this document is to give you a better understanding of today's modernization options and help establish the path forward for your application and business.

If you stay in character mode, you're counting your days."

VERN JORGENSEN
Sr. Application Architect,
Rover Data

← Back to contents

Understand your options

Rewrite your application in a traditional relational database

It's easy to get romanced into lifting and shifting your application onto another database, but it is risky, expensive, and can take years. MV applications consist of code (essentially PICK BASIC) and an embedded database/platform that support a different data model than relational databases.

The difficulty of lifting and shifting is not just migrating data but, more importantly, the business rules/logic.

Additionally, the nature of an MV data structure—where you can assign more than one value to a record's attribute—doesn't translate well into traditional relational databases such as SQL. The road is littered with abandoned efforts to move off MV, which can cost millions of dollars and, in some cases, have even bankrupted businesses.

Buy off the shelf

The idea of buying an off-the-shelf software system usually sounds better than it is. Don't be wooed by shiny screens that lack the substance and mature business logic you need. Your business-critical MultiValue application provides efficiency and allows you to stand out from the competition. Refined over decades, proven MV applications have perfected how to handle complex workflows, and employees use specific features deep in the application that might not be initially obvious.

Also, it's one thing to introduce a new point product where features are standard across many industries; it's another to try replacing the entire order management or accounting system.

Lastly, it's critical to consider the resources and effort required to customize the application to fit your needs. Packaged solutions either require you to change your business, or ask the vendor to customize the packaged application, perhaps to a level you are truly locked in.

However, if an off-the-shelf application does 85% or more of what you need, then perhaps this is a promising path. But that's rarely the case. It's not uncommon for a vendor to be resource-constrained, and the customizations to take years. These customizations can add delays and double or even triple the cost of a new software system. The entire process can distract you from your core business, which comes with immeasurable expense and risk.

Modernize

The way MV apps look often causes a business to start investigating options. But, before going to the extremes of ripping and replacing your MV applications, pause to consider recent developments with APIs and powerful new web technologies. Modern UI frameworks are community/developer-driven tools that are open-source and free to use. This means these tools are constantly being improved and supported vs. proprietary frameworks by vendors that can come and go. Redesigning the interface using tools like Angular by Google or React by Facebook provides JavaScript libraries to quickly kick-start your modernization efforts while using Rocket® MVIS to talk to your proven back end logic. Modernizing your current application will allow you to work with new, younger developers.

By utilizing APIs and today's web frameworks, you will gain access to talent who can see themselves working in this forked development framework. This approach empowers you to use the best tool for the job based on its strengths while developers focus on what each part is doing.

Focusing on the business logic layer, exposing it via normal RESTful layers, and communicating to an industry-standard UI transforms your MV application into a good citizen in any future modern application stack. It also allows current and prospective developers to understand each layer's place and work with them more easily.

Modernization fixes the need for everyones to know MV."

CHRIS BENNETT
Technical Lead,
MultiValue Consultant

← Back to contents

Young developers can increase productivity by using modern programming languages like Python, which is easy to use, efficient, and addresses skills gapswhile optimizing the performance of your business application. In addition to being a top language that new developers learn in school, Python has the added benefit of allowing your BASIC developers to leverage the Python open-source library. Hundreds of thousands of solutions at your fingertips let you quickly add new features to improve your application (i.e., using a Python module to quickly enable sending SMS text messages from your MV app).

You can also access MV data and existing business rules stored in a Rocket MV Application Platform environment. The Rocket databases now support object-oriented concepts, some via Python extensions and others via embedded native BASIC enhancements.

Do nothing

Whatever the circumstances, maintaining the status quo and hoping for the best is not a good business strategy. You must stop delaying your modernization journey. The time has never been better to modernize. Not only are you facing the continued difficulty of modifying your application, but the UI is making it harder to compete and train new developers and users. Does your senior management know you can modernize the back end to take advantage of new technology like DevOps, deployment, and RESTful APIs?

"I saw it important to demonstrate that modernization was also a viable option. The Rocket suite of tools has been the answer to all our needs. Python is mitigating longer-term risks of an aging workforce, and we're providing quite a lot of modernization with it and MVIS. The MV platform continues to evolve to meet new business needs, making it the logical choice to modernize as we move towards connected microservices rather than a monolithic application architecture."

GEOFF BISHOP
IT Engagement Manager,
Eurotunnel Le Shuttle

← Back to contents

The big shift

New technologies make modernization possible

Two exciting technologies have converged to make it easier to update the look and feel of your MV application while keeping you free of any proprietary technology:

- New community-driven open-source web technologies make front-end development a snap.
- Simultaneously, using Rocket's MVIS, you can add RESTful Web Services to your MV system.

These frameworks for building front-end web applications have come a long way, and most are free. Not only has cloud computing and SaaS revolutionized how we think about running our businesses, but the tools have continued to leap forward.

Benefits include:

01 Speed, flexibility, and performance.

02 Out-of-the-box functionality with default setups.

03 Reusable components.

04 Cross-platform development.

05 Rich user interfaces.

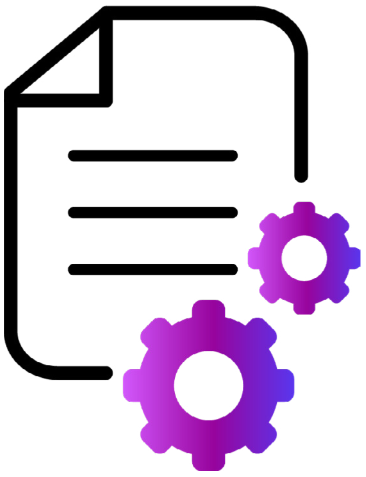
Similarly, APIs are now the universal method of connecting business systems.

When modernizing your MV system, you will use RESTful APIs, an architectural style that uses HTTP requests to access and use data.

Thanks to Rocket MVIS, writing RESTful APIs is quick and easy, and most importantly, this appeals to younger, full stack developers.

← Back to contents

Have a plan and field the right team



Deciding to modernize is a big step and can be overwhelming, especially since your application is probably complex and woven tightly into the day-to-day operations of your business. In all cases, it will take a group of professionals—MV developers, web developers, and business analysts—to execute your vision. Here are some tips:

Get buy-in from the top

This may sound obvious, but don't skip or underestimate the importance of this step. Getting buy-in from your executives and senior management team ensures the project aligns with your organization's business goals and objectives. It also sets expectations of how long the project will take and how many resources, personnel, and budget it'll require.

This is where you demonstrate to the non-technical people in your business that an MV application can be modernized and made more open; that modernizing takes less time, less budget and is less risky than rewriting or buying a package off the shelf; and will make sure you're not limited to an aging workforce.

Look beyond the technical team for ideas

An application is only as efficient as its users. Avoid modernizing your app in a silo without feedback from the people who use the application every day. Collecting feedback effectively from various users is critical to a successful roll out. This will also keep leaders from developing tunnel vision around any pet features they may want but aren't essential for daily power users. A huge bonus is that your power users can help train staff when you roll out new web apps or integrations. Surveying a variety of users for feedback and looking for trends will be an invaluable step to modernizing and understanding what's important.

← Back to contents

Now vs. later vs. never

Knowing what to modernize first vs. later vs. never will go a long way in helping you set priorities Certain parts of the process will have to be done regardless, like refactoring or separating the business logic from the presentation logic (UI) and choosing which front-end technologies to use. By breaking modernization into phases, you'll ensure the right things get worked on first. This is another reason why getting buy-in from the top is crucial to any modernization effort.

And remember, you can always utilize a terminal emulator like AccuTerm to link to less important screens, which can remain in character mode until that part of the application is tackled. Additionally, you may find that certain features of your MV application aren't needed or used at all. This can serve as an opportunity to trim the fat and simplify overall functionality.

Consider outsourcing

If you don't have developers with the necessary skills on staff, consider outsourcing these roles as an alternative. This path will allow you to tap into a broader talent network while offsetting some of the financial risks of hiring full-time. When outsourcing development, it's critical to have someone with a clear vision of the end goal overseeing the work. Your outsourced talent will unlikely have the same domain expertise as you and your staff. Also, compare several vendors, domestic and offshore, and select the one you are most comfortable with after careful review. As with most things in life, the strength of the relationship is critical to success. While outsourcing can be a great way to supplement your team and accelerate a project, it's not without risk. If left unchecked, projects can become a financial burden and strain your team's time.

Even if things go well, you must have a solution for supporting the application post-launch and evaluate whether to continue the outsourced relationship or bring those responsibilities in-house. It's essential to have a plan for the project and life post-launch, and to set expectations early and often with any outsourced team.

It is perfectly acceptable to weigh the outsourcing option or opt to hire directly. Should you choose the latter path, don't be afraid to be picky about how you hire. Expect to find someone who can become a domain expert and marry that expertise with their pre-existing technical knowhow.

Your executives will see the business benefits modernization provides if you complete projects early on that add value.

← Back to contents

If you have someone with limited experience on staff already, you can also consider a hybrid approach where you empower them to take on a team lead role and then supplement with outsourced resources. This allows you to offer career growth opportunities to your existing team member while also realizing the benefit of adding outside resources to accelerate the project.

Embrace APIS and restful services

APIs aren't strictly a component to modernizing your application's look and feel; you can also use them to integrate with third-party service providers to supercharge your application.

- Outsource sales tax calculation and reporting to Avalara, ditch your legacy email server for SendGrid, leverage Amazon for e Commerce or use Salesforce as your CRM.
- Expose endpoints to give third-party services access to your system, such as submitting orders, checking inventory levels, or checking the status of items.

All of this is made possible thanks to MVIS.

Get your MV developers to help

Your current MV developer(s) will play a crucial role in modernizing your application. If they are on the verge of retirement, they must help with proper documentation and educate your modernization team on how the system runs and what's mission-critical for the business.

Although difficult for Flat/SQL developers to understand, MV data structures align very well with modern object structures. Modern developers are used to working with object structures that include business methods tied to those objects; this relates easily to MV records and code. Your MV developers will quickly understand and excel at building RESTful endpoints that expose complicated MV structures like JSON and business logic like Methods. Together, the MV developers and the REST UI developers will find themselves speaking the same language (data and methods). Eventually, your newer developers will find the MV structures and code easier to understand.

Your MV developers will be key in describing and training your next generation of developers.

← Back to contents

If you have an MV developer willing to learn new tricks, it might not be as hard as they think. Modern languages like JavaScript are quite different from MV BASIC, but often a willing developer has the mindset to learn new things and solve challenges in development.

With an open mind and numerous online resources, an MV developer can learn the necessary skills to help modernize an application. At a minimum, they can write the MV API endpoints and know enough to help other developers bridge the gap and make the connections needed to ensure a successful outcome.

Your challenge as a business leader will be making an honest assessment of your current development team. Are they willing to adapt and work with others? How long do you have until your developer retires? In a retirement scenario, you must prioritize proper documentation and work towards a successful hand-off. Leveraging their in-depth historical knowledge will ensure the application they built, which is a valuable legacy, lives on.
If your developer is willing to modernize, you must invest in education, finding resources, and creating space and time for them to learn.

Have a plan so your MV app can and will continue to evolve

By hiring developers that know Python and object oriented-technologies to develop server code or do full-stack development, you and your MV application will be in great shape for future projects.

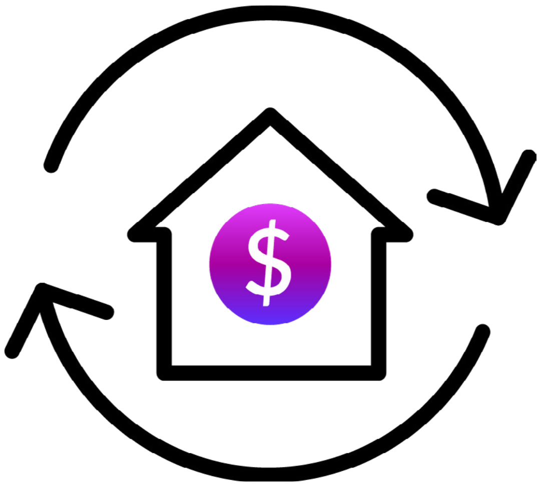
My experience has been if I put an MV developer that understands JSON in the room with a modern rest UI developer, they speak the same language of "data" and "methods". It's a different discussion than the SQL days when the talk centered on selects/rows/tables. They discuss a JSON payload for the data required. Then the MV developer will build the rest endpoint that exposes that data. Any business logic needed will be exposed as 'methods.' The bottom line is that the fight around normalization is gone."

PATRICK PAYNE, SR.
Software Architect,
Rocket Software

← Back to contents

Sierra Pacific Mortgage

Sierra Pacific is a 30-year-old provider of mortgage lending services based in Folsom, California. With nearly 1,000 employees across all its locations, the company is licensed in 49 states and has originated and purchased billions of dollars in residential loans.



01

Business situation:

- Reliant on a custom mortgage solution for loan writing, funding, etc.
- Application contains 30 years of business rules and had a stigma for looking old because it was character-based.
- Highly competitive marketplace, therefore, needed to have a competitive edge with efficiency.
- Constant business change.
- Needed the right tools.

02

Biggest upside:

- To recruit the best sales team, the company needed a highly functional tool that was also elegant to appeal to the sales team and help them grow and be successful.

03

The decision:

- Buy or build? Settled on “build” due to the amount of customization in the existing application.
- Developed ExpressLoan with the help of 30 offshore development resources.

04

The results:

- The new application interface is intuitive and uses APIs to call existing MV business rules.
- ExpressLoan is now viewed as one of the premier apps for the industry and attracts 4,000+ daily users.

← Back to contents

05

The technology:

- Utilized Angular and Google's free and open-source web application framework.

06

Time required:

- Three years from inception to full roll-out.

07

Advice:

- Involve users to ensure the development effort truly solves a business problem.
- Build or Buy is a big decision, so, have a good understanding of what you'll get if you decide to "buy".
- Consider the value for employee recruiting and retention.

← Back to contents

Eurotunnel Le Shuttle

If you've ever traveled by rail (Eurotunnel Le Shuttle or Eurostar) from the United Kingdom to France, you've gone through the Channel Tunnel, a 50km undersea tunnel that connects mainland Europe to the United Kingdom. It's the third-longest railway tunnel in the world, and 10,000 vehicles typically cross it daily. Opened in 1994, Eurotunnel is owned by French company Getlink and forms the linchpin of its billion-euro business. The tunnel provides two types of train shuttle services: one for trucks and commercial vehicles and a second for the passenger market. Eurotunnel also provides the infrastructure for Eurostar, the traditional railroad passenger trains that operate between the UK, France, Belgium, and the Netherlands.

01

Business situation:

- In 2018, a PricewaterhouseCoopers (PwC) IT audit of Eurotunnel MIS business systems identified that Eurotunnel's MV-based reservation system needed to be either modernized or replaced. Eurotunnel Le Shuttle faced a difficult decision:
  - Replace their proven, business-critical system, which would cost several million euros, take years to complete, and introduce a large number of risks, or
  - Adapt and modernize the proven MV-based reservation system, a project that would deliver maximum ROI with a smaller price tag and shorter time to market while minimizing risk.

02

The decision:

- Eurotunnel decided to modernize its reservation system to take advantage of new business opportunities that would generate additional revenue.

03

The results:

- Using MVIS, Eurotunnel Le Shuttle implemented a newer, more streamlined version of the application, which replaced older technologies with lightweight RESTful APIs. This resulted in a substantial reduction in application overhead.

- The Eurotunnel Le Shuttle Commercial team saw a business opportunity to base the price of passage on the size of the vehicle, for example, charging more for SUVs, which need more space, than for superminis. Eurotunnel Le Shuttle implemented what they named "a policeman" in the check-in lanes. This separate vehicle measurement system, based on laser technology, sends the actual dimensions of the vehicles that arrive at the toll plazas to be validated within the MV-based reservation system. This modernization project is driving a multi-million-euro uplift for their passenger business.

04

Time required:

- In an unanticipated twist of fate, in 2020, COVID-19 bought the project manager the time he needed to prove he could modernize MV.
- The first two phases went live in 2021.
- Many more innovative solutions based on MV are planned for the next five years.

05

Advice:

- You can't modernize just for the sake of it. Have defined business goals.
- Get buy-in from your executives.
- With Rocket MV, there's very little you can't do.

← Back to contents

Zumasys

Zumasys is a 20-year-old developer of vertical market software based in San Clemente, California. The company employs 35 people and owns six different packaged ERP software solutions, each at their own stage of the modernization journey. All the company's software products and consulting services are based on MV technology. One of Zumasys' modernization stories is highlighted below:

Rover ERP:

A manufacturing software suite with decades of longevity (some customers have been on the platform for 30+ years), Rover was already several phases into its modernization journey by the time Zumasys acquired it. Customers have long enjoyed its Windows-based GUI (written in Delphi). Under life at Zumasys, the team recently completed its re-launch of the GUI application as a more modern Windows app written in .NET/C#. Simultaneously, Zumasys sought to launch a Software-as-a-Service (SaaS) offering of the product to allow customers to go with a fully cloud-hosted option (delivered via Microsoft Azure). The Zumasys team also injected some familiar elements of its long-proselytized modernization recipe, launching next-generation applications including a Customer Portal (written with Vue.js), a web-based version of the Desktop client (written with Vue.js), a web-based reporting and business intelligence platform, a standardized API layer with connectivity to popular third-party services including Avalara, Send Grid, Salesforce, and more.

01

Business situation:

- The existing GUI application was written in Delphi and lacked support. Even though it was graphical, it had a dated "Windows XP" look and feel.
- Customers were requesting web-based options for interacting with the system, and some were looking for API options to integrate their systems into Rover.

02

The Decision:

- Replace the Delphi GUI application with a modern .NET/C# Windows application.
  - A third-party consultant was used to seed the project, and in-house developers were added. Over time the work shifted to 100% in-house.
  - Create a new validation layer to fuel the new standardized APIs.
- Build out an ecosystem of web applications that use the API layer to interact with Rover. Web applications include:
  - Reporting/BI platform (third-party integration).
  - Customer portal and web version of the desktop GUI created using a mixture of in-house and offshore developers.

← Back to contents

03

The results:

- Modernized the application for desktop and web
- Replicated existing functionality while adding new features.
- Increased the accessibility of the application for end-users and reduced friction around training and on boarding.
- Created new sales opportunities. The modernized app appeals to new customers who have outgrown solutions such as QuickBooks but are not prospects for something like Netsuite.

04

Time required:

- Delphi GUI to .NET/C# will take several years due to the complexity of the application. The API layer and web front ends took only a few months of work.

05

Advice:

- Know your audience! Feedback from your users is one of the most important factors when deciding what and how to modernize. If your users won't be compelled to use what you create, there's not much sense in creating it.
- Similarly, know your business risks. For Rover, carrying a legacy Delphi application with few options for support and maintenance was a real risk to the business.

Get Started Today

Wherever you are in the modernization process, the goal is to get started on a path forward, set realistic milestones, and quickly get to a Proof of Concept (POC). Building a POC will help create good processes and team structure so you can keep momentum as you get further into the modernization journey.

Contact your Rocket sales person or the Customer Solution Engineering team at solutioning@rocketsoftware.com today and get more information on our MV Modernization Discovery package and our API Jump Start package.

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.

The future won't wait—modernize today.

Visit RocketSoftware.com ›

Request a demo

Rocket software