

6 decisions that determine implementation success

AGENTIC AI IN PRACTICE

How to validate the quality, cost, and scalability of agentic systems before going to production

Executive Summary

Agentic AI is becoming one of the key directions in the evolution of enterprise artificial intelligence. More and more organizations are exploring how agentic systems can support processes that require specialized expertise, analysis of large volumes of information, and complex decision-making.

At the same time, organizations face a fundamental question: how can they distinguish an impressive demonstration from a solution that will actually perform in a production environment? A system may successfully analyze a sample document and produce the expected output, only to deliver significantly lower-quality results when given material with a different structure, length, or level of complexity.

In practice, the difference between a successful pilot and an effective Agentic AI implementation rarely comes down to framework selection alone. Decisions around the model, data, architecture, process controls, accountability for outcomes, and solution economics have a far greater impact.

Our experiment showed that successful Agentic AI implementation is less about selecting a single best tool and more about deliberately designing how quality will be assessed, costs controlled, and responsibility for outcomes assigned.

The purpose of this material is not to identify one best model or framework. Instead, it explains how to approach Agentic AI so that a solution can be evaluated, controlled, and scaled – and when deciding not to deploy an agentic architecture may be the best possible outcome of the analysis.

Key Findings

- A working Proof of Concept does not mean a solution is production-ready
- Model selection has a direct impact on solution quality and economics
- A business taxonomy is essential for repeatability
- Operational complexity increases with the number of agents and dependencies
- A larger context window is no substitute for an information management strategy
- Human-in-the-loop is a mechanism for controlling risk and cost
- Agentic AI costs should be measured at the individual process level
- Agentic AI can support decision-making rather than replace experts

From pilot to production

Where the real challenges emerge?

Agentic system demonstrations can perform exceptionally well under controlled conditions. A model analyzes a document, draws conclusions, and generates the expected output. The true value of the solution, however, only becomes apparent when the system receives a new document with a different format, length, and structure.

This is one of the fundamental challenges of moving from a demonstration to a real-world application. A solution tuned to the material used during development may perform significantly worse on data it has never processed before. This does not necessarily mean the implementation itself is flawed. The underlying issue may simply be that the system was never tested under genuinely new conditions.

A production implementation differs from a pilot in three fundamental ways:

- the scale of the documentation being processed,
- the variability of input data,
- clearly assigned accountability for the outcome.

Each of these introduces a different set of risks. Scale affects processing time and cost. Data variability tests the system's ability to generalize. Accountability, meanwhile, requires an understanding of how the result was produced and who is authorized to approve it.

> "An agentic system demo looks impressive. The problem starts the next day."

How we ran the experiment



Objective

The purpose of the experiment was to determine whether different multi-agent architectures could produce a useful solution concept based on complex documentation for an enterprise system implementation.

The test was designed not only to determine whether the task could be completed, but above all to assess whether the solution could perform on new data that the teams had not seen while designing the system.

We selected solution concept development as the test scenario because it reflects the complexity of expert-driven processes particularly well. It requires analyzing large volumes of information, identifying requirements, detecting gaps, developing alternatives, assessing risks, and formulating recommendations that experts can subsequently validate.

Participants and team setup

The experiment involved 23 people from Sii's Digital Competency Center. They were divided into five cross-functional teams comprising architects with experience across different projects and technology stacks, as well as varying levels of familiarity with language models.

Process

The experiment lasted two days. On the first day, the teams were introduced to the challenge on-site and developed their solutions using two sets of anonymized and publicly available materials describing requirements for enterprise system implementations. They did not work from ready-made templates prepared in advance for a specific use case.

On the second day, the teams received two new document sets that they had not seen before and that varied in complexity. They were not allowed to tune their prompts for the specific case. Only the outputs generated from these new data sets were submitted to the technical and business review panel.

Evaluation

The results were assessed by a five-person panel: three reviewers focused on technical aspects, while two evaluated the outputs from a business perspective.

Key quantitative result

All five teams successfully ran their complete process flows on both new document sets without tuning prompts for the specific case. Each team generated material that was submitted to the technical and business panel for evaluation.

What does success in Agentic AI actually mean?

A functioning end-to-end process flow is not, by itself, sufficient evidence of business value. Evaluating an Agentic AI project requires several perspectives to be considered together.

The business case should include at least four metrics:

- a composite quality score covering factors such as completeness of requirements, accuracy of recommendations, and risk coverage,
- the time required to generate the result plus the time an expert needs to review it,
- the cost of processing a single case,
- the number of human interventions required.

Only by looking at these metrics together can an organization determine whether the system reduces process time and cost while maintaining the required quality level.

6 decisions to make before scaling Agentic AI

1 Select the model based on quality, cost, and risk

This was the most consistently repeated conclusion from the experiment and the one most clearly visible during the presentation of results. The same architecture and the same pipeline, when run with a different model, could produce an output of completely different value.

> “Depending on the model we use, the same solution can range from ‘this seems useful’ to ‘this is completely unusable.’”

For an organization, the implications extend well beyond response quality. The model affects the unit cost of the process, where data is processed, the degree of vendor lock-in, and the ability to meet regulatory requirements.

In many projects, the model is selected only once, during the Proof of Concept stage. The choice may be driven by access to a particular API key, the team’s previous experience, or simply how easy the solution is to launch. That same model may then move into production without being reassessed.

We recommend reversing this sequence. The first step should be to create a test set using the organization’s own representative documents. The complete pipeline should then be run with several models, comparing output quality, cost, and execution time. Only then should the final decision be made.

Model selection should therefore be based on three metrics measured against the organization’s own data:

- output quality,
- unit cost,
- execution time.

The test set may ultimately prove to be a more durable asset than the model itself. Models and model versions will change, but the organization’s ability to determine whether a new solution delivers a better result will continue to provide value.

2 Without a taxonomy, there is no repeatability

A shared vocabulary turns unstructured text into structure. That structure can then be validated, versioned, and compared across successive runs. Without it, every output remains an isolated event that cannot be reliably compared with the one before it.

This is a prerequisite for any audit. If an organization does not clearly define concepts such as "non-functional requirement," "risk," or "integration," the model may interpret them differently on each run.

The consequences are tangible. Without a shared conceptual model, it is impossible to:

- reliably detect regressions,
- compare two solution alternatives,
- assess changes in quality between versions,
- demonstrate to a client or auditor that the process remains under control.

The taxonomy often already exists within the organization, but is scattered across knowledge platforms, proposal templates, spreadsheets, and the expertise of individual specialists. An agentic implementation requires that taxonomy to be identified, structured, and formalized.

> "The domain model is an asset, not a cost."



3 Every additional agent increases operational complexity

Building a single agent is an engineering task with relatively well-defined complexity. Coordinating several agents is considerably more challenging. It requires managing contracts between them, passing state, maintaining naming conventions, and handling invalid or incomplete outputs.

Two agents with a straightforward handoff may not be significantly more difficult to manage than one. The real challenges emerge as the number of agents grows and feedback loops, parallel processing, and task renegotiation mechanisms are introduced.

Complexity can then increase nonlinearly. As it does, so do: the variance in estimates, the cost of diagnosing errors, the time required to explain why the system produced a particular result.

Explainability becomes particularly important after the first failed run. If the system does not provide an execution trace that allows each stage of the process to be reconstructed, the organization can quickly lose confidence in the solution as a whole.

We therefore recommend starting with the smallest topology capable of solving the problem. Another agent should only be added when its role, responsibilities, inputs, and expected output can be clearly defined.

4 A larger context window does not solve the problem

The limitations of context became especially clear when the teams worked with the more extensive document set on the second day of the experiment. Passing all available material in a single prompt can simultaneously increase cost and reduce output quality.

The continued expansion of context windows may have created the impression that large volumes of information can simply be handled by increasing model capacity. In practice, however, a system overloaded with data may struggle to prioritize information correctly and may treat conflicting passages as equally important.

Enterprise documentation creates additional challenges. It may contain repetition, internal inconsistencies, and sections prepared by different people at different times. Excess content is not an exception—it is common.

Understanding the nature of the documents makes it possible to select appropriate context management mechanisms, such as:

- RAG instead of passing all content directly into the prompt,
- prompt caching,
- preliminary structuring of documents,
- removal of unnecessary content,
- chunking based on semantics rather than character count alone.

Context strategy also has another dimension. The moment information is passed to a prompt may be the moment it leaves the organization's perimeter. The scope of the information exposed to the model should therefore be designed with a clear understanding of the associated level of exposure.



5 Human-in-the-loop is a mechanism for controlling risk and cost

Human involvement in a process is sometimes interpreted as a limitation on automation. Our experience shows, however, that a well-positioned control point can reduce the number of unnecessary alternatives and, in turn, the time and cost of further processing..

The principle is to stop unjustified lines of analysis before the system develops them into fully fledged alternatives. The value of a checkpoint can then be measured through:

- the number of alternatives stopped,
- the cost of model calls avoided,
- the time required to make the decision.

There is also a consideration that technology alone cannot resolve. In processes with material business impact, accountability for the outcome must be clearly assigned to a human. Saying that an agent made the decision is not a governance mechanism.

The architecture can be supplemented with a parallel evaluation module that automatically assesses output quality and flags deviations. The decision to continue the process, however, should remain with a person.

“A checkpoint at every step is not oversight. It is paralysis.”

6 Token cost is an operating budget line item

For a single run, the amount may be relatively small. At the scale of document analysis, ticket handling, or contract review, however, it becomes a cost that must be reflected in the budget.

The relevant metric is not simply the monthly cost of the tool, but the unit cost of handling a specific case.

The cost of using models can be harder to predict than the cost of a traditional software license. It varies with model pricing, the number of repeated calls, and the length of the sequence of operations performed by agents. Running the system without appropriate limits creates both the risk of escalating costs and the risk that the process will be terminated before completion.

> “Token cost is a budget item, not background noise.”

The available optimization techniques are well known and can be effective. They include:

- selecting a model appropriate to the scale and complexity of the problem,
- deliberate context management,
- caching,
- cost and operational limits for individual runs.

The prerequisite, however, is to measure the cost of an individual case from the beginning of the process. Cost should be assessed together with quality, not instead of it.

An agentic architecture will not always make economic sense. For some processes, token costs may exceed the value of the expert time saved. Establishing this before deployment is one of the most valuable outcomes an analysis phase can deliver.



From documents to decision

Where Agentic AI creates the most value?

An experienced architect may spend several days completing this type of process. That provides a natural benchmark for further testing. The expert's working time should be compared with both the time required to generate the material and the time needed to review and correct the AI-generated output. Only this comparison reveals the actual time savings.

The fundamental question behind the experiment was how much of the process could be delegated to an agentic system and where the boundary should be drawn.

The most useful answer is neither full automation nor complete rejection of the technology. In our tests, agents showed the greatest potential in: reading documentation, normalizing information, identifying contradictions, detecting gaps, preparing initial alternatives.

These are tasks for which clear quality criteria can be established while leaving the expert responsible for assessing whether the proposed solution actually makes sense.

Agentic AI as a decision validation mechanism

One of the most interesting applications of Agentic AI does not necessarily involve automatically developing a solution from scratch.

An agentic system can serve as an additional validation mechanism. It can review an existing concept, compare it with requirements, identify overlooked risks, and generate questions an expert should ask before making a decision.

From the client's perspective, this could mean:

- independently validating an architecture proposed by a vendor,
- comparing technology alternatives,
- analyzing a migration plan,
- preparing questions before making a costly decision.

The same pattern can be applied to:

- tender documentation analysis,
- due diligence,
- claims and complaint handling,
- vendor assessment,
- contract compliance reviews,
- onboarding,
- analysis of legacy system technical documentation before migration.

The potential value does not come from replacing people. It comes from transferring repetitive elements of the analysis to the system while leaving validation and decision-making with the expert.

Whether this approach actually reduces delivery time and cost must be measured for each specific process.

Framework for preparing and validating an Agentic AI implementation

The six decisions outlined above can be translated into a practical framework for preparing and validating an implementation.

Discover as the starting point

Before assessing a client's readiness or selecting a model, we begin by understanding the client's business. We analyze the available data, processes, and actual problems to identify areas where an agentic architecture could generate measurable value.

This stage defines the project scope and establishes the success criteria for the subsequent areas of support.

Readiness assessment and model selection

We build a test set using the client's documents and run selected models against it, measuring both quality and cost. The client receives a decision based on its own data, together with a mechanism that allows the assessment to be repeated when a new model or model version becomes available.

Domain model and taxonomy

We identify and formalize the organization's vocabulary. We then design output schemas and validation rules. This is essential to ensuring repeatability and auditability across the entire process.

Orchestration design and observability

We select the smallest topology that solves the problem. We define contracts between agents and ensure that an execution trace is available from the very first run.

Context and data strategy

We design how information is passed to the model. This may include RAG, caching, preliminary document structuring, and defining the boundary of the data perimeter. Where information cannot leave a controlled environment, we assess the option of using a dedicated endpoint with an open-weight model.

Human-in-the-loop and evaluation

We place decision points where they have the greatest impact on cost and risk and complement them with mechanisms that automatically evaluate output quality.

FinOps for AI

We model the cost of handling an individual case before implementation. This includes the number of model calls, context length, potential retries, escalation scenarios, and the expert time required to review the result.

We establish cost and operational limits for individual runs, together with monitoring that quickly identifies situations in which the system begins generating too many steps, alternatives, or expensive queries.

The most important aspect of FinOps for AI, however, is the willingness to reach a clear conclusion: if the economics do not work, the agentic architecture should not move into production.

> "Sometimes the most valuable outcome of the analysis is the decision not to deploy agents."

Experience matters more than expertise in a single framework

In the experiment, five teams built different solutions and ran them on new data without tuning their prompts for a specific document. This kind of hands-on experience makes it possible to design quality tests, safeguards, observability, and a cost model before a solution is introduced into a production environment.

When Agentic AI is not the right choice

An agentic architecture is not the optimal solution for every process.

In some cases, the cost of using models may exceed the value of the time saved. In others, the additional complexity of orchestration, observability, controls, and system maintenance may not be justified by the nature of the task.

Before beginning an implementation, organizations should therefore answer several fundamental questions:

- Is the process sufficiently repeatable?
- Can clear quality criteria be defined?
- Does the organization have representative test data?
- Does the value of the time saved exceed the cost of running and maintaining the solution?
- Can it be clearly established which decisions remain with humans?
- Could simpler automation solve the problem at lower cost and with less risk?



An implementation that passes the reality check

Agentic architectures continue to evolve rapidly. They can generate substantial costs, produce outputs of varying quality, and change faster than the documentation describing specific approaches can keep up.

This is not an argument against experimentation. It is an argument for conducting experiments with a team that has tested different approaches in practice and knows where the most common problems emerge.

This knowledge cannot be built from presentations or documentation alone. It requires building systems, running them against new data, and presenting the results to people responsible for technology and project delivery.

Organizations do not have to choose between speed and control. They can achieve both if they make the six decisions described above deliberately:

- select the model based on their own data,
- structure the domain model,
- minimize unnecessary architectural complexity,
- design a context and data strategy,
- establish the right human control points,
- measure cost at the business process level.

The objective is not to deploy agents for their own sake. The objective is a better decision, faster analysis, or a lower process cost while maintaining human control and accountability.

“We do not deploy agents. We support business processes.”

Assess your organization’s readiness

Want to find out which processes in your organization are genuinely suited to agentic automation? Let’s discuss your readiness, data quality, potential architecture, and the solution’s cost model.

Learn more



Dariusz Adamowski
Competency Center Director