Omdia
by informa techtarget

Modernizing code: refactoring and transforming business-critical applications with AI-assistance

Author: Michael Azoff

September 2025

Omdia commissioned research, sponsored by IBM

Contents

Introduction: all code needs modernization

Various technologists have described software as being less like gold and more like lettuce, i.e. it goes stale over time. It is inevitable that a software application will need to be revisited by developers. The application may have delayed refactoring to improve maintainability, there may be new feature requests, there may be new regulations and compliance requirements that necessitate code changes. The application environment, the surrounding systems that it integrates with, may undergo changes such as upgrades, new features, or replacement, and these changes require adjustments in the application. An indication that all is not as it should be is when an application has run reliably for 20 years, the original developers have departed, and current developers are afraid to modify it with change requests.

Java is the dominant programming language in the enterprise, and it has been evolving at rapid pace, Java Platform, Standard Edition is currently on version 24. However, organizations are unlikely to have migrated to the next edition at every new release, and the longer a migration is stalled, the greater the task to modernize the code.

Organizations that have invested in mainframes continue to enjoy their resiliency, reliability, and security. The modern mainframe can run Linux and Java but there are still

Omdia commissioned research, sponsored by IBM

business-critical languages such as COBOL that institutions, especially in finance, rely on. There is a continuing need to modernize COBOL and other programming languages on the mainframe, whether it is refactoring the code for improved maintainability, or transforming the code to a modern language like Java.

Business-critical applications that were built on earlier architectural practices, such as monoliths where code is tightly coupled, suffer from maintainability issues, not least of which is readability and understanding of the code. Software today is typically built in a more modular style, and many greenfield projects adopt cloud native computing architecture based on microservices running in containers and managed by Kubernetes. A microservice typically performs a single business function and is decoupled from other microservices to the extent that it can be swapped out in live production without shutting down the whole application, a significant advantage in not disrupting end users.

Breaking a monolith into modular components, adopting new architectures and updating no longer supported language versions gives an organization new freedoms to rethink how they want applications to support the business. The business can move faster and evolve its applications at the pace it needs to adapt to changing market conditions. Application modernization makes this possible.

AI-assisted software modernization is a must have

Artificial intelligence (AI) and especially in its latest technological leap, generative AI and large language models (LLMs), is having a huge disrupting impact across many industries including software engineering. Application modernization embracing AI can bring multiple benefits which together have a significant impact in reducing the challenge, risk, and cost of such activity.

Knowledge retention and dissemination

Using techniques such as retrieval augmented generation (RAG) it is possible to augment an LLM model with private data, so the codebase, code documentation, engineering communications, and project documents can all be brought within the scope of the LLM's knowledge base and allows the LLM to respond to queries relating to the specific details of the computing infrastructure of the organization.

A hazard for enterprises is when a veteran software engineer retires and walks out the door with essential knowledge that has not been preserved. Using LLMs augmented with private data corpuses it is possible to mitigate against loss of knowledge.

Omdia commissioned research, sponsored by IBM

Onboarding new members to a development team has an impact on the time of the existing team members who are focused on their job. This is the syndrome Fred Brooks Jr talked about in his classic book Mythical Man-Month, that productivity drops as you grow the size of a team. This can be mitigated with microservices architecture where small teams focus on a single business function and decouples teams, allowing the number of small teams to expand with demand. The role of AI can be significant here, essentially acting as a mentor to a new team member and reducing the teaching burden on other developers.

AI Automation

AI-based software development tools can introduce automation that considerably eases the workload of developers. No-code low-code capability can be introduced into the developer's preferred IDE, open files are observed by the AI and when the developer queries the AI through the chat interface these files provide the context for the AI to respond.

Java code migration from old to new versions is a typical task that the AI-assistant in the IDE can perform. Transformation between different programming languages is another task the AI can perform. Modernizing mainframe applications, for example, may involve a host of specialist languages: COBOL, PL/I, REXX, RPG, Fortran and more, and AI can help migrate applications out of these languages into a modern language. AI can provide understanding into the function these different applications perform and provide insight into the application landscape.

Adopting a modern architecture like cloud native computing requires considerable skills and experience, which an enterprise may lack in-house. This is where AI-assistance can provide the needed guidance with code suggestions, working alongside the developer as they write code in the IDE.

IBM internal case study: Java modernization

IBM needed to modernize a 20-year-old financial system written in Java 8, which was accruing a backlog of enhancements and technical debt – modernization was essential. IBM used this internal requirement to test out its AI-assisted software development portfolio of tools, performing a variety of tasks: automated code fixes, discovering code dependencies, and transitioning to cloud native architecture. The result was cutting a typical 12-week modernization effort (without AI-assistance) to just three weeks with AI tools. The developers saved time in application discovery and in code and configuration remediation.

Concluding thoughts

Modernization is one of the most difficult and postponed activities in the IT department, and AI can change this dynamic, making the activity faster, delivering value to the business quicker. The first step in modernization is to understand the code, a task made difficult as often the modernizers are not the creators of the application and code documentation may not exist. AI can help provide valuable intelligence into code component function, and map how components relate. Developers can interrogate the AI for specific information as they work in their preferred IDE.

AI-assisted software development helps improve code quality by acting as a second pair of eyes to scrutinize code for potential issues, and helps developers throughout the software development lifecycle from application creation and deployment, to production maintenance and upgrades.

Appendix

Further reading

Omdia Universe: No-Low-Pro IDE Assistants, 2025 (May 2025)

Michael Azoff, Chief Analyst, Cloud & Data Center Practice
askananalyst@omdia.com

Omdia consulting

Omdia is a market-leading data, research, and consulting business focused on helping digital service providers, technology companies, and enterprise decision makers thrive in the connected digital economy. Through our global base of analysts, we offer expert analysis and strategic insight across the IT, telecoms, and media industries.

We create business advantage for our customers by providing actionable insight to support business planning, product development, and go-to-market initiatives.

Our unique combination of authoritative data, market analysis, and vertical industry expertise is designed to empower decision-making, helping our clients profit from new technologies and capitalize on evolving business models.

Omdia is part of Informa TechTarget, a B2B information services business serving the technology, media, and telecoms sector. The Informa group is listed on the London Stock Exchange.

We hope that this analysis will help you make informed and imaginative business decisions. If you have further requirements, Omdia's consulting team may be able to help your company identify future trends and opportunities.

Get in touch

www.omdia.com
askananalyst@omdia.com

© 2025 TechTarget, Inc. All rights reserved. Unauthorized reproduction prohibited.