

EXECUTIVE EDITION · FOR MANAGEMENT AND DECISION-MAKERS

Agentic Software Delivery in Financial Institutions

What AI agents mean for how banks, insurers and asset managers build software — and what it takes to make it real

IN THIS EDITION

- What is changing, and why now
- How IT projects change when AI agents do much of the work
- A worked example: migrating an investment reporting platform
- What has to be built, and how it connects to existing systems
- What regulators expect
- What to set firmwide, per department and per project
- How to start now, before the full platform exists
- A roadmap, a 90-day plan and the decisions management needs to take

The full FINATUM white paper of the same title provides the technical detail and all sources.

At a glance

THE SHORT VERSION

Software in banks, insurers and asset managers is still built entirely by people. That is starting to change. AI agents¹ can now take on much of the analysis, documentation, coding and testing in an IT project. Firms that organise for this will deliver faster and with better documentation. The work that remains for people moves towards deciding, reviewing and taking responsibility. Making this safe in a regulated firm takes more than buying a tool: it needs a new delivery model, a controlled technical platform and clear rules for who decides what.

Seven key messages

- 01 **The shift has started at the top of the industry.**
Goldman Sachs runs hundreds of AI coding agents alongside about 12,000 human developers and plans to scale further (Section 1).
- 02 **Projects keep their roles and documents, but their content changes.**
Agents draft and build; people set direction, review and sign off. Approval points become more important, not less (Section 2).
- 03 **Legacy migrations are the best place to start.**
The old system shows exactly what the new one must produce, so agent work can be checked automatically (Section 3).
- 04 **Success depends on a platform, not on individual tools.**
Agents need a controlled workspace, governed access to company systems and a complete audit trail (Section 4).
- 05 **Regulators treat AI-written code like any other IT change.**
Existing rules on change management, testing and records apply in full; the platform must produce the evidence (Section 5).
- 06 **Set the rules centrally, leave the business knowledge local.**
Platform, controls and approved tools are firmwide; departments add their domain knowledge; projects work within both (Section 6).
- 07 **Start now; do not wait for the full platform.**
With an approved AI assistant and a few rules, teams can already extract business rules or design screens. Nothing is wasted (Section 7).

What management needs to decide

- **Ambition and scope:** which kinds of projects should use agents first, and how far they may act on their own.
- **Platform and providers:** which cloud and AI model providers to use, taking existing outsourcing and DORA² assessments into account.

¹ An AI agent is software built on a large language model that does not just answer questions but carries out a multi-step task on its own: it reads documents, writes and tests code, uses tools and reports back. A human sets the task and checks the result.

² DORA, the EU Digital Operational Resilience Act, has applied since January 2025. It sets binding rules for how financial firms manage IT risk, IT changes and IT service providers.

- **Investment:** a small platform team for about three months, plus one or two pilot projects.
- **Governance:** who approves what, and how risk, compliance and data protection are involved from the start.
- **Operating model:** what is fixed for the whole firm, what departments may adapt, and what each project decides.
- **Early start:** which AI-assisted tasks teams may begin now, under which minimum rules.
- **People:** how roles, training and career paths change, including for junior staff.

How to read this paper: small orange numbers point to footnotes that explain technical terms. Numbers in square brackets, such as [1], refer to the sources at the end. A glossary of abbreviations is at the end.

01 What is changing, and why now

In July 2025, Goldman Sachs became the first major bank to put an autonomous coding agent to work at scale. Its Chief Information Officer (CIO), Marco Argenti, said the agent, called Devin, would be “like our new employee who’s going to start doing stuff on the behalf of our developers” [1]. Goldman started with hundreds of these agents next to its roughly 12,000 human developers and said it could scale into the thousands. It expected a productivity gain of three to four times over the AI tools it had used before [1], and Argenti described the underlying models as “basically just as good as any developer” [2].

“It’s really about people and AIs working side by side.”

Marco Argenti, CIO, Goldman Sachs [1]

From assistant to agent

Most banks and insurers already give developers an AI assistant that suggests code while they type. These assistants run on large language models (LLMs)³, the same technology behind well-known chat tools. An agent goes a step further. It is given a task, such as “move this report to the new platform and prove the numbers still match”, and then plans and carries out the steps itself: it reads the old code, writes the new code, runs tests and reports back. A person still sets the task and checks the result, but the person no longer does every step by hand.

Why now

- **The technology has matured.** Agents can now work on large, real codebases and complete multi-step tasks, not just suggest small snippets.
- **The cost of producing documents and code has collapsed.** What used to be the slowest part of a project, writing specifications, code and tests, is now the fastest. Deciding and reviewing become the bottleneck.
- **Peers are moving.** By 2026 Goldman was also building AI agents with Anthropic for trade accounting and client onboarding, beyond software development [3].
- **Many firms face large migrations anyway.** For example, mainstream maintenance for the widely used SAP BW 7.5 data warehouse ends in 2027 [13]. Agents can make these programmes faster and

³ A large language model is an AI model trained on very large amounts of text and code. It can read, write and reason about language and code. Well-known examples are the models behind Claude, ChatGPT and Gemini.

cheaper.

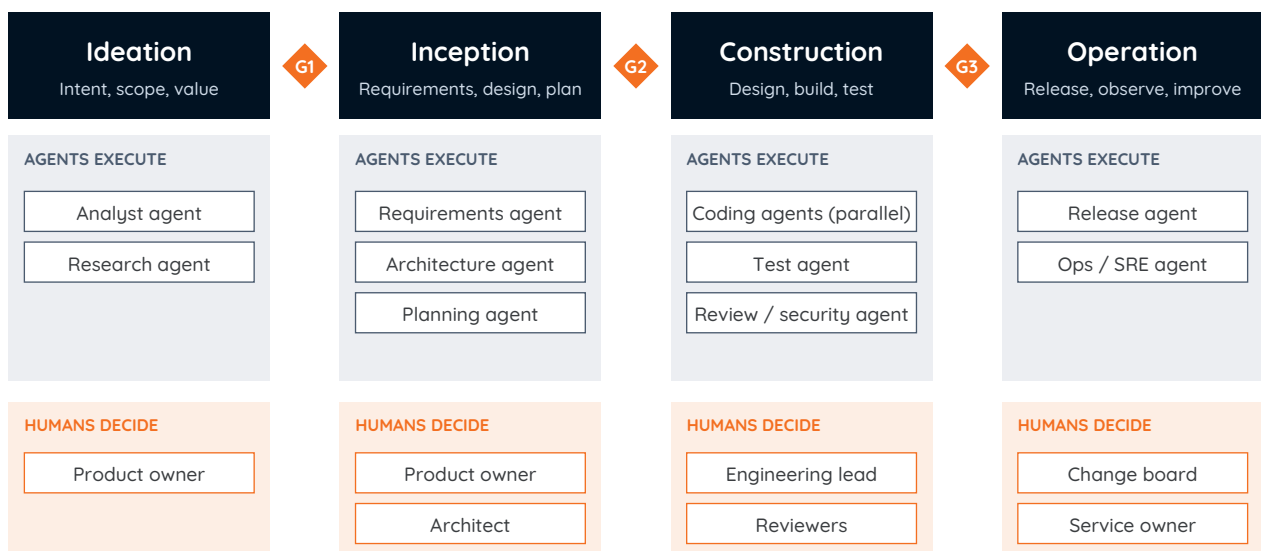
Two cautions

First, agents are not yet reliable when the task is vague. A review of customer deployments by Devin's maker found that it understands code like a senior engineer but executes more like a junior one when objectives are loosely defined [4]. The quality of the instructions decides the quality of the result. Second, regulators expect AI-written code to be controlled like any other IT change. In January 2026, BaFin, the German Federal Financial Supervisory Authority, published guidance that places AI firmly within the existing DORA rules and calls for careful review of AI-generated code [5][6]. **The conclusion of this paper: agents bring speed only if instructions are precise, approval points are explicit and every step is recorded.**

02 How IT projects change

Today, a typical IT project in a financial institution moves through fixed stages: initiation, analysis, design, build, test and release. Each stage has its specialists and its documents, such as the project charter, the business requirements, the IT functional specification, the data model, the backlog of work items⁴, the test cases and the change request. Each specialist turns one document into the next, and context is lost at every hand-over.

In the agentic model the stages and most documents remain, but who produces them changes. **Agents execute; people decide.** The best-documented public method for this is the AI-Driven Development Lifecycle (AI-DLC) published by Amazon Web Services (AWS) [8]. It defines phases, the documents each phase produces and a human approval point, or gate, at each phase boundary. Figure 1 shows a simplified version.



G1 scope sign-off · G2 design and readiness sign-off · G3 release approval · all artefacts and approvals versioned and logged

Figure 1. The agentic delivery lifecycle (simplified from AWS AI-DLC [8]). In every phase, agents do the work and people decide. Gates G1–G3 map to the firm's existing approval bodies.

⁴ A backlog is the prioritised list of work items for a project. In agile methods the items are often called product backlog items (PBIs) or user stories.

What happens to the roles

Roles do not disappear. Their focus moves from producing to directing and checking. Management consultancies report that firms seeing the largest gains have redesigned roles, not just added tools [11]. One open-source method, BMAD, even maps every agile role to an agent [9]; in practice, though, the human counterparts stay essential.

Role today	Role tomorrow	What the person still owns
Project manager	Delivery lead	Priorities, budget, escalation, commitments to stakeholders
Business analyst / requirements engineer	Intent owner: defines precisely what “correct” means	The business rules and acceptance criteria; resolving conflicts with the business
Architect	Architect and owner of the project’s binding standards	Target design, technical standards, design sign-off
Developer	Engineer who supervises agents	Review of all code, the hard cases, accountability for what goes live
Tester / test manager	Quality engineer	Test strategy and the decision that quality is good enough
(new)	Agent platform engineer; AI quality lead	Running the platform; measuring the quality of agent work

Table 1. How roles change. The second column describes responsibilities, not headcount.

Documents become specifications

Agents work from written instructions, so the documents that describe what to build become the most important product of a project. Leading practitioners call this spec-driven development⁵ [10]. Specifications are kept in the same place as the code, are versioned like code, and link every business rule to the code and tests that implement it. This makes projects easier to audit than today, not harder.

How much may an agent do alone?

Not every change deserves the same freedom. Firms should define autonomy levels and let the platform enforce them. At the lowest level, agents only suggest and humans write. In the middle, agents complete a task and a person reviews every change before it is accepted. At the highest level, agents act within pre-approved rules, for example applying routine security updates. Critical and client-facing systems typically start with review of every change.

03 An example: migrating an investment reporting platform

This is an illustrative scenario; durations and team sizes are assumptions, not benchmarks.

An asset manager produces performance, risk and holdings reports for institutional clients from a platform built over fifteen years. Most of the business logic sits in old database code and in spreadsheets; a few people understand it; and operations fixes several recurring errors by hand every month-end. The firm wants to move to a modern data platform without changing a single reported number unless it decides to. Such a migration is a strong first use case for agents: the old system shows exactly what the new one must produce, most of the effort is reading and translating old logic, and the

⁵ Spec-driven development: the team first writes a precise, structured specification of what the software must do, approves it, and only then lets agents build from it. The specification stays the reference for testing and later changes.

work can be split across many agents in parallel. Legacy modernisation is also the first use case Goldman named [1].

How it runs

- 01 Discover.**
Agents inventory every database object, job, report and spreadsheet and map their dependencies. Unused code is flagged.
- 02 Write down the rules.**
Agents extract each business rule into a numbered catalogue. Business experts confirm every rule. This is the first gate.
- 03 Capture the reference.**
The old system is run on, say, 24 past month-ends and all outputs are stored as a “golden master”⁶.
- 04 Build in waves.**
Agents translate the logic module by module and write tests for each rule. Engineers review every change.
- 05 Compare and decide.**
Old and new systems run side by side. Every difference is classified and either fixed or formally accepted (Figure 2).
- 06 Switch over.**
The change board approves the cut-over based on an evidence pack the platform assembles automatically.

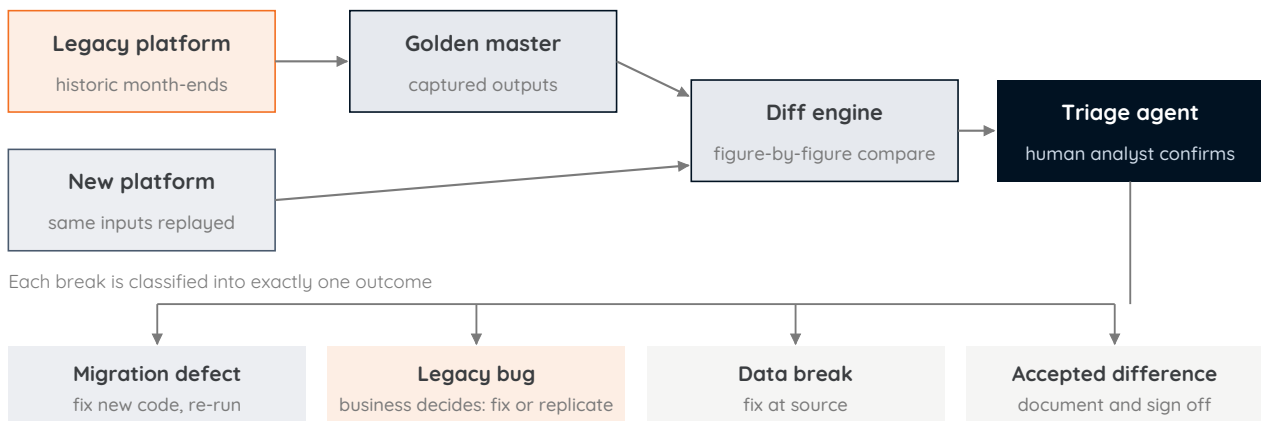


Figure 2. Every difference between old and new results is classified and then fixed or signed off. Nothing is accepted silently.

What the firm gains besides a new platform

- **Hidden errors found.** The comparison exposes errors in the old system that nobody had noticed. Whether to fix them, since clients have seen the old numbers, is a business decision, recorded in a decision log.
- **Manual fixes removed.** Recurring month-end errors are traced to their cause and fixed at the source.
- **Less and cleaner code.** Unused and duplicated logic is not migrated, and the data model is tidied up.

⁶ Golden master: a saved set of outputs from the old system that serves as the reference. The new system must reproduce them exactly, or every difference must be explained.

- **Knowledge captured.** The rule catalogue replaces knowledge that sat in a few people’s heads.
- **Audit readiness.** Every change can be traced from business rule to code, test and approval.

In this example, a 26-week plan replaces what would conventionally be a larger and longer effort. The team is smaller but more senior: a delivery lead, a product owner from the business, one or two intent owners, an architect, three or four engineers who supervise the agents and a quality engineer. **The scarce resource is the time of the business experts who confirm the rules; it should be secured before the project starts.** The same approach works when the legacy platform is SAP BW and the target is, for example, Databricks⁷ with Microsoft Power BI for reporting.

04

What has to be built: the agent platform

Individual AI tools on developers’ laptops are not enough for a regulated firm. Agents need a shared, controlled environment that gives them exactly the access they need, records what they do and keeps people in charge. We call this the agent platform. It has six building blocks.

Building block	What it does, in plain terms	Why it matters
1 Model access and gateway	Connects agents to approved AI models, ideally in EU data centres, through a central gateway ⁸	Data stays where it is allowed to be; costs and usage are visible per team
2 Agent workspace	A sealed-off working environment (sandbox ⁹) where agents read, write and test code	Agents cannot reach production systems or the internet by accident
3 Connections to company systems	Controlled links to ticketing, code repositories, databases, data platforms, test tools and IT service management, using the Model Context Protocol (MCP) ¹⁰	Agents work with real context; access is read-only by default and every action is logged
4 Identity and permissions	Each agent has its own identity, linked to the person supervising it, with the minimum rights it needs	Every action is attributable to an agent and a human
5 Audit trail and monitoring	Records every agent action, approval and result, and feeds the firm’s security monitoring	Provides the evidence regulators and auditors ask for
6 Methods, templates and quality gates	Standard project templates, the firm’s binding technical standards, review rules and automated checks before anything goes live	Makes results repeatable and keeps the human approval points mandatory

Table 2. The six building blocks of an agent platform.

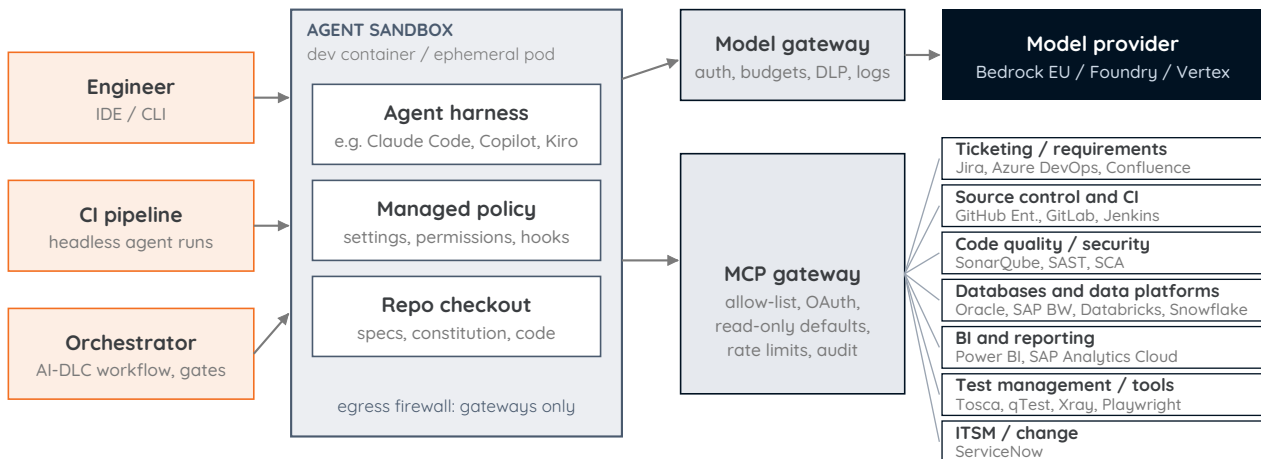
Figure 3 shows how the blocks fit together. The key design choice is that the agent workspace has only two ways out: one to the AI models and one to company systems, both through gateways that check and record everything.

⁷ Databricks is a widely used cloud data platform (a “lakehouse”) that combines a data warehouse with large-scale data processing and machine learning.

⁸ Gateway: a central checkpoint that all requests pass through. It checks who is asking, applies rules and budgets, and records the traffic.

⁹ Sandbox: an isolated computing environment. What happens inside cannot affect other systems, and it can only reach destinations that are explicitly allowed.

¹⁰ MCP is an open standard, introduced in late 2024, for connecting AI agents to other software. A system that offers an “MCP server” can be used by any compatible agent, much as a USB device works with any computer.

**IDENTITY AND SECRETS**

Corporate IdP (Entra ID / Okta) · agent identities (Entra Agent ID, IAM roles) · short-lived tokens · secrets vault

TELEMETRY AND AUDIT

OpenTelemetry from harness, gateways and MCP servers → collector → SIEM (Splunk / Sentinel) + immutable audit store → DORA evidence

Figure 3. How the agent platform connects to the firm. Detailed product options for each element are in Chapter 6 of the full white paper.

Which systems can agents use today?

Most systems found in banks, insurers and asset managers can already be connected. Maturity varies, which matters for planning:

Type of system	Examples	Connection available today
Requirements and ticketing	Jira, Confluence, Azure DevOps	Vendor connectors; for self-hosted Jira (Data Center), community-built ones
Code and build pipelines	GitHub, GitLab	Vendor connectors, generally available
Relational databases	Oracle, SQL Server, PostgreSQL	Vendor and open-source connectors
SAP data warehouse	SAP BW 7.5, SAP BW/4HANA	Open-source connectors from the SAP partner community; no SAP product yet
SAP's successors	SAP Datasphere, SAP Business Data Cloud, SAP HANA Cloud	SAP is adding MCP support (HANA Cloud since early 2026); still early
Cloud data platforms	Databricks, Snowflake	Vendor connectors, generally available, governed by the platform's own permissions
Reporting	Microsoft Power BI	Vendor connectors in public preview: querying and building report models
Testing	Tricentis Tosca and qTest; Xray	Vendor connectors for Tricentis; Xray through its programming interface
IT service and change management	ServiceNow	Vendor connector (requires ServiceNow's AI add-on)

Table 3. Connection maturity as of October 2026. Sources and details are in the full white paper.

The rule for all connections: agents get read access first, change systems only through reviewed proposals¹¹, and never see real client data. They work on masked copies or synthetic test data.

05 What regulators expect

There is no separate rulebook for AI-written software. BaFin's guidance of January 2026 confirms that AI systems fall under DORA and the firm's existing IT risk framework [5]. For software development this means:

- **Review of AI-generated code** and of external code libraries, as for any other code [6].
- **Complete version history** of code, configuration and the AI models used, and **tested rollback** when a model or tool changes [6].
- **Change management as usual:** AI-built changes pass the same approval bodies; the platform provides the evidence.
- **Third-party oversight:** AI model providers and cloud platforms are IT service providers under DORA and need the usual assessment, contracts and exit plans.

Two industry references help turn this into concrete controls. The FINOS AI Governance Framework¹² provides a catalogue of agent risks with matching controls [7]. And a recent BCG reference architecture for banking sets the principle that an agent never approves a regulated action on its own [12]. In software delivery this is the four-eyes principle¹³ applied to machines: the agent that writes a change may not approve it.

06 Who sets what: firm, department and project

Two things go wrong when firms adopt AI in software delivery. If every project sets itself up, controls and quality differ from team to team and nothing is reused. If everything is decided centrally, the central team becomes a bottleneck and the business knowledge that projects depend on is lost. A simple rule helps: **decide centrally what concerns risk, control and reuse; leave locally what concerns business knowledge**. This gives three levels, shown in Figure 4.

¹¹ In software teams, changes are proposed as a "pull request": the change is visible side by side with the old version and can only be merged after review and automated checks.

¹² FINOS, the Fintech Open Source Foundation, is an industry body of banks, insurers and technology firms under the Linux Foundation. Its AI Governance Framework lists AI risks and matching controls, including a section on AI agents.

¹³ Four-eyes (or maker-checker) principle: whoever creates a change may not approve it. A second, independent person must check and approve.

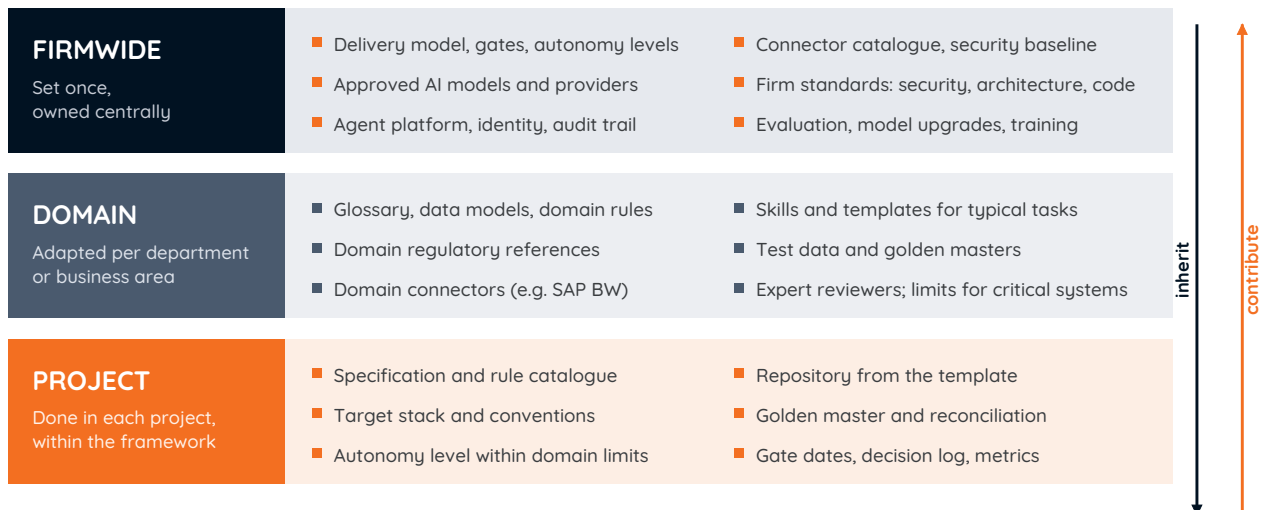


Figure 4. Three levels. Departments and projects inherit the firm's rules and may tighten them, but not relax them. Useful work is passed upwards for others to reuse.

Level	What is decided here	Who owns it
Firmwide	How projects run and who approves what; how much agents may do on their own; which AI models and providers are allowed; the agent platform, identities and audit trail; which system connections are approved and how they are secured; common technical standards; training; how success is measured	Chief Information Officer (or Chief Technology Officer) with the platform team; sign-off by risk management
Department or domain	The vocabulary, data and business rules of the area; its regulatory requirements (for example capital reporting for banks, Solvency II for insurers, performance reporting for asset managers); connections to the area's own systems, such as SAP BW in finance; ready-made templates for recurring tasks; test data; stricter limits for critical systems; the experts who review	Lead architect of the domain together with the business owners of the data
Project	What exactly is to be built and the rules it must follow; the target technology; how much freedom agents get within the domain's limits; the dates of the approval gates; the record of decisions; progress measures	Project lead, product owner and architect

Table 4. Who decides what at each level.

Firm rules should be built into the platform rather than written into a handbook alone: projects receive them automatically when they start. In the other direction, a project that builds something reusable, such as a new system connection or a template, hands it to its department and from there to the firm, after review. This “inner source” model¹⁴ lets the platform grow without the central team doing all the work.

In practice, the firm builds a minimum central layer first, then the first department layer for the pilot. The second department reuses the central layer and only adds its own knowledge.

07 Starting now, before the platform is ready

Building the full platform takes time, often a year or more until it covers many teams. Firms should not wait. A great deal can be done today with an approved AI assistant and a few clear rules, as long as one principle holds: **AI prepares drafts within the existing process steps; it never replaces an approval.**

Figure 5 shows three stages. The first two need no agent platform.

¹⁴ Inner source: applying the working methods of open-source software inside a company. Teams can use and improve each other's components, and a central owner reviews contributions before they become standard.



The outputs of stages A and B — rule catalogues, specifications, prototypes, test cases — are exactly the inputs stage C needs.

Figure 5. Three stages of adoption. Stages A and B can start immediately and prepare the ground for stage C.

In **stage A**, individuals use the firm’s approved AI assistant for analysis and drafting and copy the results into the normal project tools. In **stage B**, teams share tested instructions, templates and reference documents in shared workspaces of the assistant, so results become consistent and repeatable. **Stage C** is the agentic delivery described in this paper, where agents work directly in company systems.

What teams can do now

Task	What the AI does	Who checks
Extracting business rules	Reads old program code and spreadsheet macros and writes down every business rule in plain language, with a pointer to where it is found	A business expert confirms each rule
Designing screens	Turns requirements and screenshots of old screens into screen layouts and clickable prototypes ¹⁵	Future users try them out; design and accessibility review
Writing requirements	Turns workshop notes into precise requirements with acceptance criteria and open questions	Product owner and business
Designing tests	Derives test cases and edge cases from the requirements	Quality engineer
Documenting old systems	Explains undocumented code and spreadsheet tools, including risks	Owner of the system

Table 5. Useful tasks that need only an approved AI assistant.

Example: business rules. Before any migration, someone has to find out what the old system really does. A team exports the old program code as text, removes any passwords or client data, and asks the assistant, module by module, to list every rule in a fixed template: number, description, formula, conditions and source reference. A second request lists the code that no rule covers, so gaps become visible. Business experts then confirm each rule. The result is exactly the rule catalogue that an agentic project needs at its first approval gate, so the later project starts weeks ahead.

Example: screen design. New screens are hard to describe in words. Given the requirements, screenshots of the old screens and the firm’s design system¹⁶, an assistant produces layouts and clickable prototypes within hours, including several variants for users to compare. The approved prototype becomes the specification. It is not the finished application, which is built later in the normal way. Screenshots and examples must never contain real client data.

¹⁵ A clickable prototype looks and behaves like the future application on screen, so users can try it out, but it has no real data or functions behind it.

¹⁶ A design system is a firm’s set of rules and building blocks for user interfaces: colours, fonts, buttons, forms and layout rules, so that all applications look and behave alike.

Minimum rules for an early start

- **One approved tool** under an enterprise contract: no use of the firm's data for model training, data kept in the EU where required, central administration. Private accounts are shadow IT¹⁷ [6].
- **A clear data rule:** what may be given to the assistant (usually code and documentation) and what may not (client and personal data).
- **A named person** who reviews and signs every result, exactly as without AI.
- **Labelling and traceability:** AI-assisted drafts are marked, and the instructions used are kept with them.
- **Shared templates** for each type of task, which is what moves a team from stage A to stage B.
- **A simple register** of AI-assisted work, so risk and compliance keep an overview and evidence for the business case builds up.

These early stages have limits: results are copied by hand, the assistant cannot see company systems, and the record is only as good as what people save. When volumes grow, when several teams do the same work differently, or when system-generated evidence is needed, it is time for stage C.

08 Making it real

The move to agentic delivery is best done in three steps, with a management decision between each. Figure 6 summarises the roadmap.



G1: decision to run the pilot once the foundation is in place · G2: decision to scale, based on measured pilot results

Figure 6. A phased roadmap with two management decision points.

¹⁷ Shadow IT: technology used by staff without approval or control by the IT function. BaFin expects regulated firms to prevent it.

The first 90 days

- 01 Appoint a sponsor and a platform owner.**
A senior sponsor with budget authority and a product owner for the agent platform.
- 02 Agree the delivery model and controls with risk and compliance.**
Phases, gates, autonomy levels and the evidence to be produced. Involve the data protection officer early.
- 03 Allow early AI-assisted work.**
Approve one enterprise AI assistant and minimum rules (Section 7) so teams can start extracting business rules, drafting requirements and designing screens.
- 04 Choose model and cloud providers.**
Prefer providers already assessed under DORA; confirm EU data residency and that your data is not used to train models.
- 05 Build the minimum platform.**
Model gateway, agent workspace, identity, first read-only connections, audit trail. A team of four to five, with part-time help from risk and compliance, can do this in about three months.
- 06 Select and prepare the pilot.**
A contained migration with a clear reference, such as a reporting platform. Secure business experts' time.
- 07 Train the first team.**
Engineers, analysts and testers learn to write precise instructions and to review agent work.
- 08 Define success measures.**
Agree targets before starting (see below) so the decision at G1 and G2 rests on evidence.

Who is needed

A small platform team builds and runs the platform: a product owner, two or three platform engineers and a security engineer, with part-time support from IT risk, compliance and data protection. Each delivery team then needs intent owners, engineers who supervise agents and a quality engineer. New skills are needed above all in writing precise specifications and in reviewing at scale. Firms should also plan how junior staff will build experience when agents take over routine tasks.

What to measure

Measure	What it tells management
Lead time from approved requirement to production	Whether delivery is actually faster
Share of changes that fail after release	Whether speed costs quality
Human review time per change	Whether reviewers are a new bottleneck
Defects found after release	Whether reviews and tests catch enough
Cost per delivered change, including AI usage	Whether the economics work
Open differences between old and new system (migrations)	Progress towards a safe switch-over

Table 6. Suggested measures for the pilot and for scaling decisions.

What it costs

The main cost items are the platform team, AI model usage (billed by volume), licences for agent tools and connectors, and training. Usage costs can be controlled per team through the gateway. Against this stand shorter projects, smaller teams for routine work and, in migrations, the removal of recurring manual fixes.

09 Risks and how to manage them

Risk	How it shows	How to manage it
Vague instructions	Agents build the wrong thing quickly and confidently	Precise, testable rules; business sign-off before build
Review overload	People approve large volumes of AI output without real checking	Small units of work; automated checks; independent review agents; plan reviewer capacity
Too much autonomy	Agents act outside their task or reach live systems	Autonomy levels enforced by the platform; minimum rights; sealed workspace
Data leakage	Confidential or client data reaches AI models	Gateway with data-loss controls; masked test data; EU-hosted models
Manipulated input	Instructions hidden in documents or tool output mislead an agent (prompt injection ¹⁸)	Read-only access by default; vetted connections; human gates
Dependence on one provider	Hard to switch model or tool vendor	Tool-neutral method; documented exit plan as DORA requires
Loss of skills	Fewer junior roles; future seniors lack experience	Structured apprenticeship in supervising and reviewing agents

Table 7. Principal risks.

¹⁸ Prompt injection: an attack in which hidden instructions are placed in content that an AI reads, such as a document or web page, to make it do something it should not.

Glossary of abbreviations

AI	Artificial intelligence	FINOS	Fintech Open Source Foundation, an industry body
AI-DLC	AI-Driven Development Lifecycle, a delivery method published by AWS	IT	Information technology
AWS	Amazon Web Services, Amazon's cloud	LLM	Large language model, the AI behind assistants and agents
BaFin	German Federal Financial Supervisory Authority	MCP	Model Context Protocol, a standard for connecting agents to software
BCG	Boston Consulting Group, a consultancy	PBI	Product backlog item, a unit of work in agile projects
BMAD	Open-source method mapping agile roles to AI agents	SAP	German enterprise software vendor
CIO	Chief Information Officer	SAP BW	SAP Business Warehouse; SAP BW/4HANA is its newer generation
DORA	Digital Operational Resilience Act: EU rules on IT risk in financial firms, since 2025	SAP HANA	SAP's in-memory database and its cloud service
EU	European Union		

Sources

Key sources for this edition. All product details and the full list of 66 sources are in the detailed FINATUM white paper of the same title.

- [1] Fortune, "Goldman Sachs doesn't have to hire a \$180,000 software engineer — meet Devin", July 2025.
<https://www.fortune.com/2025/07/14/goldman-sachs-ai-powered-software-engineer-devin-new-employee-increase-productivity-fears-of-job-replacement>
- [2] Quartz, "Goldman Sachs just hired its first AI coder", July 2025.
<https://qz.com/goldman-sachs-ai-coder-devin-cognition>
- [3] ai2.work, "Goldman Sachs puts hundreds of AI software engineers on dev teams", 2026.
<https://ai2.work/blog/goldman-sachs-puts-hundreds-of-ai-software-engineers-on-dev-teams>
- [4] The Cryptonomist, "Goldman Sachs Devin AI", August 2026.
<https://en.cryptonomist.ch/2026/08/06/goldman-sachs-devin-ai/>
- [5] Jones Day, "BaFin's expectations for ICT risk management and the use of AI", January 2026.
<https://www.jonesday.com/en/insights/2026/01/bafins-expectations-for-ict-risk-management-and-the-use-of-ai>
- [6] Famulor, summary of BaFin guidance on AI under DORA, 2026.
<https://www.famulor.io/blog/voice-ai-in-banking-bafin-and-dora-rules-for-2026>
- [7] FINOS, "AI Governance Framework v2.0 — addressing agentic AI risks", November 2025.
<https://www.finos.org/blog/finos-ai-governance-framework-v2.0-addressing-agentic-ai-risks-in-a-rapidly-evolving-landscape>
- [8] AI-Driven Development Life Cycle (AI-DLC) overview.
<https://agentic-ai.readthedocs.io/en/latest/Standards/aidlc/>
- [9] CodeMySpec, "The BMAD Method explained".
<https://codemyspec.com/blog/bmad-method-explained>
- [10] Thoughtworks Technology Radar, "Spec-driven development", November 2025.
<https://www.thoughtworks.com/radar/techniques/spec-driven-development>
- [11] McKinsey, "Beyond the copilot: Scaling the agentic product development life cycle".
<https://www.mckinsey.com/capabilities/tech-and-ai/our-insights/beyond-the-copilot-scaling-the-agentic-product-development-life-cycle>
- [12] Boston Consulting Group, "The Agent is the Easy Part", September 2026.
<https://web-assets.bcg.com/65/7e/bcffe4df494db793dce490db7f9/the-agent-is-the-easy-part.pdf>
- [13] BARC, "End of maintenance for SAP BW: what's next?"
<https://barc.com/end-of-maintenance-sap-bw/>