

Before You Build: The Agent Blueprint Most Teams Skip

Why the real challenge is not building the agent – it's defining the problem it should solve

Nicolas Freitag
Merve Usanmaz
Jan Otis Ernst

June 2026
Blogpost
Copyright © COREtransform GmbH

Public

Key Facts

- › **Agentic AI is the fastest-growing enterprise software category on record.** Gartner predicts 33% of enterprise software will include agentic AI by 2028, compared to less than 1% in 2024. No prior enterprise technology category has moved from marginal to mainstream in under four years
- › **The failure rate is high and preventable.** 85% of AI initiatives without a clearly defined business outcome will deliver less value than expected and most will be discontinued. The main cause is not a technical fault, but the lack of a clearly defined problem statement at the outset
- › **The cost of late discovery is punishing.** Whilst a requirements error identified during the design phase costs one work unit to rectify, the same error costs ten work units during development, and if it is only discovered after deployment, it costs a hundred. Teams that complete the full pre-build framework typically reduce total development effort by 30 to 50 percent
- › **Not every problem is suited for an AI agent.** A good use case scores high on four dimensions: frequency and volume, rule clarity, data readiness, and error tolerance. If you are missing two or more reconsider before you build
- › **Use case complexity is not linear.** Every agent activates some combination of six components: perception, memory, planning and reasoning, tools, execution, and self-correction. A use case that activates all six is an order of magnitude more complex than one that activates two. Mapping the component profile before committing determines whether the project is scoped realistically
- › The AI Agent Business Model Canvas forces teams to answer nine structural questions before a single technical decision is made: covering users, value delivered, data and system requirements, cost structure, and measurable success criteria
- › Every agent workflow must answer three questions before development begins: What goes in? What happens between? What comes out? A completed workflow design is the entry ticket to development

¹ (Gartner, 2024)

² (Gartner, 2026)

³ (CORE/EPAM Systems, 2026)

1. The window is open – but not for long

The headlines are no longer hypothetical. JP Morgan deploys AI agents to process 12,000 commercial loan agreements overnight. Klarna replaced 700 customer service roles with a single agent in 30 days. Salesforce reports that autonomous agents are resolving 83% of customer cases without human involvement. What was a technology experiment 18 months ago is now a boardroom agenda item. In many industries, it's already a competitive differentiator.⁴

Gartner predicts that by 2028, 33% of enterprise software will include agentic AI, up from less than 1% in 2024. The pace of adoption means organizations that move with structure and discipline now will establish operational advantages that compound over time. Those that wait for certainty will be catching up to a moving target.⁵

At EPAM Advisory, we see the same pattern in every client engagement: the intent is there, the budget is allocated, and the pressure from leadership is real. Yet the structured groundwork that turns that intent into a working, production-grade agent is missing. In workshop after workshop, when business and technology teams sit down to “build an agent,” a familiar pattern emerges: the conversations moves quickly toward technology choices and architecture, and the energy in the room is real and genuine. Yet in that momentum, one critical question often gets crowded out: What problem, exactly, are we solving? This is not a failure of intent or capability. It reflects a broader challenge: agentic AI is moving faster than the organizational playbooks designed to govern it. Teams are navigating genuinely new territory, often under significant leadership pressure, and without established frameworks to fall back on. The instinct to move quickly is understandable, even rational. But without a clearly defined problem at the center, the path from prototype to production becomes far harder than it needs to be.

The consequences are well-documented: scope creep after the first sprint, integration surprises that were obvious in hindsight, agents that perform flawlessly in a controlled demo and break on the first live input. Rework at this stage is expensive in engineering hours and erodes stakeholder confidence, narrowing the political window for future AI investment. More fundamentally, when the problem to be solved was never precisely defined, success itself becomes undefinable. Without a clear baseline and measurable outcome, organizations cannot determine whether their agent is delivering value, performing adequately, or quietly failing. This ambiguity makes course correction nearly impossible.⁶

1.1 What is an AI agent and why it's different

Before assessing whether a use case is suited for an AI agent, it helps to be precise about what an agent actually is. The term is used loosely, and that looseness can cause confusion.

An AI agent is an autonomous AI system that perceives its environment, makes decisions, and takes actions to achieve specific goals.⁷ Unlike a chatbot which responds to a single prompt and forgets the conversation the moment it ends, an agent operates across multiple steps, uses

⁴ (JPMorgan Chase & Co., 2025) (Klarna Press Reslease, 2024) (Salesforce, 2026)

⁵ (Gartner, 2024)

⁶ (CORE/EPAM Systems, 2026)

⁷ (Beth Stackpole, MIT, 2026)

external tools, maintains persistent memory, and self-corrects when it encounters unexpected results. The shift is fundamental: from systems that respond to systems that act.

This distinction matters for use-case design. A chatbot can answer a question. An agent can receive an email, extract the key request, search three internal systems, draft a proposal, route it for approval, and send it without a human touching the keyboard. The capabilities are qualitatively different, which means the selection criteria must be different too.

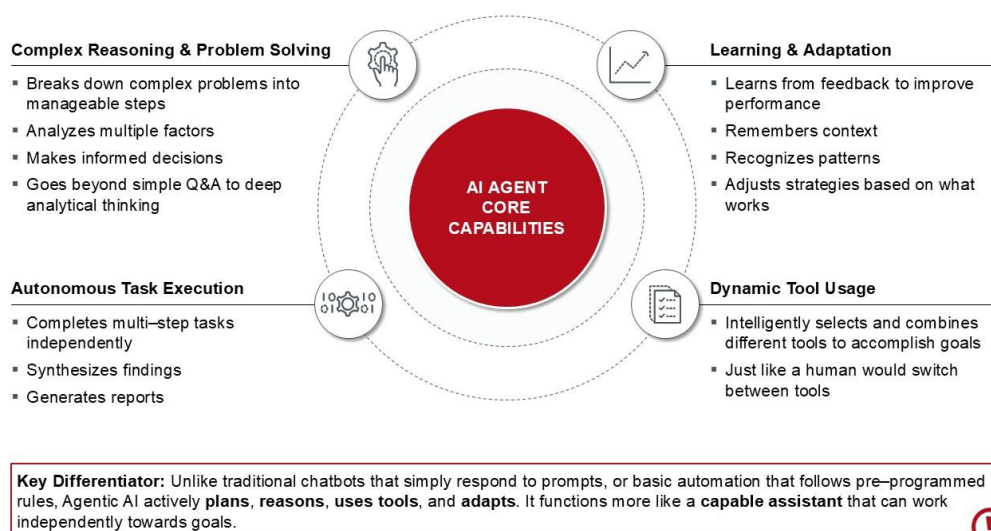


Figure 1: Agentic AI extends beyond prompting toward autonomous execution

Understanding the full capability range of an agent is what sharpens use-case selection. The higher the complexity of what the agent could do, the more precisely the team must define what it should do before a single line of code is written.

1.2 Why the pre-phase pays

Organizations that invest structured effort before development begins consistently outperform those that do not across four measurable dimensions:

- **Alignment.** When business and technology teams work through the problem definition together before development starts, they arrive at build speaking the same language. Misalignments that typically surface in sprint review meetings (e.g. 'that is not what we meant by approval') are resolved on paper together in a workshop
- **Clarity of scope.** A team that has done the structured pre-build work knows exactly what they are building and why. That clarity translates directly into tighter user stories, better-defined development tasks, and fewer 'how should we handle this' interruptions during the sprint
- **Cost reduction** through fewer iterations. Fixing a requirement error in design costs one unit of effort; fixing it during development costs ten; fixing it after go-live costs one hundred. A well-run pre-phase eliminates most requirement errors before they enter the

codebase. Teams that complete the full pre-build framework typically reduce their total development effort by 30 to 50 percent, because they build less of the wrong thing⁸

- **Faster time-to-value.** Counter-intuitively, spending more time in the pre-phase shortens total project duration. A scoped, well-defined agent can be built and deployed in days. An under-specified agent accumulates edge cases, exception-handling patches, and 'one more thing' requests that extend the timeline indefinitely

Taken together, these four dimensions point to the same conclusion: the pre-phase is not overhead, it is the highest-return investment in any agent project. The organizations that treat it as optional are the ones that end up rebuilding. Those that treat it as non-negotiable are the ones that ship, scale, and measure results. This is the pattern that separates organizations that extract real value from AI agents from those that produce expensive, short-lived demos: the quality of the groundwork done before the build begins. The technology is no longer the bottleneck. Clarity is.

2 The missing prerequisite: qualify your use case before you build

The most common and most costly mistake in AI agent projects is choosing the use case wrong or not choosing it at all and instead letting the technology choose for you. When a team starts with “we want to use AI agents” rather than “here is a problem worth solving,” the resulting agent typically solves nothing particularly well. Use the matrix below to stress-test your candidate use case before making any development commitment.

Assessment matrix						
Use Case	Example	Frequency & Volume	Rule Clarity	Data Readiness	Error Tolerance	
Before you build Agentic AI works best when the problem fits the tool. Use this matrix to assess your use case across four dimensions – And find out if an agent is the right answer	1 Strategic/Creative <ul style="list-style-type: none"> ▪ Design a new product concept ▪ Define a go-to-market strategy 	Low / one-time – Every instance requires fresh thinking	No clear rules – Highly judgment-Driven and context-Dependent	Relies on tacit knowledge – Inputs are unstructured or missing	Very low – Wrong decisions have lasting business consequences	⊕
	2 Operational/Transactional <ul style="list-style-type: none"> ▪ Weekly status reports from project data ▪ Meeting notes & action items after client calls 	High volume, daily recurrence – Strong ROI from automation	Clear logic, predictable edge cases – Easy to encode in prompts	Structured inputs available via ERP, databases, or APIs	High – Deviations are quickly detectable and reversible	⊕
	3 Research & Synthesis <ul style="list-style-type: none"> ▪ Summarize sector trends ▪ Compile competitor benchmarks into a structured overview 	Medium/high – Recurring but not fully standardized	Framework exists, but requires prompt tuning and iteration	Mostly available – May need light data preparation	Medium – Human spot-checks at key steps keep quality high	⊕


The rule of thumb: High frequency + clear rules + clean data + recoverable errors. Miss two or more – Rethink before you build. 

Figure 2: The Use Case Assessment Matrix: four dimensions to evaluate before you commit to building

A good use case for an AI agent meets four criteria: it occurs frequently enough to justify the build and operating cost, its logic is clear enough to encode in a prompt and a workflow, the required data is accessible, structured, and sufficiently up to date for the agent to act on reliably, and when

⁸ (CORE/EPAM Systems, 2026)

the agent makes a mistake the error is detectable, recoverable, and a human can intervene before consequences compound. Miss two or more and you should reconsider before you build.

2.1 Three strategic categories of use cases suited for an AI agent build

Not every business problem is suited for an AI agent, and not every agent is equally complex to build. A useful first filter is to classify candidate use cases across three strategic categories:

1. Automation use cases replace repetitive, rule-based tasks that currently consume human capacity without requiring human judgment: high-volume processing, structured data handling, and routine coordination. These offer the clearest return on investment and the lowest delivery risk. They are the natural starting point for most organizations.
2. Augmentation use cases enhance human decision-making by giving people better, faster access to synthesized information: research compilation, recommendation generation, knowledge retrieval across fragmented systems. Here, the agent does not replace the decision, but it improves its quality and speed. Human-in-the-loop checkpoints are non-negotiable.
3. Orchestration use cases coordinate complex, multi-step processes that span systems, teams, or departments where the value lies not in automating a single task, but in eliminating the friction between many. These are the highest-value category and the most demanding to build well.

2.2 The six components of an AI agent: building the component profile

Every agent is assembled from some combination of six building blocks: perception (how it receives input), memory (what it retains across steps), planning and reasoning (how it decides what to do next), tools (which external systems it can access and act upon), execution (what actions it takes), and self-correction (how it handles unexpected results).

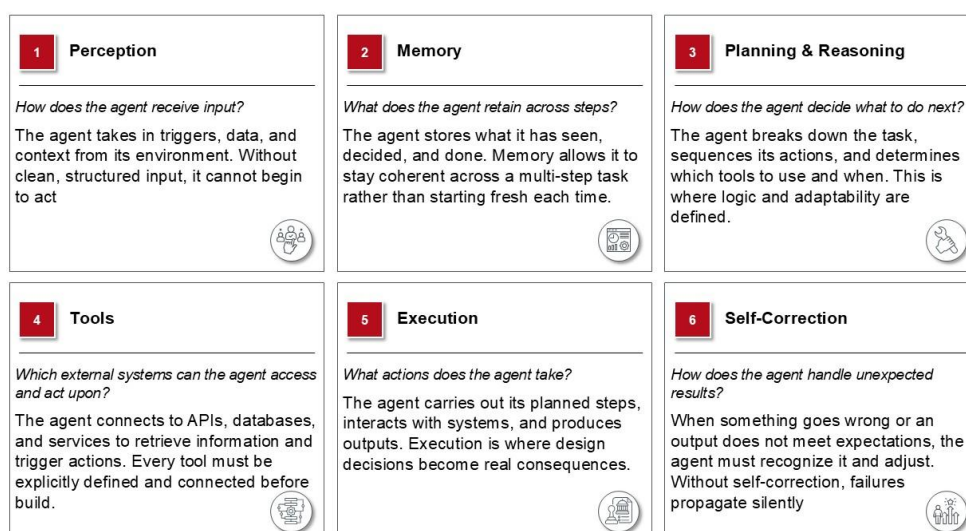


Figure 3: Six building blocks determine whether an agent can do its job. Miss one and the gap shows up in production

A use case that activates all six components is not incrementally harder than one that activates two. It is an order of magnitude more complex to specify, build, and test. Understanding this before you commit determines whether you scope the project realistically or underestimate it entirely.

Mapping a candidate use case against both dimensions strategic category and component profile gives a team a concrete basis for prioritization and scoping before a single line of code is written.

2.3 From use-case to blueprint: the 'AI Agent Business Model Canvas'

Once a use case has cleared the assessment matrix, been classified by strategic category, and mapped against its component profile, the team knows the problem is worth building for and has a realistic sense of complexity. But qualification alone is not enough to build from. This is where most teams still fall short: they know "what" but have not answered "who exactly," "how exactly," and "what does success look like?"

The 'AI Agent Business Model Canvas' adapted from the classic Business Model Canvas by Osterwalder provides a structured template for thinking through all the dimensions of an agent use case before a single technical decision is made. It forces teams to answer nine questions:

- › What unique value it delivers compared to the manual process (key propositions),
- › Who the agent serves (customer segments),
- › Which external tools, APIs, or data sources it must connect to (key partners),
- › What tasks it will perform step by step (key activities),
- › What internal data, documents, and system access it needs (key resources),
- › How it interacts with users; autonomously or with human review (customer relationships),
- › How users will trigger and interact with it (channels),
- › What it will cost to build and run (cost structure),
- › What measurable impact it will create (value created).

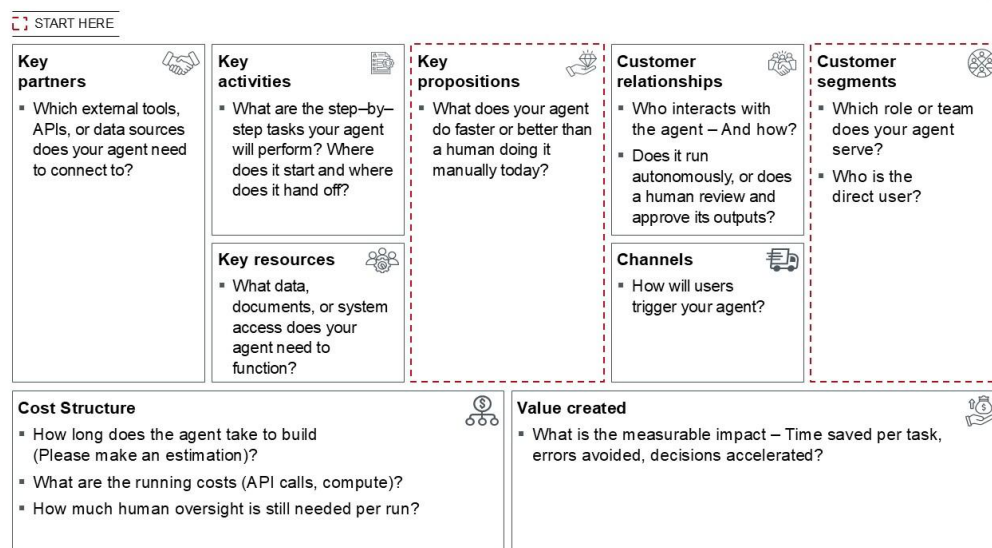


Figure 4: The AI Agent Business Model Canvas (CORE proprietary template)

The canvas is not a bureaucratic exercise. It is a forcing function. Teams that complete it discover gaps they would have hit mid-build: missing data access, unclear success criteria, human-in-the-loop boundaries that no one had defined. Discovering those gaps on a canvas takes minutes. Discovering them after six weeks of development takes much longer.

3 From canvas to code: workflow design before agent building

A completed canvas answers what the agent should do. A workflow diagram answers how, in what order, and with what safeguards. These are different questions and require different thinking. Skipping workflow design before building is equivalent to starting construction without architectural drawings: structural problems surface at the most expensive possible moment. One practical reality: the people best positioned to define the workflow are often those who currently perform the task manually and who may be reluctant to document it in detail for obvious reasons. Addressing that dynamic openly and making clear that human oversight remains a design requirement rather than a compromise, is a prerequisite for getting accurate specifications. Every AI Agent workflow must answer three questions:

3.1 What Goes In?

Define the trigger and the exact data the agent receives. A vague input is a guaranteed failure at step one. Specify how the trigger is initiated, through what channel or system, and what the data structure looks like field by field. If the input is ambiguous, the agent will fail before it has taken a single action.

3.2 What Happens Between?

Map every action the agent takes, every tool or system it connects to, every decision point, and every location where a human must intervene before the workflow continues. There are four core patterns to combine: sequential steps, conditional branching, parallel execution, and human-in-

the-loop checkpoints. Workflows that write to, send, or publish anything external must include a human approval gate not as a nice-to-have, but as a non-negotiable design constraint.

3.3 What Comes Out?

Most blueprints stay vague here and that's where agents disappoint. A useful output definition specifies three things: the format of the output, where it lands, and who acts on it. Each dimension has downstream consequences: wrong format breaks integrations, wrong destination makes outputs invisible, wrong recipient delays action.

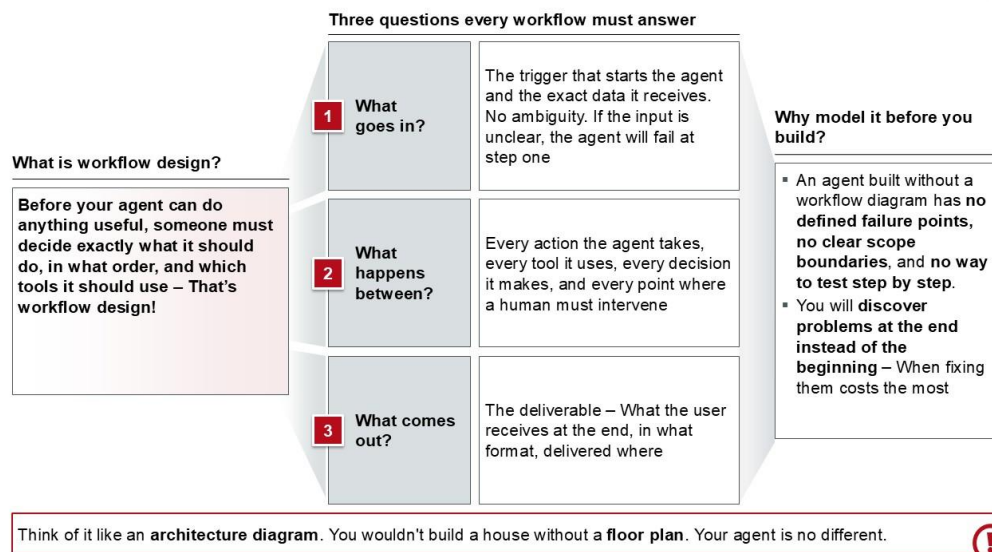


Figure 5: Three questions every agent workflow must answer before build begins

With all three questions answered, the workflow design is complete. The agent has a defined input, a mapped execution path, and a specified output. Now the engineering team has everything it needs to move from design to build.

4. Conclusion: From blueprint to build

AI agents can genuinely transform knowledge work, but only when the right problem has been chosen, properly structured, and thoroughly designed before development begins. The organizations that will capture lasting value from agentic AI are not those with the largest budgets or the most advanced models. They are the ones that invest disciplined thought into the pre-phase: qualifying the use case, completing the canvas, and modelling the workflow end to end.

The three-step framework is deliberately lightweight. Qualifying a use case takes half a day. Checking it against four dimensions, classifying it as automation, augmentation, or orchestration, and mapping it against the six agent components to confirm the problem is worth building for and the complexity is understood. Completing the AI Agent Business Model Canvas takes one structured workshop yielding defined user segments, value proposition, data requirements, cost structure, and measurable success criteria. Designing the workflow takes a second session producing a step-by-step execution map with specified inputs, decision logic, tool integrations,

human approval gates, and output format. At the end of this pre-phase, typically two to three focused sessions, the team holds a development-ready agent blueprint: a precise, agreed-upon specification that engineering can act on immediately.

That blueprint is the entry ticket to development. With it, engineering teams can write accurate user stories, estimate realistic timelines, and build with confidence. Because every critical design decision has already been made, discussed, and agreed. Without it, even the best development team is building in the dark.

Once the blueprint is in place, the technology decision becomes straightforward. The market offers a range of capable agent-building platforms from workflow automation tools like n8n, Make, or Zapier to purpose-built agentic frameworks. At EPAM, we complement these with [CodeMie](#) and [Elitea](#), our proprietary platforms already in use across a growing number of client engagements. The right choice depends on integration requirements, scalability needs, and the complexity of the designed workflow all of which the pre-phase will have already surfaced.

In conclusion, the bottleneck in AI agent projects is almost never the technology but the absence of a clearly defined problem, a structured use case, and a modelled workflow. Complete the pre-phase and the build becomes the easy part. Accordingly, the time to start is not when the technology is perfect. It is now – with a whiteboard, the right stakeholders in the room, and the discipline to answer the hard questions before writing a single line of code.

Sources

CORE/EPAM Systems. (2026). *CORE/EPAM internal project data*.

Gartner. (2024). *Hype Cycle for Artificial Intelligence*. Retrieved from <https://www.gartner.com/en/articles/hype-cycle-for-artificial-intelligence>

Gartner. (2026). *Six Steps to Manage AI Agent Sprawl*. Retrieved from Gartner Predicts 2025: AI Agents Will Require New Governance Frameworks

JPMorgan Chase & Co. (2025). *JPMorgan Chase Annual Report 2025*. Retrieved from Annual Report JPMorgan Chase: <https://www.jpmorganchase.com/ir/annual-report>

Klarna Press Release. (2024). *Klarna AI Assistant Handles Two-Thirds of Customer Service Chats*. Retrieved from Klarna AI assistant handles two-thirds of customer service chats in its first month: <https://www.klarna.com/international/press/klarna-ai-assistant-handles-two-thirds-of-customer-service-chats-in-its-first-month/>

Osterwalder, A. &. (2010). *Business Model Generation*. John Wiley & Sons.

Salesforce. (2026). *Salesforce Agentforce Customer Stories Salesforce*. Retrieved from <https://www.salesforce.com/agentforce/customer-stories/>



Nicolas Freitag is Director at EPAM Advisory and Co-Lead of the Financial Services Industry practice. He has extensive experience across the European banking landscape and has supported a wide range of organizations in large-scale transformation programs. His work focuses on shaping and executing strategy, operating model, and technology initiatives that help organizations restore efficiency, relevance, and sustainable profitability.

Mail: Nicolas.freitag@epam.com



Merve Usanmaz is a Consultant at EPAM Advisory and part of the M&A and Media practices. With an academic background in entrepreneurship and innovation, she works at the intersection of strategic transformation and applied AI supporting clients in designing and implementing initiatives where intelligent process automation and sustainable business model development go hand in hand. Her expertise spans process optimization, AI-driven innovation, and the translation of emerging technologies into tangible business outcomes.

Mail: Merve.usanmaz@epam.com



Jan Otis Ernst is Senior Consultant at EPAM Advisory and part of the Financial Services Industry practice. He supports banks and financial institutions in strategy- and technology-driven transformation initiatives, with a focus on IT and digital strategy, product development and innovation, and the institutionalization of outcome-driven governance models. His work bridges analytical rigor and practical execution, translating strategic ambition into actionable transformation roadmaps.

Mail: Janotis.ernst@epam.com



EPAM is a leading global software development company with over 30 years' industry experience and more than 62,000 employees. EPAM is able to scale its development resources flexibly and in line with requirements, even for complex, large-scale projects.



CORE is EPAM's strategic advisory unit, which has been providing strategic guidance to companies worldwide for over 15 years. Its advisory services are based on sound market knowledge, in-depth technological expertise and a strong methodological approach.

About COREresearch

As an technology think tank, we investigate the patterns of technology-driven transformations in industries where IT plays a significant role in the value creation process. As part of our research activities, we analyse markets and technologies, examine the structures, causes and mechanisms of technological change, and curate findings for clients and the general public. Furthermore, we make selected results of our interdisciplinary research available to a wider audience through comprehensive publications, individual studies and presentations.

Disclaimer

The content and structure of our publications are protected by copyright. The reproduction of content, in particular the use of texts, parts of texts or images, requires prior consent. The logos shown are the property of the respective companies. COREtransform GmbH holds no rights to the logos and uses them exclusively for academic purposes.

COREtransform GmbH
Kurfürstendamm 194
10707 Berlin | Germany
<https://core.se/>
Phone: +49 30 263 440 20
sharedcoreoffice@epam.com

COREtransform GmbH
Limmatquai 1
8001 Zürich | Helvetia
<https://core.se/>
Phone: +41 44 261 0143
sharedcoreoffice@epam.com

COREtransform Ltd.
9 Devonshire Square, 5th Floor
London EC2 4YF
Great Britain
<https://core.se/>
Phone: +44 20 328 563 61
sharedcoreoffice@epam.com