

Intelligent applications, faster development

How GenAI is transforming
modern software engineering



Atos

Executive Summary

Generative AI (GenAI) is rapidly evolving from an experimental tool into a decisive competitive factor in enterprise IT, with an enormous economic potential: according to McKinsey [1], large language and multimodal models can generate up to 4.4 trillion USD in value annually. In practice, GenAI is already becoming a productivity engine, not only for standard software but increasingly for custom-developed business applications.

GenAI is delivering tangible efficiency gains across the software development lifecycle. Developers complete individual tasks up to 55% faster [2], repetitive activities are automated, prototyping accelerates, and knowledge is sustainably embedded through automated documentation. AI-based code reviewers strengthen quality assurance, reduce defects, and support faster onboarding of new team members.

At the same time, integrating GenAI capabilities into business applications unlocks new potential for usability, data quality, and process automation. Conversational input, intelligent form processing, automated validations, and natural-language reporting can fundamentally simplify existing interactions. This is particularly valuable in environments where users work daily with complex systems that require significant time and attention. With the help of cross-application analytics tools, organizations can pinpoint and address exactly those friction points in the application landscape where GenAI promises the greatest optimization potential.

The introduction of generative functionality can often be achieved efficiently and with manageable effort, for example through text suggestions, conversational interfaces, document extraction, or context-aware in-app assistance.

Many organizations begin with isolated pilot projects. However, to ensure long-term scalability, governance, and reusability, it is advisable to establish a central GenAI platform architecture that provides generative intelligence as a reusable service. This enables consistent knowledge access, standardized security, and flexible integration – independent of systems or business domains.

While leading SaaS providers such as Microsoft, SAP, and Salesforce now embed AI assistants into their platforms, it is ultimately the responsibility of the organizations that are developing custom enterprise software to strategically anchor GenAI themselves. Those who fail to act early risk seeing their core business processes overtaken by competitors, not because these competitors possess superior domain expertise, but because they can deliver modern user experiences, automated workflows, and adaptive logic more quickly.

The result is a steadily widening innovation gap.

As a strategic partner, Atos supports organizations throughout their AI-transformation journey – from use-case analysis and architectural design to technology integration and secure, stable operations. In this way, GenAI becomes not only an efficiency driver, but a sustainable enabler of innovation and digital competitiveness.

In this whitepaper, we demonstrate how organizations can effectively incorporate generative AI into their custom-built applications and development processes to maximize value, regardless of industry, technology stack, or operating model. We outline practical approaches for realizing potential, addressing compliance challenges, mitigating risks, and evolving existing systems in a meaningful and future-proof manner.



Accelerating the software development process

Increasing productivity: From idea to code

The most visible impact of GenAI in software development is the massive acceleration of routine tasks. Where developers previously had to search through documentation, build repetitive boilerplate structures, or write simple code fragments manually, AI-powered tools now provide actionable suggestions within seconds. This shift not only speeds up individual tasks but changes the overall cadence of software projects.

Studies already demonstrate impressive productivity gains: a GitHub Copilot study found that developers required on average 55% less time [2] to complete a specific task when assisted by AI. This effect is particularly strong for common frameworks, repetitive tasks, or structured requirements – areas where developers traditionally spent significant time on onboarding or copy-and-paste patterns.

The path from ideation to creating code becomes more direct: requirements can be formulated in natural language and translated directly into executable code suggestions. What once required multiple steps – specification, review, and implementation – can now be streamlined seamlessly with drastically shortened feedback cycles. Even non-developers such as business analysts or product owners can contribute to early prototypes without deep technical knowledge.

However, using GenAI for code generation introduces several important considerations which ought to be addressed early on:



Tool and platform selection: Whether GitHub Copilot, Claude Code, Cursor, Amazon CodeWhisperer, JetBrains AI Assistant, or self-hosted solutions—organizations must make conscious decisions about which tools integrate seamlessly into existing development environments, meet required data protection standards, and provide sufficient flexibility to adapt to their specific needs.



Data protection: Many cloud-based GenAI services like GitHub Copilot, Amazon CodeWhisperer, or Azure OpenAI process submitted prompts and generated code on external servers, in some cases outside the EU. While certain enterprise plans offer options for data localization, uncontrolled use still carries the risk that sensitive information, such as source code, business logic, or access credentials may inadvertently be transmitted to and temporarily stored by cloud providers.



Organizations must therefore implement both technical and organizational safeguards to ensure compliance with GDPR and internal policies, and to give employees clarity regarding which GenAI tools they are permitted to use.

Intellectual property: An AI-generated source code is generally considered the intellectual property of the individual or organization generating it. However, risks can arise if the generated code closely resembles existing open-source components protected under copyleft licenses that may have been included in the model's training data. An ongoing class-action lawsuit in the United States targets GitHub Copilot for potential violations of open-source licenses such as MIT, GPL, and Apache. The EU AI Act further obligates model providers to respect intellectual property rights – backed by potentially significant penalties for non-compliance. In practice, it is nearly impossible for organizations to determine the exact training data on which a given AI suggestion is based. As a result, an individual risk assessment of IP exposure should always be performed when using GenAI in software development.



Quality control and testing: Not every AI-generated suggestion is correct, performant, or secure. Without systematic review mechanisms, flawed code may unknowingly enter production systems. Confidence in automated suggestions must therefore be supported through automated tests wherever possible, complemented by guidelines and manual reviews.



Shift in skills within the team: Developers need to boost their skillsets for effective prompt design, critical evaluation of AI outputs, and the ability to integrate generated code into existing architectures. Without targeted training, the organization risks achieving mere automation instead of genuine, sustainable quality improvements.



Technical dependencies and vendor lock-in risks: Several tools are built on proprietary models or cloud platforms. Organizations that commit too strongly to a single provider too early risk facing high switching costs or reduced flexibility in later phases, particularly in areas related to security and governance.

Reducing errors: Ensuring quality through AI

Modern AI-powered code reviewers such as DeepCode or Copilot PR Bots analyze requests in real time, propose patches, and highlight security vulnerabilities before they reach the main branch. IBM reports that teams using automated reviews experience significantly fewer critical defects in staging builds [3]. A GitHub study further concludes that code produced with Copilot tends to be more functional, more readable, and more maintainable than code written entirely by hand [4]. More recently, autonomous bug-fix agents have emerged that can execute regression tests independently and even generate hotfixes, further reducing mean time to repair (MTTR) [5].

Anchoring knowledge: Maintaining a team-independent code flow

Generative models automatically produce concise method comments, UML sketches, or architectural descriptions for existing sections of a codebase. A 2024 research study indicates that up to 70% of documentation gaps in legacy repositories can be closed using LLMs—without any loss of technical accuracy during peer review [6]. As a result, valuable domain knowledge no longer resides solely in people's minds or scattered across chat threads; instead, it becomes a machine-readable artifact embedded directly into the code history.

Enabling innovation: Shifting focus toward creative work

As routine tasks disappear, developers can redirect time previously spent on routine tasks like typing and searching toward architectural decisions, usability testing, and experimentation with new features. A six-month cross-industry study of more than 7,000 knowledge workers found that developers spent 20% more time on conceptual and creative activities once GenAI assistants began handling standard tasks [7]. In this way, AI becomes a driver of efficiency as well as a catalyst for innovation, creating the mental bandwidth necessary for entirely new user journeys or business models to emerge.

Improving scalability: Bringing new teams up to speed faster

Onboarding guides, interactive code explanations, and automated “how-to-fix” prompts have been shown to significantly reduce the ramp-up time for new team members. GitHub reports that enterprise teams experience up to 30% faster onboarding for junior developers when leveraging Copilot-based learning paths [8]. This makes it easier for staff growing projects with distributed teams without sacrificing development velocity or code consistency.



Delivering excellence by integrating GenAI into enterprise applications

After outlining how GenAI is transforming the software development process itself – through automated coding, test generation, or semantic documentation – the next question arises: How can the same intelligence become an integral part of the applications used daily in operational business?

True value creation does not stem from faster code alone, but from intelligent functionality embedded directly into the finished product. The goal is to make GenAI available precisely where users interact with systems, make decisions, or enter data. This is where the greatest untapped potential lies in many organizations. Existing applications – whether custom-built, integrated, or extended – can gain substantial improvements in efficiency, clarity, and user experience through targeted GenAI capabilities.

Integrating GenAI into enterprise applications, however, is not a copy-and-paste exercise. It requires deep understanding of each system's architecture, user context, and business processes, combined with the ability to operationalize generative models in a controlled, secure, and economically viable manner. The key is to deploy AI where real friction exists and where meaningful, measurable value can be unlocked.

Ideation and use case prioritization in the application landscape

The integration of AI into existing application landscapes does not begin with technology; it begins with the right focus. Not every application, not every process, and not every user interface benefits equally from AI capabilities. A critical success factor therefore lies in the structured selection and prioritization of use cases, based on value, feasibility, and strategic relevance.

Especially in heterogeneous application landscapes shaped by years of individual development, the first step is to gain clarity:

Where are manual efforts required today?

Which complex interactions may be simplified through natural language, contextual understanding, or automation?

Which systems are closely intertwined with the business domains? Can they generate tangible value by embedding AI-driven intelligence directly into the user interface?

For large web-based application landscapes, Atos recommends that clients use browser-based usage analytics tools as one building block of this assessment process. These tools not only provide a consolidated view of the entire application landscape, but also enable the capture of anonymized interaction patterns in the frontend, such as mouse movements, scrolling behavior, click paths, and especially the time users spend in specific forms and screens.

This makes it possible to reveal critical hotspots:

Which applications and forms do employees or users spend the most time on? Where do users abandon processes? Which fields cause uncertainty?

Which part of the user journey demands particularly high manual efforts?

These insights help identify areas where the integration of GenAI delivers the greatest value – whether through auto-completion, context-sensitive assistance, or even fully conversational interfaces.

A proven starting point is the combination of two perspectives:

- Bottom-up: Such analyses help identify inefficient workflows, overloaded forms, and frequent repetitions that can be optimized with minimal effort through AI support.
- Top-down: The focus shifts to the organization's overarching goals, such as improving data quality, reducing throughput times, or enhancing the user experience, and derives cross-functional opportunity areas for GenAI.

For an initial assessment, a simple framework can be applied:

- Value contribution: What savings or improvement potential does the use case offer in terms of time, cost, satisfaction and/or quality?
- Costs: What operating costs arise from using GenAI particularly when relying on cloud-based GenAI providers? And does the value contribution justify these additional expenses?
- Technical feasibility: Can the use case be implemented with technologies available in the market today?
- Data foundation: Are there structured data or meaningful contextual signals that the AI can effectively leverage?
- Risk and compliance: Which regulatory, security-related, or data-protection requirements must be considered?

This step unlocks a particular opportunity for custom-built applications. Unlike standard software, GenAI can be tailored precisely to the organization's processes, data models, and user needs. This does not require a full redevelopment effort, but rather targeted augmentation, manageable in scope yet offering substantial value potential.

The ideation phase therefore serves not only as a mechanism for generating ideas, but as the foundation for economically viable AI adoption. It ensures GenAI is not introduced out of opportunism or as an experimental novelty, but as a productive, value-creating element of the software landscape, guided by clear benefits, technical feasibility, and strategic intent.

Approaches for integrating GenAI into existing business applications

Once the critical pain points within the application landscape have been identified, the next step is to unlock value potential as efficiently as possible. GenAI is particularly well-suited to generating quick wins, as existing large language models (LLMs) already deliver results of such high quality that dedicated model training

is often unnecessary, and a value-creating functionality can be implemented with comparatively low investment.

Following the systematic identification of potential application areas, the natural next question concerns practical implementation: Where can GenAI be integrated into existing systems with reasonable effort to deliver tangible efficiency gains?

This is especially relevant for custom-built systems that do not automatically benefit from the AI feature enhancements offered by major SaaS platforms.

The goal is not full-scale redesigns, but targeted, impact-oriented enhancements that simplify existing interactions, reduce manual effort, or improve data quality. Frequently used forms, recurring text inputs, complex parameter configurations, or error-prone validation steps offer particularly strong starting points for this kind of augmentation.



Approaches for GenAI integration into existing business applications

Area	End-user benefits	Typical implementation	Practical example
Intelligent data capture	Fewer typos, faster completion	LLM suggests field values ("Intelli Fill") or pre-populates them based on context (e.g., last order)	Power Apps Copilot offers up to ten AI generated field suggestions instead of default column proposals.
Conversational forms	Users type or speak freely ("I need to create a new invoice for UK group")	Chat widget calls LLM API and generates JSON payload for the backend	SAP Joule and Salesforce Einstein provide copilots that fully orchestrate digital input and output.
Adaptive / dynamic forms	Only relevant fields are shown; fewer drop offs	Prompt: "Which fields do I need to onboard a supplier?" – LLM generates the schema on the fly	Copilot preview in Power Apps generates entire screens from a single natural language description.
Smart validation and coaching	Human like error messages ("Postal code does not match country—please verify")	LLM evaluates rules, explains them, or suggests valid values	Same mechanism as in §1 with additional "reason explain" prompting
NL Reporting and search	"Show me all orders > €50k in Q1" instead of building pivots manually	LLM translates question into query/DSL, calls BI API, summarizes results	Einstein Copilot answers CRM-related queries in natural language.
Document- to form extraction	PDFs, emails, or images are automatically converted into structured fields	Multimodal model (vision + text) generates JSON; human in the loop acts as review stage	Applicable with GPT 4o or specialized vision models
Workflow and rule generation	Business users describe processes in everyday language; AI builds flows and validations	RAG combination using policy documents and an LLM; output as BPMN / JSON definitions	Joule Studio generates "skills" for SAP Build workflows.
Context aware help and onboarding	Inline explanations, best practice examples, micro learning	RAG pipeline: UI context --> vector search --> LLM answer	All major copilots provide "What does this field mean?" style inline assistance.
Semantic search and integrations	Unified search across CRM, ERP, DMS	Embeddings + LLM ranking	Joule federates queries across SAP and non SAP systems.

Quick wins for existing form-based applications

The following table shows that implementing such innovative approaches can be realized as a quick win even with manageable effort, using example values for the AI optimization of a simple, existing form-based application.

Effort	Measure	Outcome
1-2 days	Integrate AI text suggestions for free-text fields via Azure/OpenAI API	20–30% reduction in typing effort
1 week	Add a conversational front end (“chat input”) as an optional tab	Higher user acceptance, especially among mobile users
2-4 weeks	Implement rule checks + natural-language error messages	Fewer help-desk tickets
~6 weeks	NL reporting (“Ask your system”) + self-service dashboards	Business units reduce BI queries

From isolated solutions to a platform strategy: GenAI as an enterprise-wide enabler

The initial adoption of GenAI in organizations often begins pragmatically – through standalone chatbots, text generators, or pilot features embedded within a specific application. These local initiatives generate valuable insights and visible success, yet they typically remain technically and organizationally isolated.

As the number of such AI islands grows, so do the challenges: redundant integrations, fragmented data flows, inconsistent security concepts, and missing governance frameworks all contribute to rising complexity and increasing maintenance effort.

The central question therefore becomes: **How can GenAI be conceived and leveraged at an enterprise scale without becoming trapped in silo structures?**

The answer lies in a clear platform strategy. By establishing a centralized AI architecture that provides generative intelligence as a shared enabler across all applications, organizations can create a scalable, consistent, and future-proof foundation. The goal is not to standardize the applications themselves, but to introduce an enterprise-wide AI layer capable of operating both linguistically and contextually while drawing on the entirety of relevant organizational knowledge.



This kind of a platform delivers three core advantages:

Knowledge consistency and context availability

Instead of connecting each application separately to models or knowledge sources, a central RAG architecture ensures the same information, policies, and terminology are used, regardless of where the GenAI functionality is invoked. This not only improves the substantive quality of responses but also ensures compliance, traceability, and consistent reasoning across the organization.

Efficiency and reusability

Prompt logic, access controls, validation patterns, and data integrations no longer need to be redeveloped for every individual application. A central platform acts as a modular toolkit that development teams can access quickly and securely, whether in the frontend (e.g., chat UIs or smart forms) or in the backend (e.g., document analysis or code generation).

Central governance, decentralized impact

Governance, monitoring, auditing, and cost management can be established centrally without limiting the flexibility of individual business units or teams. This creates a balance between standardization and freedom to innovate – a critical success factor for sustainable scaling.

Given these decisive advantages, the development of an **enterprise-wide GenAI or RAG platform** is becoming increasingly strategic [9]. It serves as both a technological and organizational bridge, linking a diverse landscape of individual software solutions with a unified layer of AI capabilities. Instead of anchoring generative intelligence in isolated applications, it is made available as a scalable, secure, and continuous learning service, with access to structured data, documents, policies, process knowledge, and contextual information from across the entire organization. This platform thus becomes not only the technical backbone for AI-supported interaction, but also the central knowledge hub that makes information discoverable, interpretable, and actionable.

At Atos, our [Polaris AI](#) provides a comprehensive solution for delivering enterprise-wide Generative AI in a secure, scalable, and impactful manner.

It is designed as a central GenAI and RAG platform that supplies custom-built applications, line-of-business systems, and existing IT infrastructures with generative intelligence. A particular focus lies on the controlled integration of enterprise knowledge. Polaris AI combines market-leading language models with powerful Retrieval-Augmented Generation (RAG) capabilities to make relevant data sources structured or unstructured, accessible in a context-aware fashion.

The platform provides core components such as **prompt governance, access and rights management, data-protection features, cost monitoring, and output validation**, while deliberately decoupling model logic from the application layer. The result is a reusable and robust foundation that can be integrated into a wide variety of use cases – from assistant-based UIs to semantic search and process automation.



How Atos is enabling GenAI transformation of application landscapes and development processes

The introduction and scaling of GenAI is not an isolated technology initiative. It requires an enterprise-wide engagement with architecture, processes, data availability, governance, IT security, and user adoption. The journey from initial use cases to a productively deployed, strategically anchored AI infrastructure places high demands on planning, implementation, and operational excellence.

As a globally leading IT service provider with extensive experience in developing, integrating, and operating complex digital platforms, Atos supports organizations throughout their entire AI transformation journey, modular or end-to-end, depending on their business needs. Our ambition is not merely to deliver technological solutions, but to enable sustainable value creation through responsible and economically viable AI adoption.

In doing so, we assume a variety of roles – from targeted support in specific sub-projects to full end-to-end delivery.



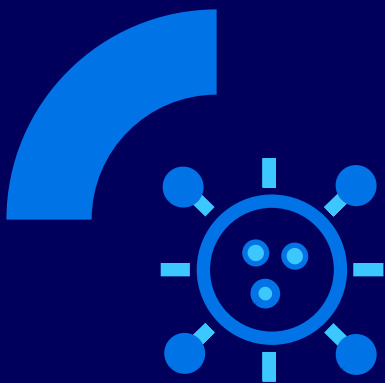
With [Atos Polaris AI](#), we additionally provide a powerful, modular GenAI platform that serves as the technological foundation for enterprise-wide use of GenAI. It consolidates essential capabilities such as knowledge integration, security and compliance components, and access to market-leading language models, enabling a unified, controlled, and future-proof AI integration across system boundaries.

Atos offers the expertise, experience, and infrastructure required to reliably turn AI visions into operational reality to support your business journey – from strategic initiation to technological implementation and operational management.

>> Learn more about how Atos Polaris AI is enabling the development, deployment and operational management of enterprise-grade autonomous AI agents: atos.net/en/services/data-and-ai/atos-polaris-ai

>> Explore how your organization can leverage Atos' applications offerings to transform into a future-fit digital enterprise: atos.net/application-services

>> Connect with us to explore how Atos can empower your business users with GenAI and agentic AI platforms to achieve tangible outcomes: atos.net/en/contact-us



References:

- [1] [Economic potential of generative AI | McKinsey](#)
- [2] [Research: quantifying GitHub Copilot's impact on developer productivity and happiness](#)
- [3] [AI code review - IBM](#)
- [4] [Does GitHub Copilot improve code quality? - The GitHub Blog](#)
- [5] [GitHub launches new AI coding agent that fixes bugs - The Economic Times](#)
- [6] [AI-Driven Automated Software Documentation Generation for Enhanced Development Productivity - ResearchGate](#)
- [7] [Shifting Work Patterns with Generative AI - arXiv](#)
- [8] [Training and onboarding developers on GitHub Copilot - GitHub](#)
- [9] [Gartner Experts Answer the Top Generative AI - Gartner](#)



Authors



Patrick Fink

Head of Innovation & AI in
Software Development, Germany

patrick.fink@atos.net



Rudy Cherian

Expert - Application Services
Digital Applications, Germany

rudy.cherian@atos.net

About Atos Group

Atos Group is a global leader in digital transformation with c. 63,000 employees and annual revenue of c. €8 billion, operating in 61 countries under two brands – Atos for services and Eviden for products. European number one in cybersecurity, cloud and high performance computing, Atos Group is committed to a secure and decarbonized future and provides tailored AI-powered, end-to-end solutions for all industries. Atos Group is the brand under which Atos SE (Societas Europaea) operates. Atos SE is listed on Euronext Paris.

The [purpose of Atos Group](#) is to help design the future of the information space. Its expertise and services support the development of knowledge, education and research in a multicultural approach and contribute to the development of scientific and technological excellence. Across the world, the Group enables its customers and employees, and members of societies at large to live, work and develop sustainably, in a safe and secure information space.

More information here:

atos.net

atos.net/career

Let's talk.



Atos is a registered trademark of Atos SE. February 2026. © Copyright 2026. Atos SE. Confidential Information owned by Atos group, to be used by the recipient only. This document, or any part of it, may not be reproduced, copied, circulated and/or distributed nor quoted without prior written approval of Atos.

The Atos logo, consisting of the word "Atos" in a bold, blue, sans-serif font. The letter 'o' is stylized with a white dot.