Stop the Rip-and-Replace Spiral: A Practical Guide to Application Modernization

wovenware

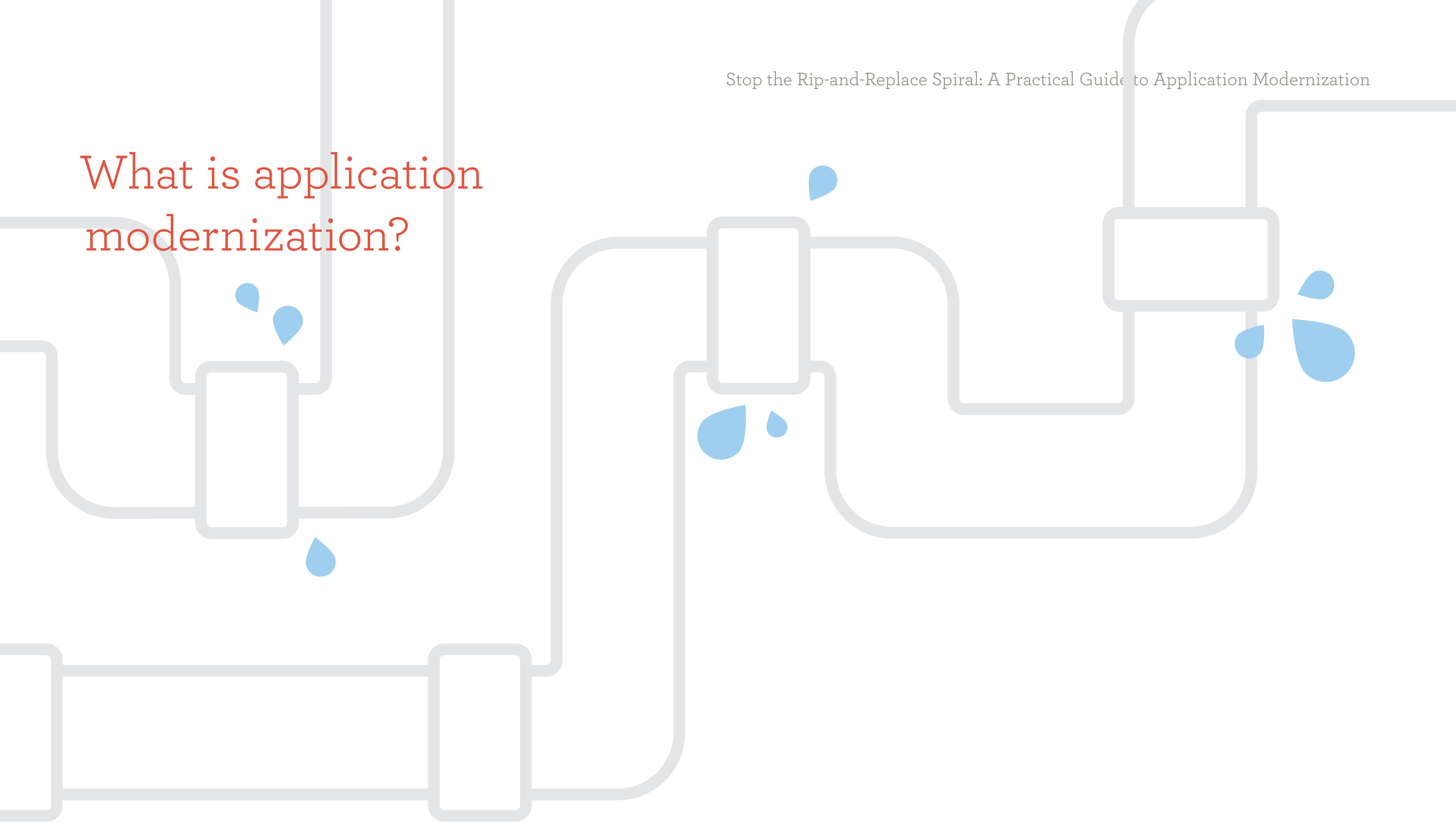
Introduction: Rethinking rip-and-replace

In many ways, being a CIO is like being a homeowner. As much as you love the structure you've built or bought, you can't help but worry, constantly, about its flaws. The hairline crack near the ceiling in the dining room. The crown moulding in the bedroom that needs to be reattached. The basement that gets damp when it rains—could that mean a crack in the foundation?

In the same way, IT leaders see all too easily the cracks and disconnects between the systems and solutions they've built or purchased. Over time, a once-elegant IT architecture can begin to resemble an old New England colonial that has been retrofitted and renovated for 200 years. The bones might be strong, but at some point, the whole things needs to be rebuilt.

Of course, wholesale IT architecture rip-and-replace is not only cost-prohibitive, it's practically impossible. The required downtime would be untenable for the business, and a new set of systems would be required anyway, by the time the whole project was finished. And, rip-and-replace tactics don't support flexibility, and flexibility is the core of modern IT systems design.

Rather than focusing on replacing old with new, CIOs are increasingly turning to application modernization, a strategy that builds on the strengths of an existing IT skeleton but updates the connective tissue.



What is application modernization?

Application modernization is a technique for updating existing IT systems in order to extend their useful life. “Useful” is the operative word here. Some systems require added capabilities that make them available on new devices, answering the needs of an increasingly mobile workforce. For others, it means transforming a user interface without fundamentally changing the plumbing underneath.

Put simply, application modernization is the redevelopment of an existing software application into a modern architecture so the business can continue to use and get value from the app. The transformation extends the value of existing software applications and data while allowing data reuse and improving user experience.

This approach delivers several benefits, including:

- Cost-effectiveness—because application modernization is about transformation within an existing architecture, it tends to be less time- and resource-intensive. In contrast, typical software replacement initiatives include extensive vendor assessments, pilot and beta-testing periods, and extensive training time to ramp up employees on new systems.

- Disruption minimization—downtime is one of the biggest enemies of technology advancement. Neither IT nor other employees have much patience for workday disruptions: they just want things to work. Customers are even less forgiving. So minimizing downtime through modernization rather than replacement can be a big selling point.

- Customization—revamping systems that already work within your organization’s unique architecture means that the solutions will be inherently customized. The writing of code to adapt aging software to a new operating environment or to meet new user requirements must be completely tailored to the particular needs of the business.

- No data migration—why migrate data into a new database schema when you can reuse your current database? Often, the weakest point in an application replacement project is the data migration process. With application modernization you can just reuse your existing data stores, thus eliminating your biggest risk and ensuing a successful project.

How does application modernization work?

There's no hard-and-fast approach to application modernization, but there are a few guidelines that can help you assess whether app modernization might be a good fit for your business.

Step One: Assessing your technical debt

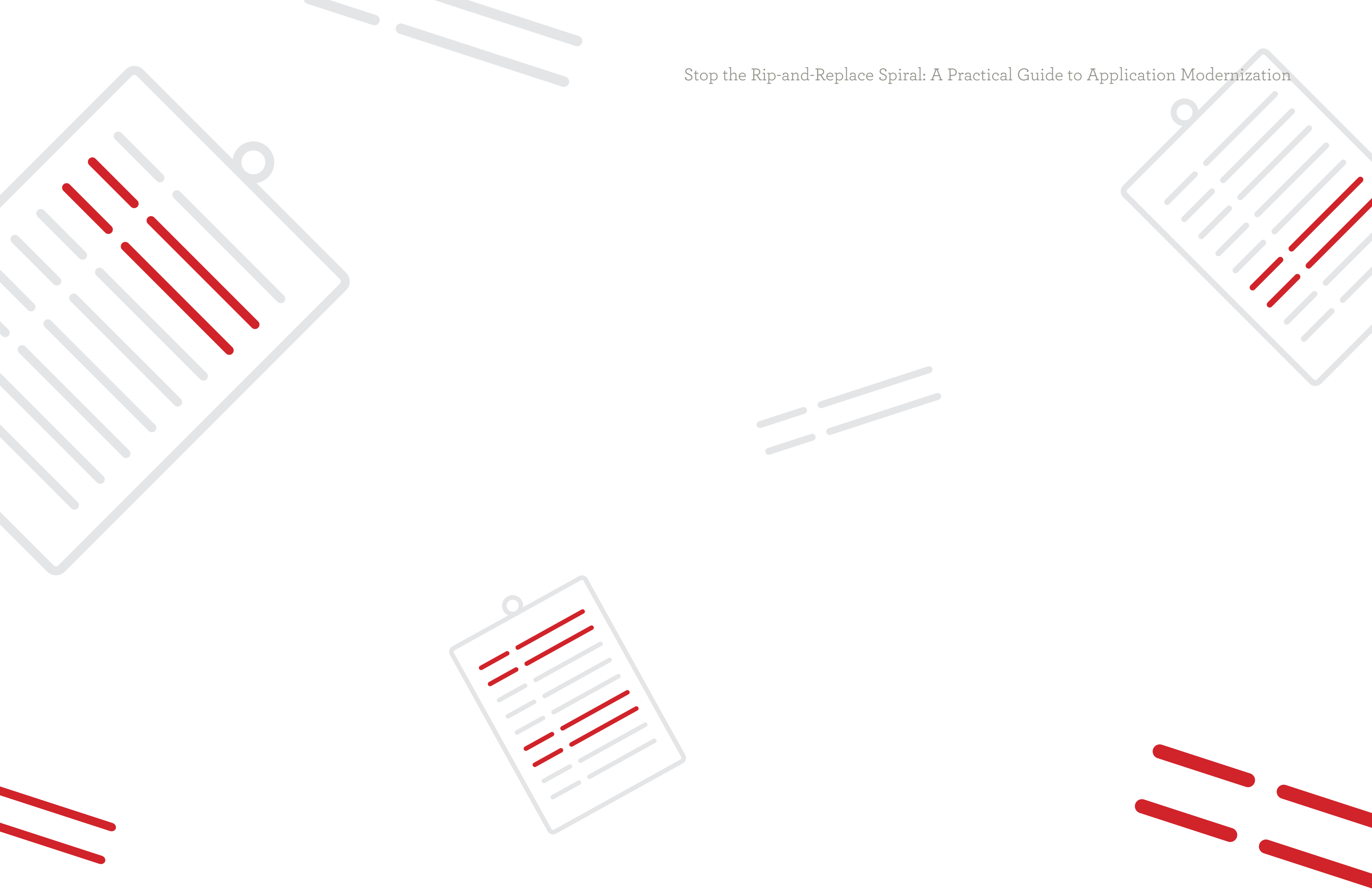
Of all the buzzwords to occupy space in the CIO's office, over the past decade technical debt is perhaps the most concrete and easiest to understand.

The concept is simple: IT investments depreciate. The value of hardware and software begins to drop once those systems reach a certain age, and the lifecycles for all technologies continue to shorten.

Depreciation is inevitable, but decisions made to adopt solutions as short-term fixes for technical challenges make depreciation worse. Every IT executive has faced the need to make quick decisions to answer immediate business needs, even knowing that the fix could cause complications down the road.

These quick-adoption decisions are like the compound interest on top of infrastructure depreciation. Together, they create a load of technical debt that can burden even the most prudent IT leader.

In an IT environment where writing down technical debt is paramount, application modernization is a smart strategy. You can extend the life of business applications and rebuild the IT infrastructure in small batches, over time.



Step Two:
Prioritizing your needs

Once an app lives within the organization long enough, it becomes critical. But it also hinders efficiency and growth in ways that go undetected—until they become undeniably obvious.

At some point, IT also has problems maintaining the application, after years of getting by with limited budget and unlimited demand for new functionality.

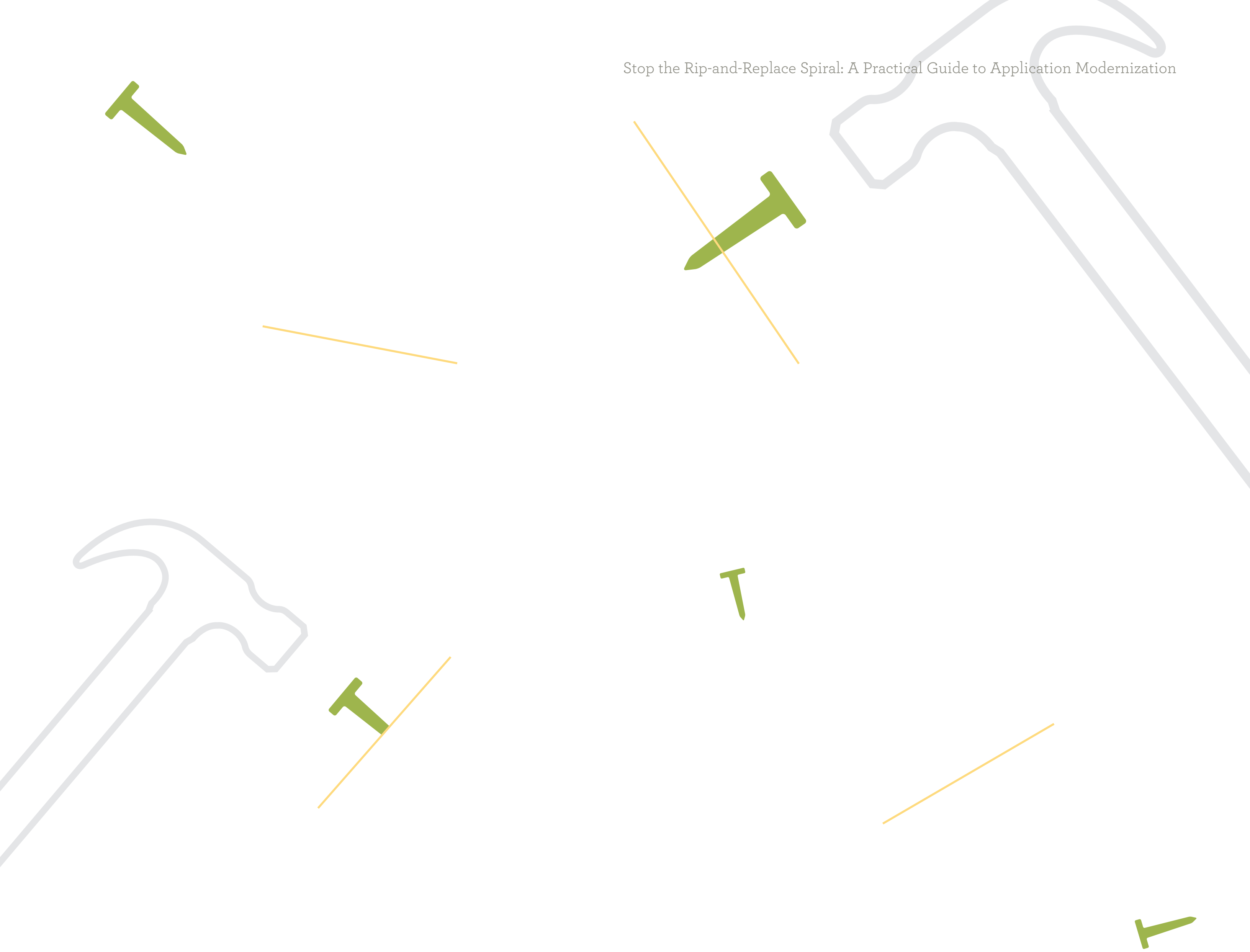
This is where technical debt begins to accumulate. At some point, the code has to be rewritten in a new language. But that also tends to happen across multiple applications, all at the same time. So how do you know which to update first?

Ultimately, any decision about application modernization is rooted in a single source: business objectives. Deciding which apps to modernize first depends on what you need to achieve as a business, and how the technology strategy will help to achieve those goals.

The following questions can help you prioritize:

- How old are your most-used applications?
- How many users rely on each application that is a candidate for modernization?
- What is the potential downtime?
- What skills do your internal team have, and how do those map to potential modernization projects?
- When will other business initiatives launch that would either affect or depend on existing applications?

By tying IT updates to business goals, application modernization becomes the perfect expression of the dual role CIOs now play as both technologists and business strategists.



Step Three: Securing buy-in

Once you've created your priority list of modernization projects, you can begin the process of internal communications. This is a multi-step process, as well, but it needn't be complicated.

First, figure out the business owners internally who are most affected by potential changes to an app. Next, gather and reflect back to them their feedback and "wish list" desired for a refreshed version of the app. Then, set expectations about which items on the list will be addressed and which need to wait for a future update.

Next, communicate throughout the implementation process. It's not necessary to be completely transparent about technical complexities, but it's essential to provide regular, direct, and open updates so that users feel a sense of empowerment and collaboration with the IT team.

Finally, circle back after implementation to ensure that users have the ongoing support and training they need—just as you would with a traditional technology implementation.

Do as much of this as possible in person, document your plans in email memos, and enlist key members of your team in the process as you are able: this helps to build trust and foster relationships across the organization that will serve as a support system if you encounter challenges or need to revise expectations during the course of the modernization project.

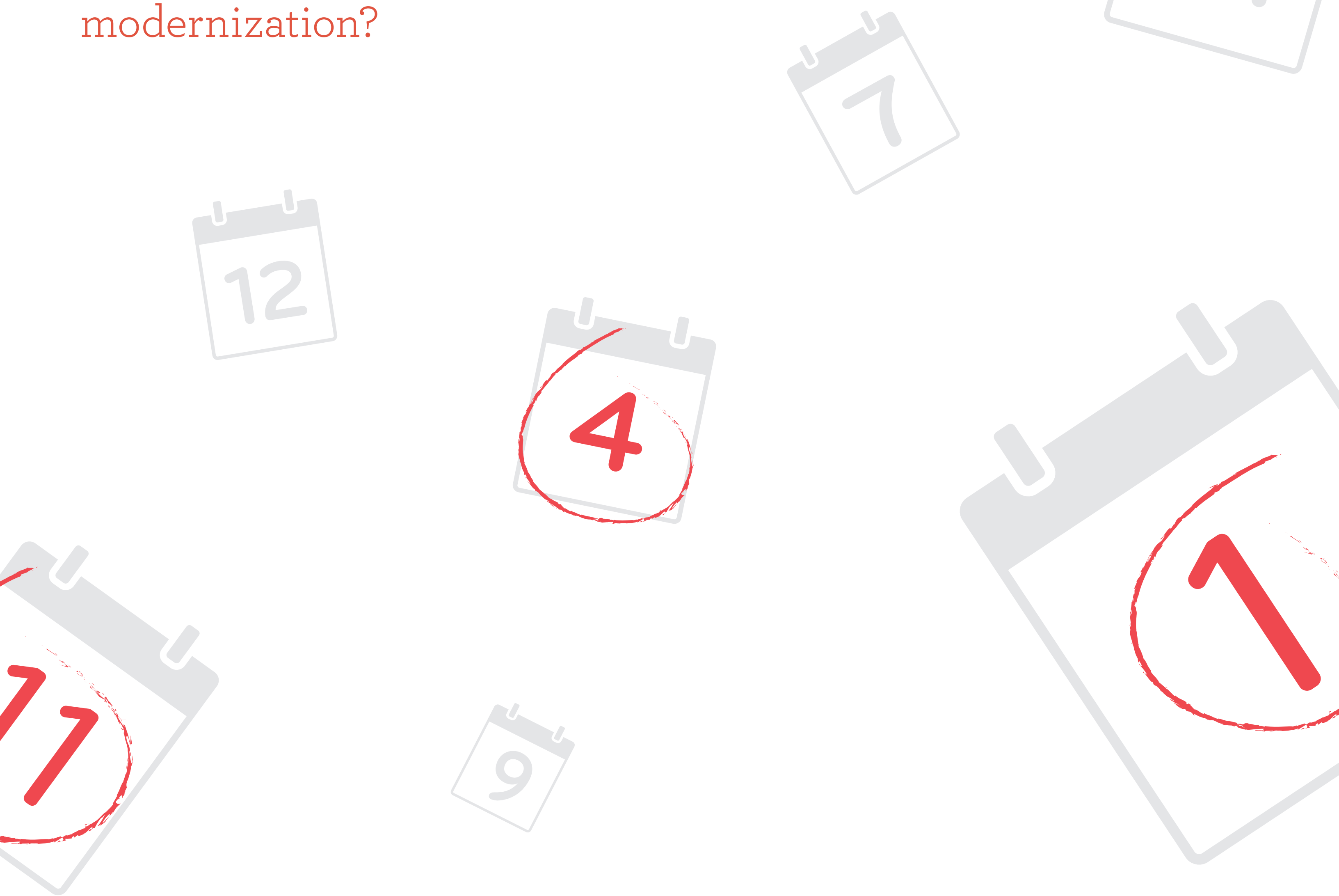
Step Four: Measure the results

It's impossible to know whether a modernization project is effective without benchmarking, measuring, and analyzing its performance. This requires a bit of work up front, but helps to make the business case for every application modernization project—now and in the future.

Accurate results measurement depends on good benchmarking and smart expectation setting. In a happy coincidence, this means communicating with internal stakeholders at the outset of a project so that performance measurement reflects human requirements for the new app. Benchmarking also requires gathering data about the existing app, such as launch and download speeds, in-app processing limits, and other factors.

The final stage of performance measurement is reporting: sharing with the rest of the business the results that are being reaped through modernization. This accomplishes two things. First, it lets the IT organization "prove" the value of investing in application modernization. Second, it enables the ongoing strengthening of relationships through greater transparency into IT strategies.

When should you consider modernization?



There is no set formula for assessing a given app's readiness for modernization. There are, however, a few rules of thumb that can help guide decision-making about revamping existing software. Those include:

- Look for strategic alignment—when setting annual IT strategy, pay attention to the areas that are most challenging or have dogged the IT function for a long time. These are generally the areas that invite software application modernization most readily.

- Measure ongoing performance—continuous software performance tracking and reporting within IT provides insight into the apps that are crashing, failing, slowing down, or frustrating users. This helps inform the process of modernization.

- Account for the whine factor—user complaints, whether overheard in the lunchroom or seen on social media, can be a massive source of insight into application performance. Smart IT leaders know how to separate the whining from the constructive feedback that sometimes hides inside complaints, and how to turn that feedback into insights that fuel positive change.

How do you estimate your application modernization project?

Every application modernization project is unique. The amount of technical debt, the complexity of the functionality, and even the selected programming language play a role in the development time needed to complete an application modernization project. So, it's impossible to provide an exact formula to estimate a given application modernization project.

However, experience has taught us that we can use guidelines that help shape an accurate estimate. In theory, it is simple: add up the time needed for development, testing, project management, and contingency. If you also need a project budget, multiply your estimated project time by your average cost rate.

But how do you actually go through that process? It all starts with a spreadsheet with an inventory of all the screens the application has in use.

First, the inventory will classify each screen as low, medium, or high functional complexity. Next, we'll assign a number of development days to each category. We use the followings as staring points:

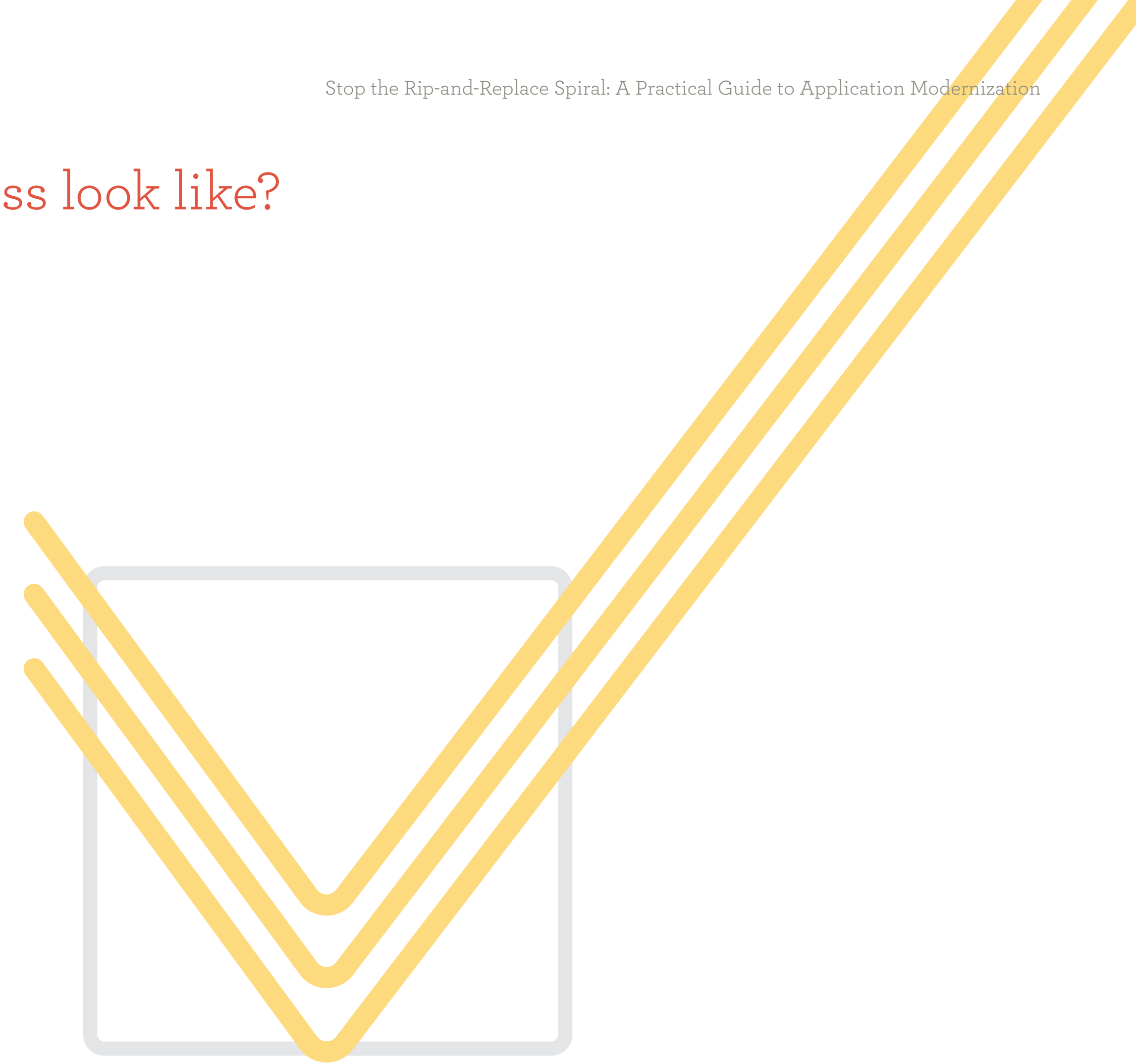
- Low complexity: 1-3 days
- Medium complexity: 2-4 days
- High complexity: 3-5 days

Finally, we add all the screens by category and multiply them by the assigned development days.

This approach provides an estimate of the development days needed to complete the project, but in order to provide an accurate estimate we need to add testing and project management time to the estimate, as well as a contingency time reserve.

Testing time can be calculated using several methods. One is to add testing days to your complexity categories or estimate how many days it will take a tester to run through the complete application and multiply that by a factor of two or three.

What does success look like?



Application modernization success brings the same kind of satisfaction that updating the electrical system in an old house brings: the process instills greater confidence that underlying infrastructure is stronger and less prone to hazardous breakdowns.

As a continuous IT improvement strategy, modernization also can act as a change agent, transforming rigid IT thinking about how to improve existing systems into a more fluid and flexible approach.

By starting to think of your IT infrastructure as a constantly improvable set of applications interacting together, rather than a firmly rooted foundation that cracks under the weight of its own permanence, your team can unlock creative ideas that lead to unforeseen solutions to business challenges.

Application modernization doesn’t look the same as tearing out software and replacing it with something new. Rather than a gut renovation, the process is more like a gradual upgrade of the vital systems within the architecture. It’s a modification of circuitry so that the application works for the modern needs of the organization and its customers.

We think application modernization holds the key to reducing technical debt and building a strong foundation for IT. And we can help you get there.

Need help with your application modernization strategy? Please give a call to:
Carlos Melendez, COO, Wovenware | 787 525 5372 | cmelendez@wovenware.com

wovenware

www.wovenware.com | info@wovenware.com