publicis
sapient

An Executive's Guide

Evaluating AI Platforms for Software Development

A practical framework for separating context-aware AI platforms from developer-focused platforms for software development

In this article

01 Executive summary

02 Executive clarity lags AI adoption

03 Why productivity claims fall short

04 Why AI platforms confuse buyers

05 Why enterprises stall even with platforms

06 A framework for evaluating AI platforms

07 What context-aware platforms enable

08 Proof of platform-level impact

09 How executives should choose

Executive Summary

For most enterprises, AI is now part of how software gets built and shipped. What remains unclear for many enterprise leaders is what to buy.

The enterprise software market is crowded with AI platforms that promise dramatic productivity gains. Most stop at coding assistance, rather than delivering agentic workflows across the full software development lifecycle with persistent enterprise context and built-in governance.

This guide offers an executive-level framework for evaluating AI platforms for software development, with a focus on long-term modernization.
It helps CIOs, CTOs and transformation leaders separate platforms that compound value across the enterprise from point tools that deliver short-term task gains but will need to be replaced as architectures evolve.

The gap between AI adoption and executive understanding

Across industries, enterprise leaders broadly agree on a few realities:

AI is critical to the future of software delivery.

Legacy modernization can no longer wait.

Budgets, risk tolerance and regulatory pressure are tightening.

And yet, despite widespread experimentation, many AI initiatives stall before delivering meaningful transformation.

The underlying issue is not a lack of ambition. It is that AI adoption has outpaced executive clarity. Most tools in the market lead with developer productivity claims, focus narrowly on coding acceleration and downplay or ignore enterprise realities like fragile legacy code, governance, security and compliance.

The result is predictable. Leaders struggle to evaluate their options, and investments that look promising in pilots fail to deliver at enterprise scale.

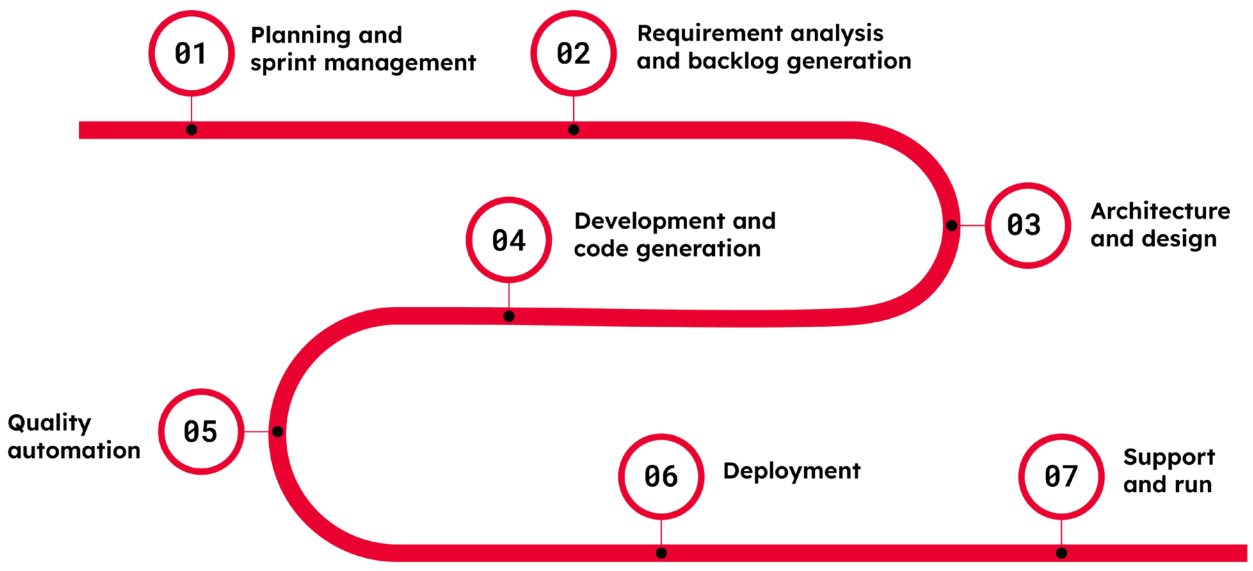
Why productivity claims are no longer enough

Nearly every AI development tool claims productivity gains. Very few explain where those gains actually show up, and where they disappear.

In enterprise software delivery, coding is just one part of the lifecycle. The largest potential production delays often occur later, during testing, integration, validation and release. When AI-enabled coding speed is introduced to this lifecycle independently, it tends to increase downstream risk and rework rather than reduce it.

Productivity gains that focus on coding alone will shift bottlenecks, not remove them. The code gets written faster, but other parts of the lifecycle—validation, compliance and release—get slower.

For enterprise leaders, faster developers won't lead to better outcomes. A platform that can accelerate the entire lifecycle will.


01 Planning and sprint management
02 Requirement analysis and backlog generation
03 Architecture and design
04 Development and code generation
05 Quality automation
06 Deployment
07 Support and run

Why AI platforms confuse buyers

Much of today's confusion comes from how AI tools and platforms are labeled. Many tools that are marketed as agentic development platforms promise more reliability than they actually provide.

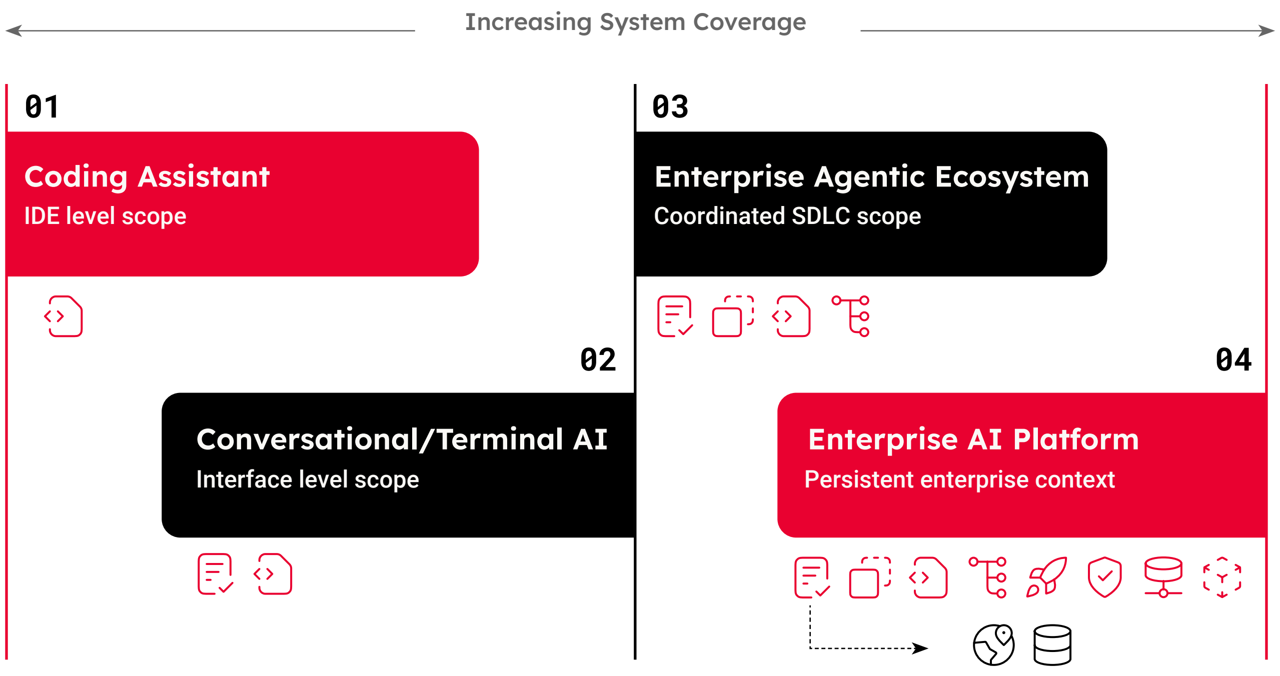
To evaluate these platforms effectively, leaders need a clearer distinction between multi-task AI coding platforms, and business context-aware AI platforms for software development.

AI coding platforms are designed to accelerate development tasks—providing code generation, debugging help and context-aware suggestions. Many operate inside IDEs, chat interfaces or terminals and optimize developers' immediate workflows.

Enterprise offerings such as GitHub Copilot Enterprise (with agent frameworks like Agent HQ) and next-generation coding AI like Claude Code and coordinated agent systems extend capabilities to longer-running, multi-step workflows. They increasingly support broader context retention, task automation and integration with enterprise governance, auditability and compliance controls.

While these tools promise a broad scope, in reality, their value is much more limited.

AI-assisted software development


Increasing System Coverage
01
Coding Assistant
IDE level scope
02
Conversational/Terminal AI
Interface level scope
03
Enterprise Agentic Ecosystem
Coordinated SDLC scope
04
Enterprise AI Platform
Persistent enterprise context

Four major categories in the AI-powered coding ecosystem

Category	Key Trait	Context Scope	Autonomy	Examples
Coding Assistants	Predictive help	Short-lived file/ inline	Low	Copilot, Code Whisperer
Conversational/ Terminal AI	CLI/Coding commands	Multi-file/ project	Medium	Claude Code CLI, Codex CLI
Enterprise Agent Ecosystem	Goal-driven agents	Cross-task and lifecycle	High + Software Context	GitHub Agent HQ, Codex app
Enterprise AI Platforms	Persistent enterprise memory	Organization-wide	High + Business and Context	Sapient Slingshot*

*Publicis Sapient's enterprise AI software development platform

Enterprise AI platforms for software development operate at a different level. They maintain enterprise and business context over time, using an enterprise context graph. They coordinate work across teams, tools, AI agents and stages. Governance, validation and traceability are built into the workflow rather than added later. Most importantly, they are designed to connect systems with business rules, not just software.

The practical difference matters. Assistants, tools and platforms help teams work faster inside existing systems. Platforms with business context change how the system itself works. They reduce risk, support on-time releases and enable repeatable modernization. Platforms do not just improve software delivery today. They expand what the organization can safely attempt in the future.

This distinction is rarely made explicit in the market, but it is essential for executive decision-making.

> **"Sapient Slingshot is our AI platform for automating and accelerating the entire software development lifecycle (SDLC). With Slingshot, enterprises can modernize legacy code, build and launch new software and transform how they work."**

Why enterprises stall when they choose the wrong platform

Developer workflow platforms can deliver real value but only within short-term timeframes and narrow boundaries.

In enterprise environments, they often:

- Accelerate codecreation without systematically discovering or preserving undocumented business logic
- Shift risk into testing, compliance and release phases rather than addressing it earlier in the lifecycle
- Introduce greater variability in standards and quality unless tightly governed through downstream controls
- Struggle with deeply entrenched legacy systems where business rules are implicit and dependencies are complex

The outcome is consistent across organizations. Teams move faster early in the lifecycle, then slow dramatically when systems must be validated, governed and modernized at scale. What looks like speed at the front of the funnel becomes friction at the back.

Enterprise software delivery is not a collection of agents and tasks. It is a system of interconnected work and AI agents that includes strategy and requirements, architecture and design, code and testing, and release, monitoring and change. Optimizing one step or one category of steps in isolation rarely improves the whole. Across large enterprises, the data is consistent. Less than half of productivity gains come from developer coding alone.

A practical framework for evaluating AI software development platforms

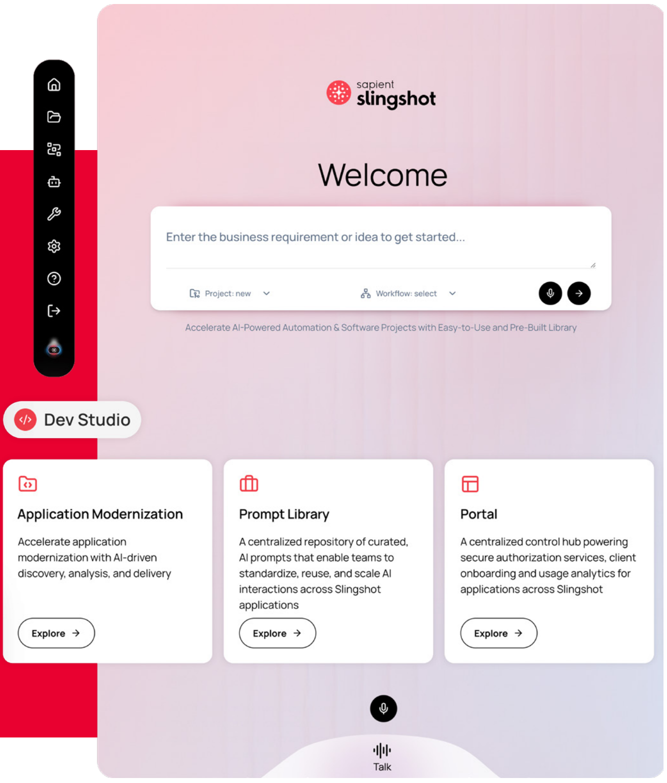
To distinguish developer-focused platforms from context-aware platforms, leaders should evaluate AI solutions across a small set of concrete dimensions:

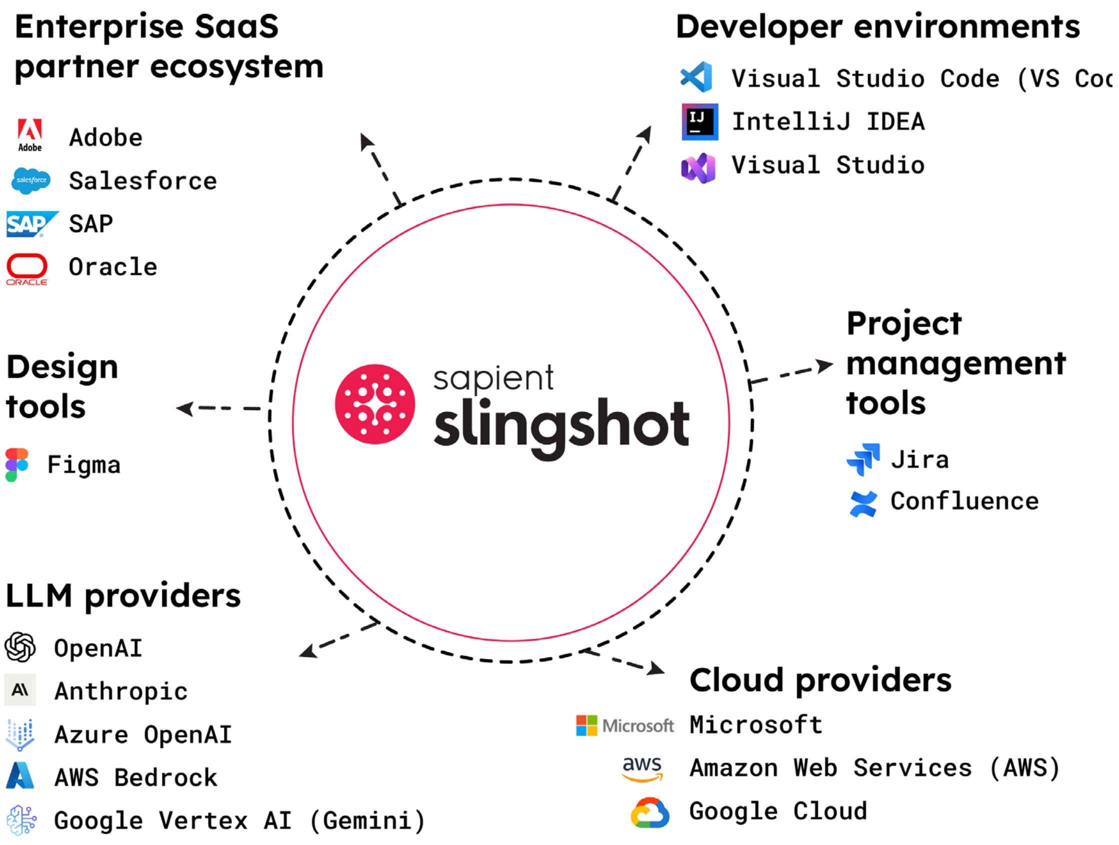
- ✓ **End-to-end lifecycle ownership.** Does the solution support planning, design, testing and deployment, or does it stop at coding?
- ✓ **Persistent enterprise software context.** Does it maintain business rules, system logic and architectural intent across teams, AI agents and time, or does context reset with every interaction?
- ✓ **Built-in governance and risk containment.** Are explainability, validation and human oversight embedded in the workflow, or bolted on after the fact?
- ✓ **Proven legacy modernization depth.** Can the system work with decades-old codebases, undocumented logic and complex dependencies, or is it limited to greenfield development?
- ✓ **Enterprise-native SDLC integration.** Does it integrate with existing SDLC tools like Jira, GitHub and Azure DevOps, or does it require wholesale replacement?

Solutions that perform well across all five behave like platforms. Those that do not are tools, regardless of how they are positioned.

The majority comes from reducing friction across planning → testing → integration → release.

Sapient Slingshot connects developer tools, cloud platforms and core business systems into one execution layer, letting enterprises modernize legacy technology and ship software faster without replacing any systems that keep the business running.


sapient slingshot
Welcome
Enter the business requirement or idea to get started...
Project: new
Workflow: select
Accelerate AI-Powered Automation & Software Projects with Easy-to-Use and Pre-Built Library
Dev Studio
Application Modernization
Accelerate application modernization with AI-driven discovery, analysis, and delivery
Explore
Prompt Library
A centralized repository of curated, AI prompts that enable teams to standardize, reuse, and scale AI interactions across Slingshot applications
Explore
Portal
A centralized control hub powering secure authorization services, client onboarding and usage analytics for applications across Slingshot
Explore
Talk

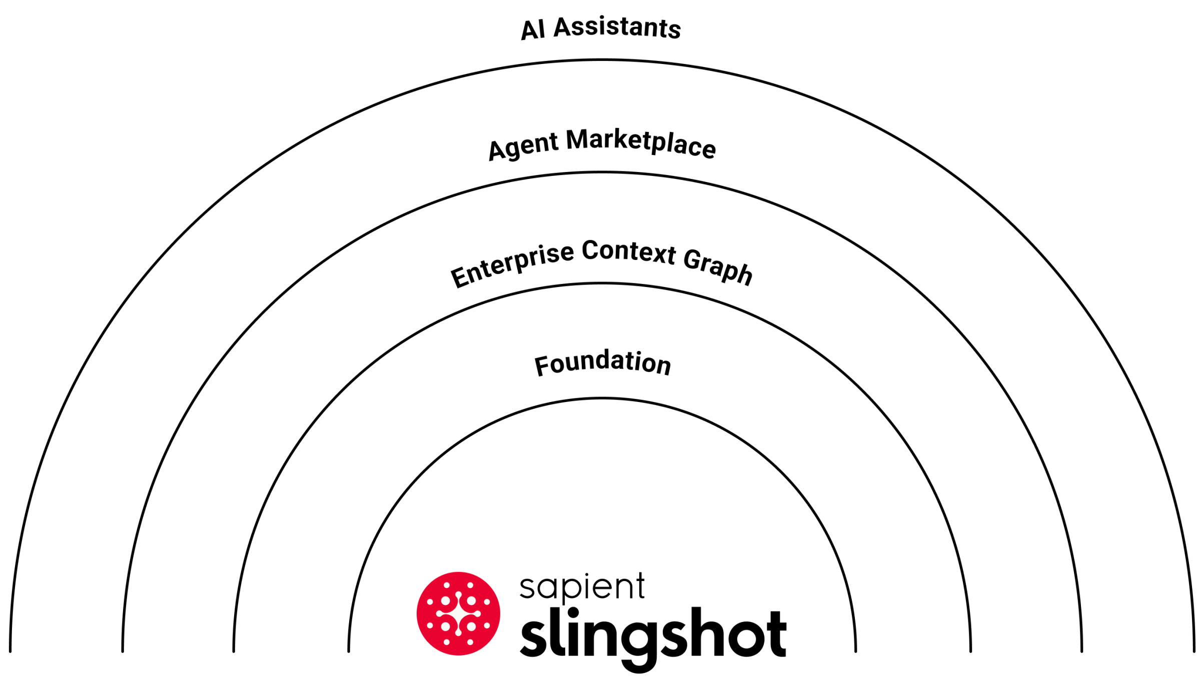


What context-aware enterprise AI platforms enable at scale

When AI is embedded as a context-aware system across the software development lifecycle, teams gain the ability to understand software as a living system—continuously understanding, changing, validating and delivering it at scale.

At this level, AI does not simply assist individual processes or tasks. Specialized agents can work together across discovery, specification → design → development → testing → release, while also carrying enterprise context forward. The result is sustained throughput, higher confidence in change and software delivery that connects with the broader business.

These outcomes are not achievable through coding acceleration or agentic software development workflows alone. **They require persistent context, lifecycle-wide orchestration and built-in governance so that speed, quality and compliance improve together rather than trading off against one another.**



Platform-level impact in practice

The following cases illustrate how context-aware platforms play out in practice across enterprise health care and energy industry software development.

Regional U.S. health system

A large regional U.S. health system relied on its public digital platform as a critical access point for patient care in a tightly regulated environment. Years of accumulated content, legacy CMS constraints and clinical integrations made changes slow and risky for a small digital team, limiting throughput rather than ambition.

Using Sapient Slingshot as an enterprise AI software development platform, the organization applied agents across content migration, component restructuring, integration mapping and validation using enterprise context across the system. This enabled the migration and re-authoring of more than 4,500 pages into a modular, headless architecture and the safe integration of real-time clinical data. More importantly, Slingshot established standardized, repeatable workflows that allow ongoing digital change to be produced continuously rather than rebuilt project by project. The result was a digital factory foundation for patient-facing software, not a one-time modernization effort.

Large European energy producer

A large European energy producer depended on a mission-critical application used to manage power plant infrastructure, but the system was over two decades old, undocumented and impossible to maintain safely. The risk was not development speed, but the inability to understand, govern or reproduce changes across similar assets.

Sapient Slingshot, Publicis Sapient's enterprise AI software development platform, was used to orchestrate agents across decompilation, refactoring, business logic extraction, documentation generation, testing and validation in a single coordinated workflow. This allowed the application to be revived in two days, with clean, modern code, full documentation, and documented time savings through automated code generation, test creation and validation. But beyond restoring one system, Slingshot converted a black box into a clear, documented and understandable application, that could connect to additional applications and sites. In effect, the organization moved from ad-hoc rescue work to a digital factory model for context-aware software development.

In both cases, the outcomes were not the result of faster development alone. Coding acceleration on its own would not have shortened a decade-long roadmap or eliminated security findings at scale. What made the difference was a coordinated approach that treated software development as an interconnected system rather than a series of independent tasks.

How executives should choose an AI software development platform

AI will continue to reshape software development. The differentiator will not be who adopts AI first. It will be who evaluates technology correctly.

Enterprises that invest in faster software development may move faster right now. Enterprises that invest in context-aware platforms built for end-to-end modernization will move faster in ways that are safer, repeatable and scalable.

This guide is intended as a starting point for making that distinction and for turning AI investment into lasting enterprise impact.

Learn more

Contributors

Jason Steiner
Senior Managing Director, AI & Technology Strategy Global Lead