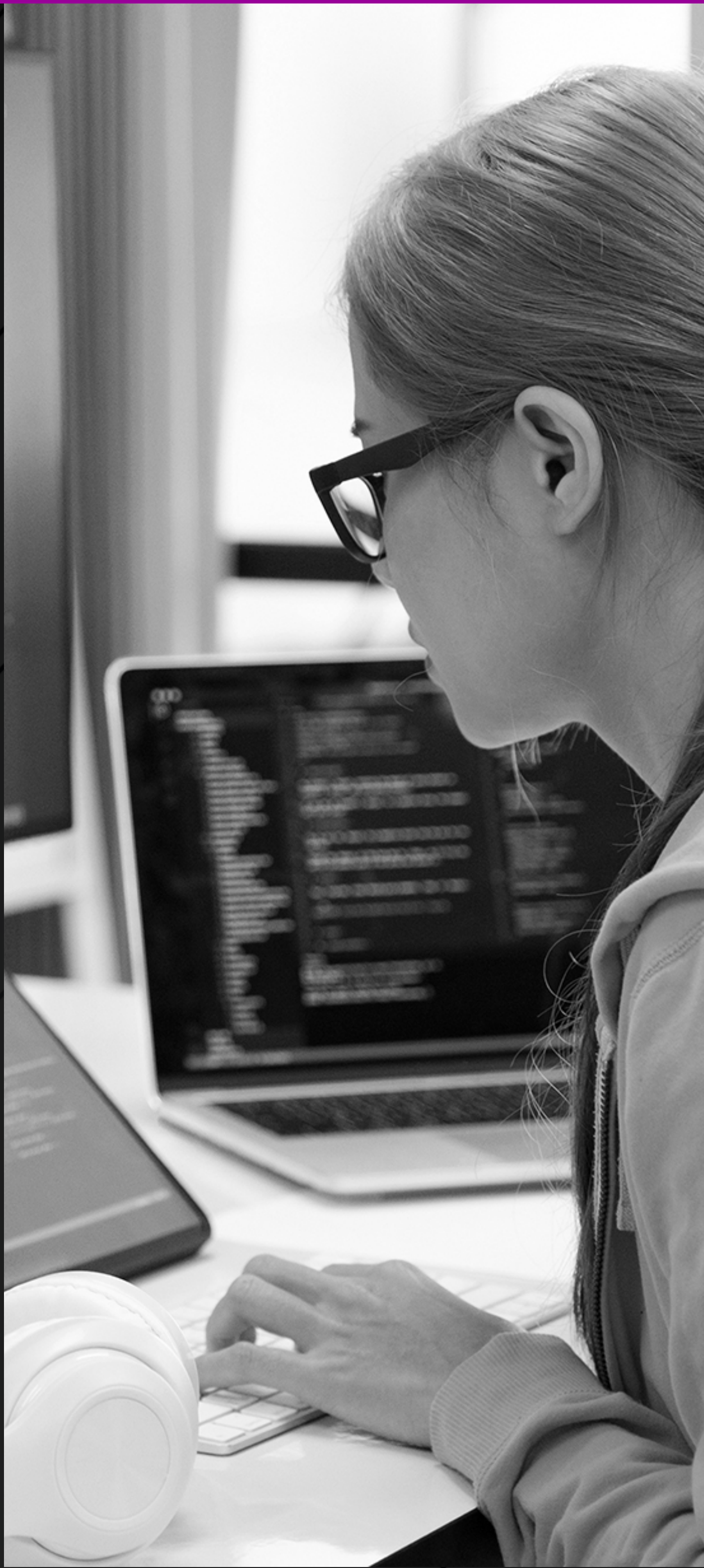
Rocket® software

The future of CI/CD for IBM® i+



Contents

03 Introduction

04 Automated testing and the right tool set are key to CI/CD

05 Leveraging test automation

06 The influence of AI on testing

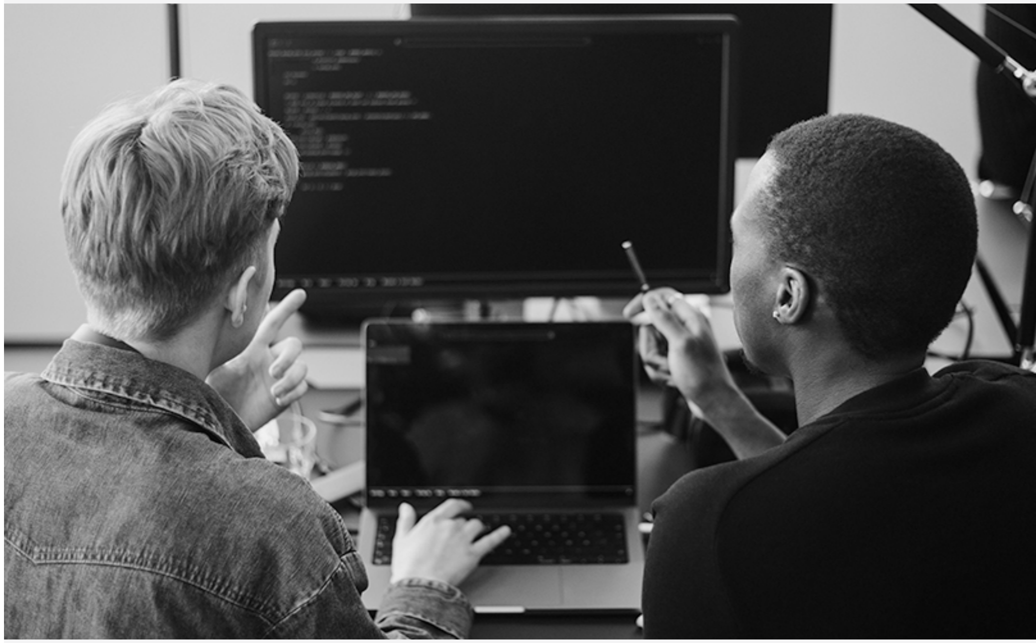
09 Intelligent testing is the way forward

10 Be future-ready

Introduction

Over the last five years, IBM® i teams have been expected to perform like other DevOps teams. That means ensuring agile development cycles with fast response times to customer and market needs, providing quality code in multi-code environments, as well as exposure to new teams, stakeholders, and workflows.

As organizations continue to recruit younger developers with modern views of what development should look like and what tools they need to navigate the existing applications — often without deep knowledge of how the applications were built and why — it's important for organizations to understand the nexus of continuous improvement/continuous delivery (CI/CD) and the trends influencing it.



IBM i owners need to be agile, work with modern toolsets, and break down silos. Modern DevOps is here to stay — and that's a good thing. It reduces time to market, increases enterprise agility, and makes businesses more resilient.

One of the principles of DevOps is CI/CD, which enables best practices and allows teams to work faster, without compromising quality.

In this eBook we will focus on two important components that make CI/CD work — automated testing and expertly implemented modern tool sets — and discover how other current trends, namely process discovery and AI enablement, have the potential to dramatically influence CI/CD.

Automated testing and the right tool set are key to CI/CD

Continuous integration is the practice of continuously absorbing new features, functions, and UI into applications, while continuous delivery is about sending code when it's ready and needs to be tested as an application at the site/server.

Once a continuous loop is set up, organizations have more freedom to innovate and experiment. There are two important parts to making CI/CD work:

01

Automated testing

Without it, the cycle slows down, and the potential for errors entering production increases. Can you get quality code released to the market quickly so your teams can get the feedback they need to fine-tune their applications to match the demands of the market?

02

The right toolset and the right implementation of that toolset

Do you have the right tools — configured and leveraged in the right way — to support and accelerate your CI/CD approach?

Leveraging test automation

Companies are moving faster than ever and automating testing to do it

In a 2021 Github survey, 57% of developers said they release code twice as fast as before, up from 25% faster just two years prior. 75% said they use or are planning to use AI or bots for test code review, and close to 25% of respondents claimed full testing automation. While IBM i development trends tend to lag behind the rest of the market, the writing is on the wall.

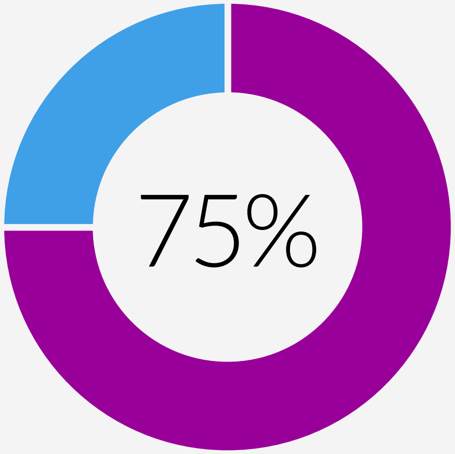
Additionally, we are in the initial stages of an emerging trend of process discovery. This is the practice by which organizations understand how users engage with an application and dataset, including the specific steps they take in workflows, including what data is accessed when, how long they spend at certain points in the workflow, which steps are skipped, shortcuts are used, and more.

Process discovery takes the guesswork out of what is needed to develop and test, accelerating the execution of automated testing. In short, testing and process discovery are evolving into more than standard steps in the DevOps process.

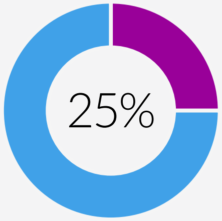
Process discovery is a prerequisite for accurate AI enablement — a trend that is fast-moving and likely to heavily influence parts, if not all, of the DevOps practice. To work, however, AI needs data. With data from process discovery, AI can support modern user interfaces, microservices, application integration and the automation of modernization development.

> Process discovery is a prerequisite for accurate AI enablement — a trend that is fast-moving and likely to heavily influence parts, if not all, of the DevOps practice.

With technology as transformative as AI, it is difficult to accurately predict its influence on DevOps more than a few years out. However, we believe in the prediction of AI's role in testing, explained in the next section, with high confidence.



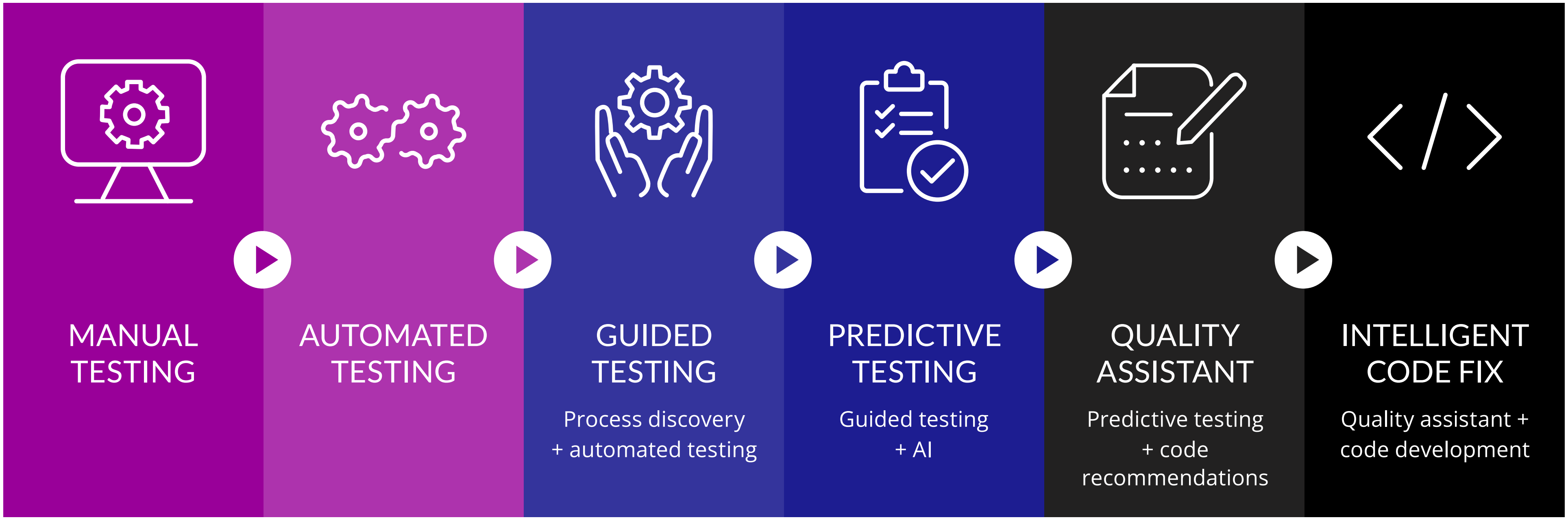
In a 2021 Github survey, 75% of developers said they use or are planning to use AI or bots for test code review,



and close to 25% of respondents claimed full test automation.

The influence of AI on testing

Evolution of AI influence in testing



MANUAL TESTING

AUTOMATED TESTING

GUIDED TESTING

Process discovery + automated testing

PREDICTIVE TESTING

Guided testing + AI

QUALITY ASSISTANT

Predictive testing + code recommendations

INTELLIGENT CODE FIX

Quality assistant + code development

Below is a brief explanation of each step in the evolutionary journey of testing. Manual and automated testing are self-explanatory; let's dive into the rest.

Guided testing

While understanding how an application is developed is important, what's key is how it is used. An application can be developed for CRM, for example, with standard database fields and typical workflows like "create new customer profile," "update profile," and so on. But what businesses really need to understand are their users, and their needs —the 'what, when, where, and how.'

When it comes to testing, if you know what you must test, you can test more often, and build more comprehensive testing scripts that can be reused. If you build a repository of tests, you can theoretically turn 2 months of testing into 2 days or even 2 hours. The repository becomes a knowledge base for the team that is continuously updated, improving over time.

Predictive testing

Once you have the data from process discovery and you're heading towards "shift left" testing, you can add AI into your testing process. Predictive testing knows what to test when code starts changing, and the AI can trigger pass/fail tests before QA even knows it's needed, lessening the burden on the testing team.

The AI will need the rules by which it will test your code. While the AI vendor will likely have a standard set of rules preprogrammed into the model, you will need to customize the model based on what your processes look like.

AI can also be built to establish relationships between tests that have already run and changes that were made. This means functional testing can be focused on specific areas with minimal human intervention.

Quality assistant

Think of this stage as an online shopping recommendation that suggests other things for you to buy based on what's currently in your cart, and what other customers bought in addition to those items. As quality assistance, AI can make recommendations about how to update code based on a failed test the AI model ran.

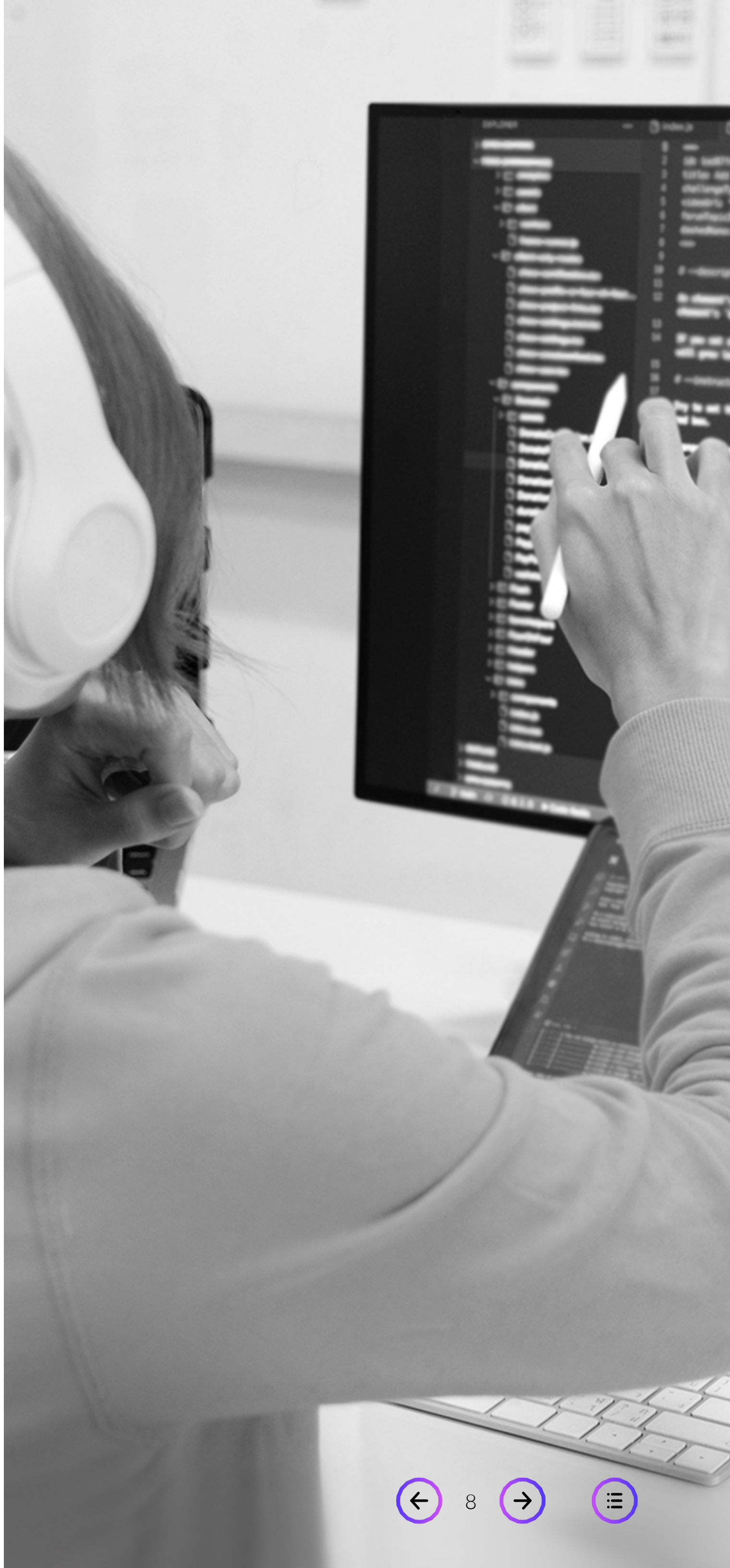
Intelligent testing is the way forward

It's probable that organizations will see an explosion of innovation enabled by QA when moving toward intelligent testing. The QA team can build more comprehensive test cases based on process discovery, and then share these with development. This empowers developers to fix errors more quickly and build higher-quality code from the beginning. While there will always be some level of functional fixes that come from user feedback, pushing higher quality code to production will increase the share of the user experience and in-situ feedback, thus speeding up the CI/CD cycle. This sets up the organization to be more competitive, uncover more opportunities for improving productivity, and align more quickly and closely with market needs.

In addition, intelligent code fixing raises the question of, "How does AI change an organization's approach to RPG?"

It's possible for AI to take over both the development and testing of RPG code, minimizing the resource challenges for IBM i systems. In fact, it's possible for an AI to be the IT admin, developer, and QA team for RPG, all in one.

We're likely years away from that last scenario becoming commonplace — if it indeed ever does. There have already been situations where the limitations of AI have caused concerns, including the serious ethical challenges AI introduces to the world economy, unconscious bias being built into AI models, and technology limitations, such as code ownership and transparency.



Be future-ready

Trends like process discovery and AI can sound overwhelming to a team still heavily using linear development and deployment processes. But if an IBM i team knows what's coming and how to prepare, they can set up their approach to DevOps in such a way that the sky is the limit for CI/CD, innovation, and experimentation.

Decide on risk tolerance for AI

Every organization will need to establish their AI risk tolerance. DevOps teams might decide to implement permissions across AI models in a similar way they do with human workers. How big of a "black box" that organizations might want to build around their critical data and applications will also be a factor. Still, one can see the potential of this technology to revolutionize not just testing, but also DevOps overall, especially within IBM i environments.

Reimagine QA in a world of intelligent testing

No matter how much automated testing you have, you will still need manual testing. But QA won't need to look for superficial bugs like color issues or the location of fields — because test cases can be easily built with intelligent testing and shared with development early in the CI/CD process. Instead, QA can focus on higher-level issues that could affect the functionality and usability of the application, such as regression. QA teams can also spend more time figuring out what needs to be tested and building more comprehensive test cases based on process discovery. The question is: what makes sense for both the DevOps process and the larger organization?

Build a data strategy... and stick to it

At its core, AI is a data synthesizer and juggler. It relies on high-quality historical data to feed the model and uses the data it receives to create uniquely aggregated outputs. Continuous data inputs then feed into the AI model, so adjustments are made to fine-tune its performance.

Start collecting data now

Since AI will rely heavily on good historical data to get started, organizations should start collecting data now.

While we can't predict the future, IBM i teams can plan for the evolution of intelligent testing. Teams know that they must be agile, work with modern toolsets, and break down silos — now.
To be future-ready, teams need to be:

- Laser-focused on CI/CD
- Clear on their risk tolerance for AI
- Understand where this technology may be integrated most effectively
- Develop policies around process discovery to ensure that any AI introduced to their systems will have access to high-quality historical data

With technology as transformative as AI, it is difficult to predict its influence on DevOps more than a few years out. We do know that intelligent testing is evolving. It's not a matter of 'if' but 'when' AI will play a significant role in testing. **Be future-ready**.

Learn more

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 ›

MAR-6798_DevOpsStudy-FutureofIBMi-CIDI_eBook_V4