Infosys topaz

WHITE PAPER

AGENTS FOR EXPERIENCE: ELEVATING DEVELOPER EXPERIENCE WITH AGENTIC AI

Infosys
Navigate your next

Executive Summary

As software development transitions from traditional GenAI-driven life cycles to the emergent paradigm of Agentic Software Development Lifecycle (SDLC), developer experience (DevX) becomes the central axis around which innovation pivots. Agentic AI—characterized by its autonomy, adaptability, and proactive intelligence—does more than automate tasks; it redefines the software development process by anticipating developer needs, streamlining workflows, and reducing cognitive friction. This evolution ensures that developers are empowered not only by intelligent automation, but also by context-aware guidance, enhanced collaboration, and trustworthy guardrails for reliability and ethics. By placing DevX at the core, the Agentic SDLC promises to unlock new heights in productivity, satisfaction, and software quality, marking a transformative leap in how software is conceived, built, and maintained.

Introduction: The Imperative of Developer Experience in the AI Era

Software is evolving quickly, and the digital economy is experiencing increased demand, making Developer Experience (DevX) a strategic priority for organizations. In addition to and beyond operational efficiency, an excellent DevX environment is now seen to be a key driver of innovation, staff retention, and leadership in marketplace positioning. This section explains what DevX is, describes the current challenges that developers have to deal with, and presents Agentic AI as an innovative technology that will help mitigate such complexity.

1.1 Defining DevX: Beyond Productivity

DevX refers to the overall experience of the environment that developers work in, rather than mere measures of productivity. It encompasses their overall satisfaction, effectiveness, and well-being as professionals. Among the major characteristics of a positive DevX, it is possible to distinguish such qualities as continuous improvement via effective feedback loops, manageable cognitive load made possible by well-organized code and easily accessible documentation, and work environment.

Some of the key items required to make DevX robust are the facts that code-test/run cycles are as fast as possible, that the development environment is stable and consistent with all the tools needed and all the configurations present, and that elements like code generation, linting, and formatting are all streamlined and automated. Efficient code reviews, robust CI tools, and organized development environments are also crucial. Further, an easy access to help and clear ownership structure are major contributors to developer satisfaction. On the other hand, the signs of bad DevX are developers facing obsolete instruments, slow systems, non-optimal resources, and a hectic work environment, resulting in low motivation and interest.

The uniformity of positive DevX coupled with positive results like developer satisfaction, increased productivity, and better morale show that the process of adding value to DevX goes beyond enhanced operations, rather it has a direct effect on business performance. This also means that when developers are happier and more productive, the quality of software will be better, and it will be developed faster. This can be immediately converted into a competitive edge, with shorter time-to-market for new products and features, and eventually, long-term business growth. As such, a focus on DevX initiatives is not just a technical aspect but a powerful business strategy that enables innovation and market leadership.

1.2 The Evolving Landscape of Software Development: Challenges and Opportunities

Software development is an active and constantly evolving discipline, with developers facing growing system complexity, tight timelines, and the ever-present need to learn new technologies. One practical impediment to developer efficiency is inefficient documentation, with a big percentage of the developers (41%) stating it as one of the major barriers. To top this, almost a third of developers take an hour or more per day searching the Internet, resulting in a loss of valuable development time in the process. Other pervasive challenges include the complexity of meeting the ever-shifting demands, inefficient cooperation among a wide range of team members, and time consumption in the field of bug tracking, the big and complex systems, in particular.

Inefficiencies in workflow are very common, and are largely attributed to excessive meetings, with developers even spending as much as 30% of their time in meetings. Repetitive and manual processes like massive testing and extremely slow deployment procedures also affect productivity. Switching between multiple activities and different applications, combined with ineffective communication and lack of real-time access to information can result in increased operational costs, reduced efficiency, and lower software quality.

Notorious context switches, mental workload, and disparaging work causing context switches are identified as major inhibitors of productivity. Once burdened by routine work, constant interruptions, and the mental burden of working with complex and unconnected systems, the ability of developers to come up with creative solutions to problems and strategically think about them in novel ways is compromised. This suggests a straight causal connection: Mitigating cognitive load is not merely aiming at increasing efficiency; it requires the freeing of human potential to innovate, which is a key differentiator in today's competitive software environment.

1.3 Introducing Agentic AI: A New Paradigm for Automation and Intelligence

The definition of Agentic AI also implies a significant shift that has occurred in AI, in that, it no longer focuses on reactive rule-based automation, but instead, is goal-directed and autonomous. The key features that define these systems are the following: They are autonomous, meaning that they can operate and make their decisions independent of a constant human supervision; they are goal-driven, meaning that they follow the specific objective and decompose it into easier-to-manage sub-goals and optimize their actions with the purpose of accomplishing the goals; they have a strong reasoning/problem-solving capability, meaning that they can analyze complex problems and develop dynamic solutions to new data; and finally, they have the capability of adaptive learning, where their performance continuously advances as a result of feedback and new information. Moreover, Agentic AI is active, which means that it thinks ahead of time, predicts needs, and makes decisions instead of simply responding to orders.

The combination of Large Language Models (LLMs) to enable complex reasoning and communication and Large Action Models (LAMs) to execute capabilities, as well as Generative AI in content creation makes Agentic AI so powerful. It has the advantage of being dynamically adaptive to real-time information and has the capability to perform complex, multi-step processes that cannot be easily performed with the traditional and fixed-rule automation.

Another development in Agentic AI is the change of approach to develop toward a goal-oriented rather than a tool-oriented interface. Traditionally, a task was completed by a developer through manual selection and using different tools. However, Agentic AI also brings the paradigm of intent where the game has moved away from how to do or what it aims to accomplish. The agent will autonomously coordinate the needed tools and sub-tasks in a wholesome way and offer an important abstraction level, freeing the developers of the mechanics of low-level tools. Such a paradigm shift means that developers must now be more strategic in thinking about the targeted outcomes instead of the targeted actions through individual elements of a tool, and they are in need of more abstract knowledge regarding system design and problem decomposition.

2 Understanding Agentic AI: Foundations and Capabilities

In order to understand the transformative promise of Agentic AI on DevX, it is critical to consider its technical underpinnings, the diversity of forms that it can facilitate, and how agentic systems are architected, enabling autonomous and adaptive behaviors.

2.1 Core Principles of Agentic AI: Autonomy, Reasoning, and Adaptability

Agentic AI systems and offerings including Infosys Agentic Foundry, part of Infosys Topaz™, are designed to pursue specific goals, breaking down complex objectives into manageable tasks. They continuously optimize their actions and improve through adaptive learning, drawing on past experiences and advanced machine learning techniques. This allows them to grow more efficient and effective over time. Figure 1 outlines the core principles of Agentic AI.

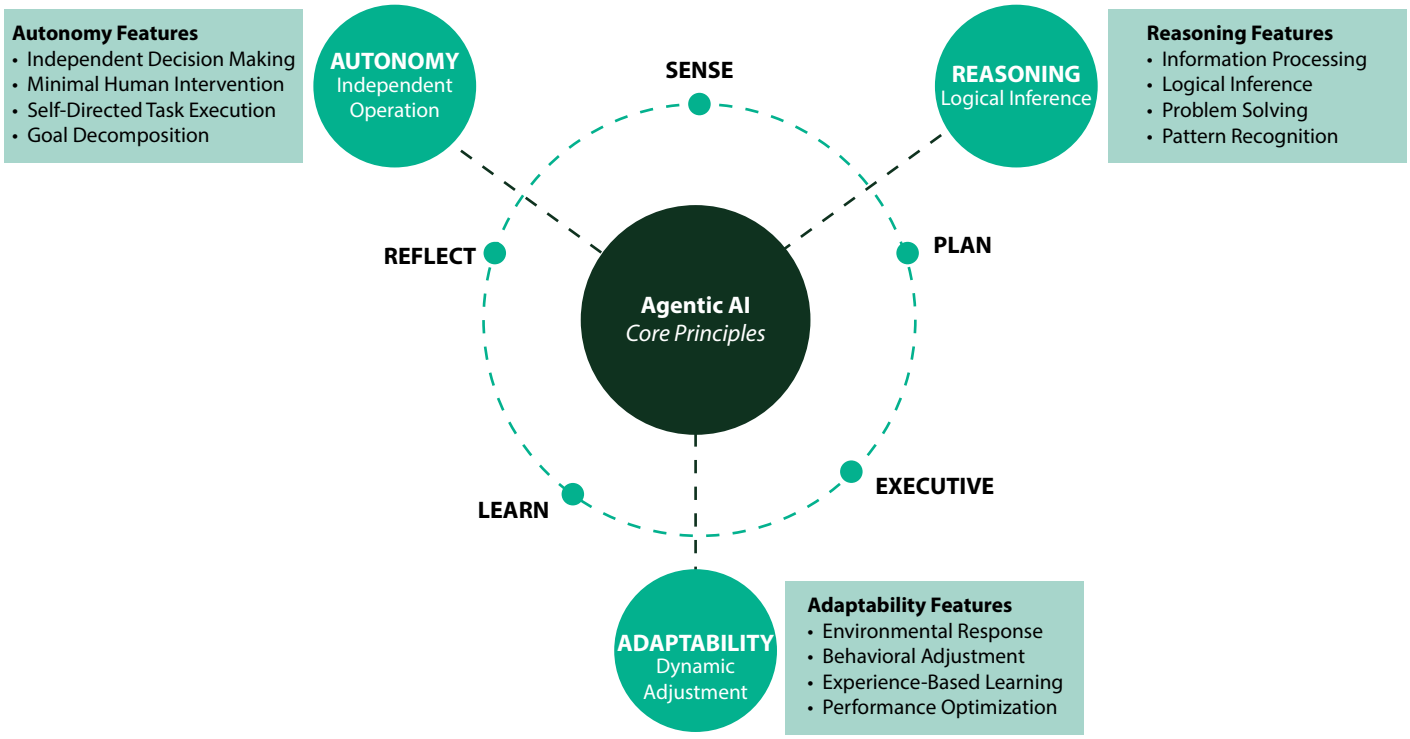


Fig 1. Agentic AI: Core Principles

Agentic AI operates through a continuous feedback loop of sensing, planning, acting, learning, and reflecting. This iterative process distinguishes it from static, rule-based systems. Effective deployment relies on strong data collection and monitoring, ensuring agents adapt to dynamic real-world situations.

2.2 Types of AI Agents and Their Roles in Software Systems

AI agents can be classified based on their level of intelligence and the sophistication of their decision-making processes. These classifications help identify their capabilities in different aspects of software development.

Agent Type	Key Features	Strengths	Limitations
Simple Reflex Agent	Responds to current perceptions using condition-action rules; no memory	Effective in structured, predictable environments	Lacks memory and long-term planning; struggles in dynamic scenarios
Model-Based Reflex Agent	Maintains internal model of environment; tracks current state	Handles partially observable environments; more informed decisions	Limited in planning and reasoning capabilities
Goal-Based Agent	Internal model plus explicit goals; uses planning and reasoning	Proactive, goal-oriented; greater foresight	Requires more computation; may be complex to design
Utility-Based Agent	Uses utility function to evaluate and maximize overall benefit	Optimal decision-making; balances competing objectives	Requires defining and computing utility; can be computationally intensive
Learning Agent	Adapts and improves via experience; includes learning and performance components	Flexible and improved over time; suited to complex, changing environments	May require significant time to learn; performance depends on learning process

Table 1: Types of AI Agent

On top of such intelligence-based categories, agents may also be classified according to the scope of operation:

Single Agent Systems: In this category, an agent works toward a given goal without controlling any other agent. These systems are most appropriate when it comes to clearly defined tasks without the need to cooperate with other AI agents.

Multi-Agent Systems (MAS): These involve multiple agents interacting to cooperate or compete in pursuit of goals. Each agent's unique roles enable MAS to tackle complex tasks, facilitating automation of intricate, multi-step processes. These systems often mimic human-like communication and collaboration.

Within the context of software development, specific types of agents are emerging to address distinct needs:

- **Coding Agents:** Generate, refactor, and modernize code, support collaborative and cross-language development, and transform legacy systems.
- **Testing Agents:** Create and execute test suites, perform regression and performance testing, automate test generation, and ensure code quality.
- **Code Review and Quality Agents:** Conduct static code analysis, enforce coding standards, scan for vulnerabilities, and maintain code best practices.
- **Infrastructure and DevOps Agents:** Automate deployment, manage CI/CD pipelines, monitor system health, optimize resources, and provide self-healing capabilities for infrastructure.
- **Project Management Agents:** Assist with planning, estimation, documentation, reporting, stakeholder communication, and risk assessment.
- **Specialized Domain Agents:** Focus on areas like security, database management, API integration, and compliance monitoring.
- **Emerging and Specialized Agents:** Support AI/ML development, cloud-native operations, and mobile/frontend development. Ensure accessibility and multi-cloud management.

As development shifts toward multi-agent systems, the focus moves from building one super agent to orchestrating specialized agents like code generation, testing, and security. The real advantage lies in their collaboration, requiring sophisticated orchestration beyond basic prompt engineering. This new landscape demands expertise in managing these diverse, coordinated agent ecosystems.

3 Agentic AI as a Catalyst for DevX

Agentic AI can be considered a robust driver of improving DevX in a direct and indirect way, positively influencing the current and historical developer pain points, substantially decreasing the overall cognitive load, and providing better feedback systems across the entire software development cycle.

3.1 Addressing Developer Pain Points with Intelligent Automation

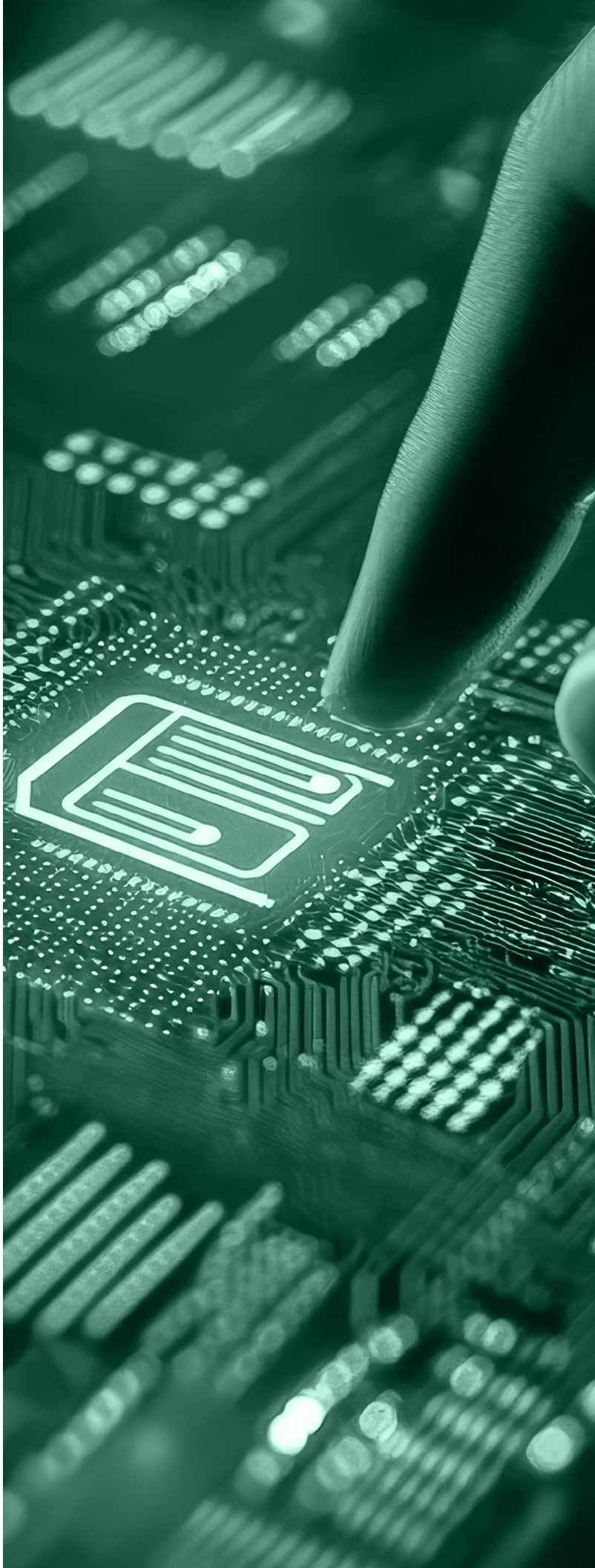
Agentic AI addresses longstanding developer challenges by automating complex, repetitive tasks and reducing cognitive load, resulting in more efficient and error-resistant workflows. Unlike rigid automation tools, agentic systems adapt to changing needs and continually improve performance. By augmenting human capabilities, these AI frameworks empower both new and experienced developers to tackle sophisticated projects with greater speed and ease, paving the way for a more inclusive and agile development landscape.

3.2 Reducing Cognitive Load and Context Switching

Agentic AI acts as an intelligent co-pilot in development environments, streamlining workflows by automating routine tasks, providing relevant documentation, and orchestrating coding, testing, and documentation processes. This significantly reduces the need for manual context switching and lowers cognitive load, allowing developers to focus on creative problem-solving rather than repetitive work. While there may be a brief adjustment period as teams adapt, Agentic AI's ability to offload routine cognitive tasks ultimately enhances productivity and reduces mental fatigue. By strategically managing extrinsic cognitive load, Agentic AI frees developers to engage more deeply with complex and innovative aspects of their projects, improving both efficiency and DevX.

3.3 Enhancing Feedback Loops and Continuous Improvement

Agentic AI systems foster a dynamic DevX by delivering continuous, real-time feedback and ongoing learning within the software lifecycle. These intelligent agents not only automate and refine workflows, but also leverage predictive analytics to proactively detect and prevent issues before they arise. By incorporating iterative improvement and self-correction, Agentic AI transforms traditional, reactive feedback into proactive quality assurance processes. This leads to faster problem resolution, reduced failure rates, and a more reliable, less stressful software development environment.



4 Transformative Applications of Agentic AI in the Software Development Lifecycle (SDLC)

Pivoting from Assisted AI/GenAI to Agentic AI means making the SDLC the operating spine. Instead of AI merely suggesting requirements, code, or tests, autonomous yet governed agents plan–act–learn across each stage. They capture and refine requirements from conversations into user stories with traceable design artifacts; scaffold policyaware code, open PRs, and run CI pipelines; generate, prioritize, and flakehunt tests; provision ephemeral environments; orchestrate progressive deployments; then watch production telemetry to trigger safe rollbacks, create incidents, and feed learnings back into the backlog. Guardrails—rolebased access, approvals, secrets hygiene, SBOM/SAST/DAST gates, data privacy, and audit trails—keep humans in the loop at decision points, while agents handle toil and tool orchestration (repos, issue trackers, CI/CD, cloud, and observability). The result is a closedloop, eventdriven SDLC where AI shifts from a helpful sidekick to a reliable teammate, accelerating velocity, improving quality and compliance, and reducing total cost of change.

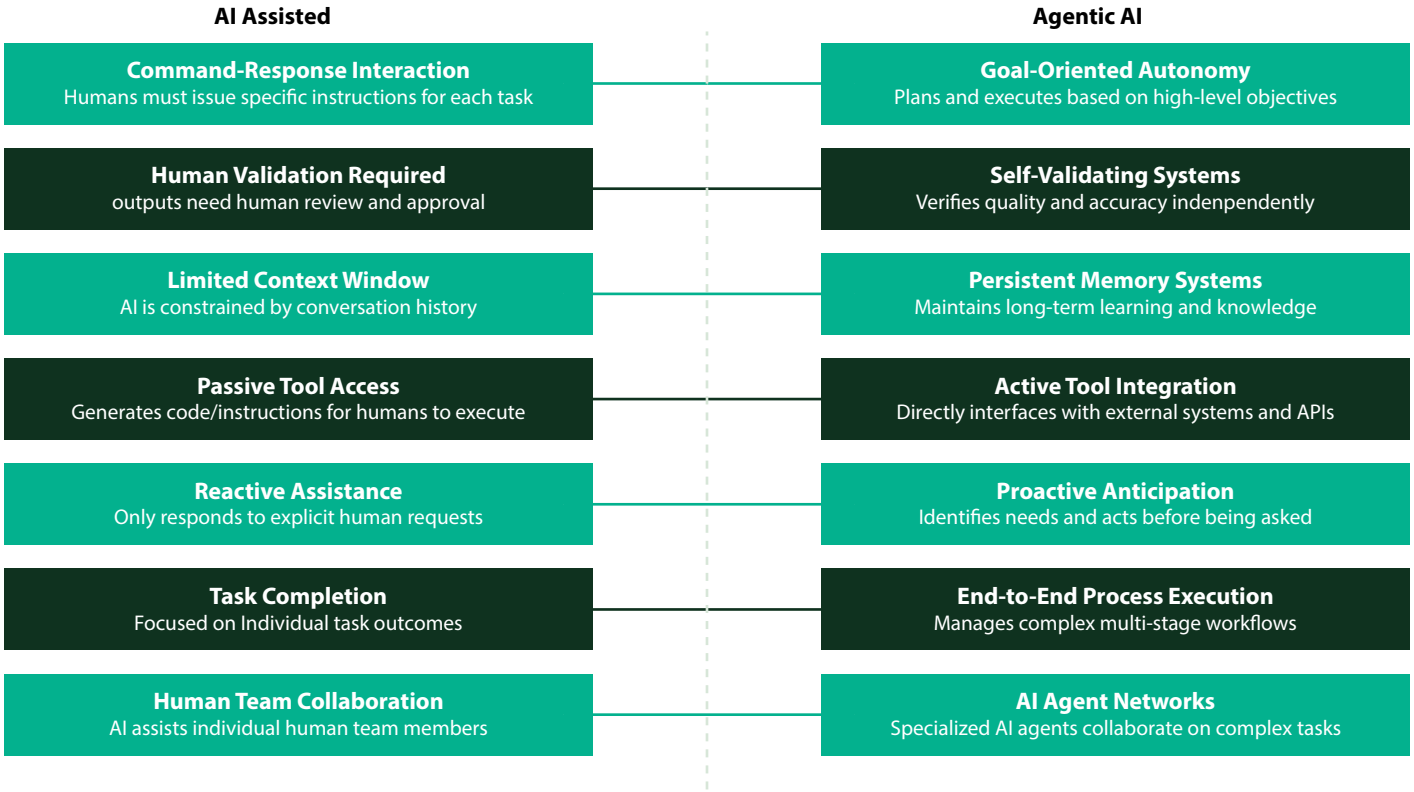


Fig 2. The Paradigm Shift: From AI-assisted to Agentic AI

Agentic AI can be applied throughout the SDLC, automating, optimizing, and enhancing various stages of software development to improve workflow efficiency.

SDLC Phase	Agentic AI Application	Traditional Pain Point Addressed	Impact on DevX Metrics
Requirements and Planning	Proactive requirements elicitation, autonomous user story synthesis, AI-powered risk anticipation, resource allocation optimization	Manual tracking, unclear requirements, poor prioritization, risk identification	40% time savings for business analysts, improved decision-making, better planning, reduced cognitive load
Design and Architecture	Generative architectural blueprints, agent-led simulation of design alternatives, dynamic design validation	Suboptimal design choices, lack of data-driven insights	Enhanced decision support, optimized solutions, accelerated design
Coding and Development	Agentic generation of code—from granular snippets to robust modules—autonomous code refactoring, multi-file orchestration, and self-organizing project structures	Manual coding effort, technical debt, inconsistent code, repetitive tasks	80-90% improvement in database code generation, 60-70% for APIs/ microservices, 60% for UI code, reduced time on routine tasks, improved code quality
Testing and Quality Assurance	Self-directed test case creation, autonomous test maintenance, intelligent test data synthesis, agent-driven shift-left strategies, adaptive and self-healing test suites	Manual test creation, fragile test scripts, insufficient test data, late bug detection	Significant time savings in test case development, reduced maintenance overhead, improved testing efficiency, proactive defect identification
Debugging and Error Resolution	Continuous real-time anomaly detection, predictive fault analytics, and agent-proposed remediations with self-initiated fixes	Time-consuming debugging, human error in detection, reactive bug fixing	Significant time savings, enhanced accuracy, continuous improvement in error identification, proactive bug prevention
Code Review and Governance	Agentic PR summarization, context-sensitive and adaptive feedback, autonomous static and security analysis, self-correcting linter, and compliance enforcement	Slow PR merges, human effort in reviews, inconsistent standards, overlooked vulnerabilities	89% faster PR merges, 34% fewer regressions, 87% AI-provided feedback, improved code quality
Documentation and Knowledge Management	Automated and context-aware documentation generation, agentic retrieval, and curation of organizational knowledge	Inefficient documentation, time spent searching for information, outdated documentation	Streamlined documentation process, reduced search time, accelerated learning, living knowledge bases
Deployment and Operations (CI/CD)	AI-orchestrated CI/CD pipelines, agent-driven bottleneck identification and root-cause analysis, autonomous incident response, intelligent rollout, and rollback management	Bottlenecks, manual processes, slow recovery from failures	Accelerated execution, reduced cycle time, improved resilience, faster incident response, optimized build/deployment

Table 2: Agentic AI in SDLC and Impact on DevX

The table effectively summarizes the concrete Return on Investment (ROI) of Agentic AI on DevX across the entire SDLC in a clear, evidence-based way, driving shared vision of technology leaders.

Infosys uses the Live Enterprise Application Development Platform platform alongside GitHub Copilot (Agent Mode) to provide AI-powered support across the SDLC. The platform assists architects, developers, and DevOps teams with everything from requirements to release management. The platform ensures generated code follows best practices, maintains strong security, and simplifies maintenance. This streamlined approach enhances productivity and code quality.

5 Challenges and Considerations for Agentic AI Adoption

Although Agentic AI holds great potential to improve DevX, its effectiveness in the developed workflows requires careful and strategic implementation of the potential of this technology to overcome its inherent challenges connected with reliability, security, ethical implications, and integration and talent development complexities. The figure outlines the main challenges and considerations for Agentic AI adoption.

Ensuring Reliability & Predictability	• Ensuring reliable agentic AI • Trust, transparency, and gradual adoption
Ethical Implications	• Transparency and Explainability • Bias and Discrimination • Accountability • Control and Misuse
Data privacy and Security concerns	• Sensitive Data • Privacy-preserving testing • Security Guardrails
Sustainability & Green IT	• Energy-efficient agent and model design • Compute, token, and carbon footprint optimization • Sustainable scaling of agentic workflows • Cost-efficient and responsible infrastructure usage • The dashes need to be bullets as used elsewhere.
Integration Complexity & Talent Investment	• Agentic AI integration Challenges • AI engineering skills expanding • Talent upskilling for Agentic era
Balancing Automation with Human Oversight & Control	• Human-guided Agentic AI collaboration • AI over-reliance risks intervention • Human-in-the-loop Agentic AI essential

Fig 3: Agentic AI: Adoption Challenges & Key Considerations

6 The Future of Software Development: Human-AI Collaboration

Looking ahead, the future of software development will be defined by a seamless partnership between humans and AI, reshaping the coder's role from routine implementer to creative architect and strategic problem-solver.

As Agentic AI platforms mature, developers will increasingly migrate toward higher-order tasks—designing unique solutions, architecting complex systems, and steering innovation—while AI agents take on repetitive coding, optimization, and even refactoring. This evolution is accompanied by the rise of highly personalized development environments tailored by AI to individual workflows, learning preferences, and project contexts, making the act of software creation more intuitive and adaptive than ever before.

Emerging protocols and standards, such as Model Context Protocol (MCP) and the ever-evolving APIs, are driving interoperability and flexibility across platforms, enabling richer, more dynamic integrations at scale.

In this new landscape, a reimagined social contract emerges: Collaboration, ethical stewardship, and continuous learning become core developer tenets, while both organizations and individuals navigate questions of accountability, privacy, and the equitable distribution of value produced by human-AI synergies. This shift not only augments productivity and creativity, but also catalyzes a culture where both human ingenuity and machine intelligence are leveraged for sustainable, responsible digital transformation.

7 Recommendations for Implementing Agentic AI to Enhance DevX

The technology leaders intending to deploy Agentic AI to powerfully enhance DevX and transform a company internally should consider the following feasible actions:

7.1 Strategic Roadmap for Phased Adoption

This strategic framework outlines a phased "Ignite-Accelerate-Autonomize" approach to integrating Agentic AI for an enhanced DevX, focusing on productivity, code quality, innovation, and responsible AI governance. Initial phases target AI-powered code completion, documentation, and review, scaling to intelligent refactoring, automated testing, and, eventually, fully autonomous development and DevSecOps capabilities. The figure shows the phases of Agentic AI adoption.

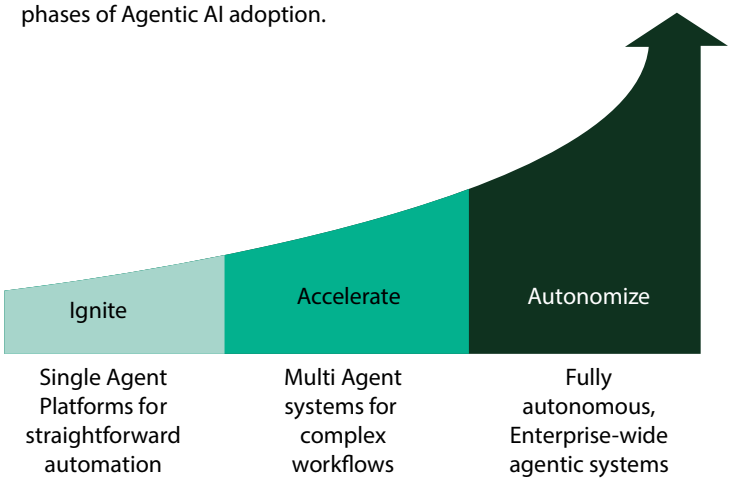


Fig 4: Phased Adoption Plan

The Automation Maturity Model offers a structured framework to evaluate and advance an organization's toolset, mapping progress across four distinct levels: Point Tools, Integrated Solution, Cognitive, and AI-led Autonomous. At the foundational level, Point Tools address specific tasks in isolation, delivering targeted automation without broader connectivity. Advancing to the Integrated Solution stage, organizations begin unifying disparate tools, enabling coordinated workflows and smoother data exchange. The Cognitive level introduces intelligent systems capable of learning from patterns and adjusting processes dynamically, bridging the gap between manual oversight and automated adaptation. At the pinnacle, AI-led Autonomous tools orchestrate development environments with minimal human input, leveraging advanced Agentic AI to drive end-to-end automation. By aligning these maturity levels with a phased adoption strategy, organizations can clearly chart their journey from basic automation toward fully autonomous, intelligent development operations. The figure below shows the levels of the Automation Maturity Model and their characteristics.

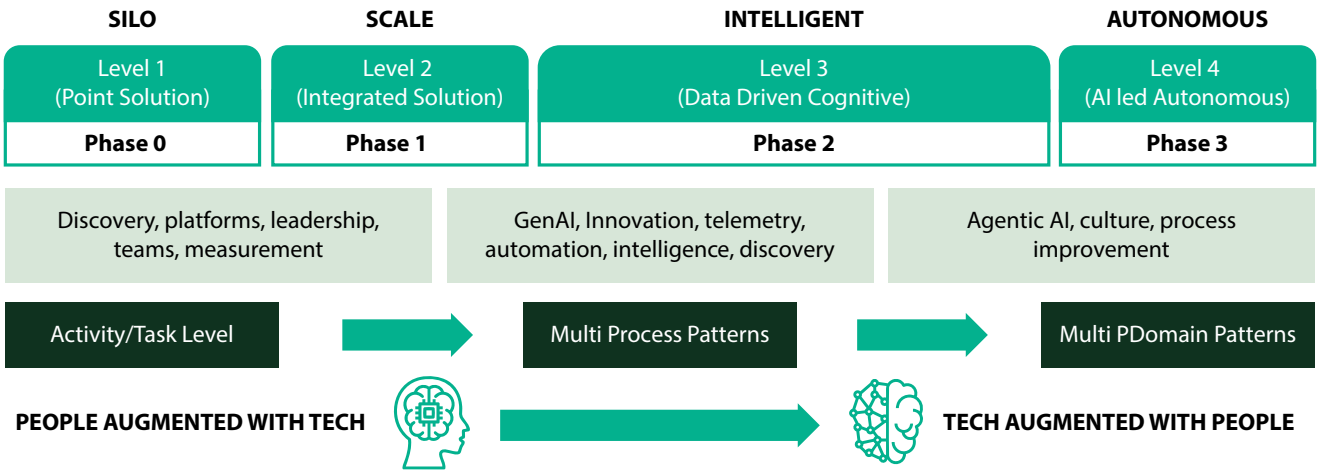


Fig 5: Automation Maturity Model

7.2 Best Practices for Governance and Security

The strong governance and security systems required are not compliance-related issues but critical enablers that can support the scaling of Agentic AI in a safe and effective manner. It is the duty of organizations to put in place broad systems of policies, controls, and monitoring known as comprehensive guardrails. The following figure provides information on security and governance considerations for Agentic AI implementation.

Implement RAI guardrails
Frameworks of policies, controls, and monitoring mechanisms

Safeguard data privacy
Isolate sensitive data, anonymize information, secure credentials

Enforce least privilege
Grant minimum necessary permissions to agents

Adopt cybersecurity risk management
Continuous logging, security monitoring, red teaming

Ensure transparency
Tools for tracking decisions and explaining suggestions

Invest in governance
Establish robust governance frameworks, tooling, platforms

Fig 6: Governance and Security Tenets

7.3 Investing in Skills and Human-AI Partnership Models

The strong governance and security systems required are not compliance-related issues but critical enablers that can support the scaling of Agentic AI in a safe and effective manner. It is the duty of organizations to put in place broad systems of policies, controls, and monitoring known as comprehensive guardrails. The following figure provides information on security and governance considerations for Agentic AI implementation.

Provide training programs
Courses in machine learning, NLP, and agent-based system design, workshops, hackathons

Develop prompt literacy
Ability to craft precise and effective AI inputs, prompt store context engineering skills

Establish collaboration frameworks
Responsibilities, review processes, performance metrics

Upskill to AI engineers
Coding proficiency, AI model, Agentic understanding, and ethics

Success Measurement
Productivity, quality, & satisfaction metrics, Human-AI collaboration

Fig 7: Team Investment

Conclusion

Agentic AI is poised to revolutionize developer experience by automating complex, repetitive tasks and reducing cognitive load across the software development lifecycle. This technology accelerates code generation, refactoring, testing, debugging, and review, resulting in greater productivity, higher-quality code, and improved developer satisfaction.

While the adoption of Agentic AI presents challenges such as reliability, data privacy, security, and ethical considerations, these can be addressed through strategic planning, robust governance, responsible AI usage, and a focus on human-AI collaboration. The future developers will act not only as coders, but also as an orchestrator, leveraging personalized AI environments to innovate and achieve more than ever before.

Integrating Agentic AI into software development marks a fundamental shift in how software is built and maintained. Autonomous, goal-driven AI agents can streamline every stage, from requirements gathering and design, to deployment and monitoring — reducing errors, lowering costs, and boosting efficiency and experience.

Specialized agents—including Supervisor, Requirement, Design, Coding, Reviewer, Testing, Deployment, and Monitoring agents—work collaboratively to optimize workflows and respond to feedback in real time. Whether creating new systems or improving existing ones, Agentic AI demonstrates versatility and effectiveness.

As Agentic AI develops, its potential applications are becoming more evident. Organizations adopting this approach can enhance competitiveness, and elevate developer experiences and digital solutions.

References

- https://www.linkedin.com/posts/nareshchoudhary_agenticsdlc-agenticai-genai-activity-7375751351013879808-OBOk
- https://blogs.infosys.com/emerging-technology-solutions/artificial-intelligence/agentic-ai-the-new-trend-in-generative-ai.html
- https://www.infosys.com/iki/research/agentic-ai-systems.html
- https://www.infosys.com/iki/techcompass/harnessing-agentic-ai.html
- https://www.infosys.com/services/data-ai-topaz/offerings/agentic-foundry.html
- https://www.infosys.com/iki/perspectives/agentic-ai-software-development.html
- https://www.atlassian.com/developer-experience
- https://virtuslab.com/blog/backend/what-is-developer-experience/
- https://swimm.io/learn/developer-experience/developer-productivity-5-metrics-4-barriers-and-how-to-overcome-them
- https://www.geeksforgeeks.org/blogs/challenges-of-software-developers/
- https://www.uplift.ltd/posts/typical-software-development-pain-points/
- https://www.calavista.com/how-to-improve-efficiency-in-software-development/
- https://develocity.io/10-developer-pain-points-that-kill-productivity/
- https://appitventures.com/blog/8-issues-in-software-development-and-how-to-tackle-them
- https://www.youtube.com/watch?v=Js70-FL-5BI
- https://aisera.com/blog/agentic-ai
- https://www.uipath.com/ai/agentic-ai
- https://www.moveworks.com/us/en/resources/blog/what-does-agentic-automation-mean
- https://www.ibm.com/think/insights/agentic-ai
- https://www.ibm.com/think/topics/ai-agent-types
- https://datafortune.com/how-agentic-ai-can-shape-the-future-of-software-development/
- https://arxiv.org/html/2507.02660v1
- https://www.destinationcrm.com/Articles/Web-Exclusives/Viewpoints/The-Agentic-AI-Design-Paradigm-Reshaping-How-We-Build-Software-169958.aspx
- https://arxiv.org/html/2506.04980v1
- https://cloud.google.com/discover/what-are-ai-agents
- https://www.langchain.com
- https://www.crewai.com/
- https://github.com/openai/swarm
- https://solutionsarchitecture.medium.com/the-agentic-ai-revolution-transforming-software-as-a-service-a7c915172b33
- https://develocity.io/10-developer-pain-points-that-kill-productivity/
- https://linearb.io/blog/how-agentic-ai-will-disrupt-your-software-delivery-lifecycle
- https://www.youtube.com/watch?v=Js70-FL-5BI
- https://servicesground.com/blog/agentic-ai-developer-productivity-ethics/
- https://about.gitlab.com/the-source/ai/emerging-agentic-ai-trends-reshaping-software-development/
- https://news.microsoft.com/source/features/ai/meet-4-developers-leading-the-way-with-ai-agents/
- https://www.gsdcouncil.org/blogs/boosting-productivity-agentic-ai-for-project-management
- https://bito.ai/product/ai-code-review-agent/

About the Author

Naresh Choudhary,
SVP - Head – Enterprise Productivity and Engineering Excellence

Co-author

Srinivasa Sujit Rao - Principal Consultant QLTY

Infosys Topaz is an AI-first set of services, solutions and platforms using generative AI technologies. It amplifies the potential of humans, enterprises, and communities to create value. With 12,000+ AI assets, 150+ pre-trained AI models, 10+ AI platforms steered by AI-first specialists and data strategists, and a 'responsible by design' approach, Infosys Topaz helps enterprises accelerate growth, unlock efficiencies at scale and build connected ecosystems. Connect with us at infosystopaz@infosys.com.

For more information, contact askus@infosys.com

Infosys®
Navigate your next