REPORT

From AI idea to pilot. A guide

How to turn an AI idea into a pilot that leads to a real decision. Scale it, rework it or deliberately stop.

Report
Written by Emanuel Stadler

Optimentum

About this report

This report is the full edition of the article of the same name on optimentum.eu. It walks through every phase, with criteria, worksheets, checklists and templates for your own work. Examples are illustrative and do not describe client engagements. Legal notes do not replace legal advice.

Report
Written by Emanuel Stadler

Optimentum GmbH · Tegetthoffstraße 3/10, 1010 Vienna, Austria · FN 686237y, Commercial Court of Vienna
optimentum.eu · office@optimentum.eu

Contents

Summary

01 Why pilots fail

02 Choosing the right task

03 Decision question and criteria

04 Data, law and security

05 People and accountability

06 Running the pilot

07 Evaluating and deciding

08 From pilot to operation

Appendix

SUMMARY

KEY FINDINGS

01 A pilot is an instrument for a decision. Without a question defined upfront, it produces impressions rather than a basis for deciding.

02 Data, ownership and processes decide whether a pilot holds up, just as much as the technology does.

03 Raising information security and privacy only at the end risks a pilot stalling just before it goes live.

04 Human oversight is designed together with the process. It shapes risk and value alike.

05 The prerequisites for operation belong in the planning of the pilot, not at its end.

KEY TAKEAWAYS

- Before you start, define the question the pilot should answer.
- Set success and stop criteria that can actually be tested.
- Test with real cases, including difficult ones, not just prepared examples.
- End the pilot with a decision, not a presentation.



CHAPTER 01

Why pilots fail

From AI idea to pilot. A guide

A prototype of an AI application can be built in a few days today. It shows that a model can, in principle, handle a task. It does not show whether the solution holds up in everyday work, whether it can be operated responsibly or whether the effort pays off. A pilot is meant to close this gap.

In practice, it often does not. Many pilots end with a positive impression, a presentation and no decision. The solution remains in limbo, neither used nor retired. This chapter describes why that happens and what a pilot has to deliver instead.

Prototype, proof of concept and pilot

The three terms are often used interchangeably. Yet they answer different questions, and anyone who confuses them expects something from a project that it cannot deliver.

A **prototype** makes an idea tangible. It serves to share a concept and gather feedback. A prototype may be incomplete. It usually works with a few, often prepared examples and runs outside the actual systems.

A **proof of concept** tests whether something is technically possible. Can a model reliably extract the required information from these documents? Can the source system be connected? A proof of concept answers a narrowly defined technical question with yes or no.

A **pilot** tests whether a solution adds value in the actual process and can be operated responsibly. It works with real cases, with the people who do the task today and with criteria set in advance. Its result is a decision on whether to continue.

Characteristic	Prototype	Proof of concept	Pilot
Guiding question	What could it look like?	Is it technically possible?	Does it pay off in everyday work, and can it be operated responsibly?
Cases	A few, often prepared examples	Selected test data	Real, representative cases, including difficult ones
People involved	Small project team	Technical team	Business unit, users, IT, security, privacy, decision-makers
Environment	Separate from operations	Test or lab environment	Within the process or alongside it
Criteria	Usually none	Technical feasibility	Quality, review effort, acceptance, operation, risk
Result	Impressions and feedback	Yes or no to the technical question	Decision: scale, rework or deliberately stop

The three forms build on one another, but they are not all required in sequence. If technical feasibility is obvious, a proof of concept can be skipped. A pilot, however, cannot be replaced by a particularly convincing prototype.

Typical patterns of failure

Pilots rarely fail because of the technology alone. More often, they fail because of questions nobody asked before the start. The following patterns occur on their own or together.

The question is missing

The pilot starts with a mandate to try out a technology. Which decision should be made at the end has not been defined. The results can then be read in many ways, and every side finds its view confirmed.

Prepared examples instead of real cases

The solution is tested with carefully selected, clean examples. Incomplete inputs, special cases and poorly legible documents are missing. The solution looks better in the pilot than it will be in everyday use.

The comparison is missing

Nobody has recorded how long the task takes today and how many errors occur. The pilot ends with a number that has no point of reference. Whether the solution is better than the current process cannot be said.

The scope grows quietly

During the pilot, further document types, teams or subtasks are added. Each extension seems small. Taken together, the pilot ends up measuring something other than what was planned, and the criteria no longer fit.

Security and privacy come too late

Questions about data access, processing location and legal basis are only raised when the solution is due to go live. By then, the architecture and the vendor have already been chosen. The pilot stalls at exactly the point where it was supposed to add value.

Nobody is responsible

The pilot is carried by an innovation team or an external service provider. There is no process owner in the business unit. After the pilot, it is unclear who takes over the solution and who decides on its future.

The presentation replaces the decision

The pilot ends with a presentation of the results to a wider audience. There is approval and there are suggestions, but no resolution. The solution keeps running, without a budget, without operational ownership and without an end.

What a pilot is for

A pilot is neither a scaled-down rollout nor a demonstration. It is an instrument for preparing a decision. That decision concerns an investment, a change to how work is done and a responsibility for operation.

> A pilot is an instrument for a decision. Without a question defined upfront, it produces impressions rather than a basis for deciding.

Three requirements follow from this. First, before the start, it must be clear which question is to be answered and how the answer will be measured. Second, the pilot must run under conditions close to later everyday use: real cases, real users, real systems or a realistic replica. Third, at the end, a person or body with decision-making authority must choose one of three options: scale, rework or deliberately stop.

A deliberately stopped pilot is not a failure. It has answered a question and prevented further budget from flowing into a solution that does not hold up. A pilot has only failed if it does not make a decision possible.

How this guide is structured

The following chapters describe the path from selecting the task to the move into operation. They follow the order in which the questions should be answered. Chapters 2 and 3 clarify what is tested and against what. Chapters 4 and 5 cover data, law, security and the role of people. Chapter 6 describes the pilot run itself, chapter 7 the evaluation and decision, chapter 8 the move into operation.

The appendix contains checklists for each phase, templates to fill in and a glossary. The templates are deliberately simple. They can be transferred into existing project tools and adapted to your own organisation.

The guide starts at the point where an AI use case has been selected in principle. Prioritising between several use cases is only touched on where it affects the choice of the specific task.

At a glance

- A prototype shows what is possible. A pilot shows whether it pays off in everyday work and can be operated responsibly.
- Pilots usually fail because of missing questions, missing comparisons and missing ownership, less often because of the technology.
- The result of a pilot is a decision, not a presentation.
- Deliberately stopping is an equally valid outcome if it is well founded.

CHAPTER 02

Choosing the right task

From AI idea to pilot. A guide

A pilot starts with a concrete task, not a technology. Even if the use case has already been chosen, it is usually too broad to be tested meaningfully in a pilot. “AI in customer service” is a field of application. “Assign incoming emails to a category and route them to the responsible team” is a task.

This chapter shows how to turn a use case into a testable task, which criteria to use when choosing between several candidates and how to record the result in a task profile.

From idea to task

A task is concrete enough when an outsider can understand it without asking. To get there, describe five points:

- **Who** does the task today, and how many people are involved?
- **What inputs** are available, in what form and from which systems?
- **What output** is produced, and in what form is it passed on?
- **Who works** with the output next, and what happens if it is wrong?
- **How often** does the task occur, and how much does the volume fluctuate?

An example: an insurer checks incoming claim notifications. Today, a claims handler reads each notification, assigns it to a type of claim, checks whether the documents are complete and requests missing information. For a pilot, this description is still too broad. A suitable task would be: “For incoming property damage notifications, identify missing mandatory information and draft the request for it.” Input, output and the person who works with it next are now clear.

It helps to treat this description like a requirement. It is agreed with the stakeholders involved, written down and changed only deliberately.

Selection criteria

Often several tasks are available to choose from. The following criteria help to find the one best suited to a pilot. Suited does not mean that the task promises the greatest conceivable value. Suited means that the pilot can deliver a clear answer.

Value

The task should account for a noticeable share of time, waiting time or errors. Ask what changes for the people involved and for the downstream steps if the task is done faster or more reliably. Value that only shows up in a cost calculation rarely convinces the people involved.

Feasibility

The required inputs must be available digitally or be digitisable with reasonable effort. The systems the data comes from and the output flows into should be accessible. Also check whether the task occurs often enough to collect sufficient cases during the pilot period.

Risk and error tolerance

How serious is a faulty output? Can it be reversed, and how quickly is it noticed? For a first pilot, tasks are suitable where an error has limited consequences and can be detected before the output is used further. Tasks that involve decisions about people, trigger payments or communicate externally require more control and a longer lead time.

Measurability

It must be possible to assess the output of the task. This is easiest when there is a right answer, such as a category or an amount. It is harder with texts such as summaries or draft replies. There, you need assessment standards agreed in advance and people who apply them.

Ownership and backing

The task needs a process owner who carries the pilot on the business side and shares in the final decision. Equally important is a team that is willing to take part in the pilot run. A task without these prerequisites is unsuitable for a pilot, even if it looks promising from a business perspective.

AI or conventional automation

Not every task needs AI. If rules are clear and inputs are predictable, conventional automation is often easier to operate and to verify. It behaves the same way for the same input, and its logic can be read.

AI can make sense when unstructured content has to be processed or variable situations have to be classified: free text, documents from different sources, requests phrased in different ways. Often a combination works best. The AI reads and classifies, and the subsequent steps run on rules.

So break the task into steps and check, for each step, whether a model, a rule or a person is best suited to it. The pilot then concentrates on the steps where the use of AI is genuinely an open question.

Assessment matrix

If several tasks are available, a simple matrix helps. Score each task per criterion on a scale of 1 to 3 and weight the criteria according to your organisation’s priorities. Set the weighting before scoring so that it is not adjusted to fit a desired result.

Criterion	Guiding question	1 = low	2 = medium	3 = high	Weighting
Value	How much would a better solution relieve the people involved and the downstream steps?	Barely noticeable	Noticeable for one team	Noticeable across several steps	define
Feasibility	Are inputs available digitally, and are the systems accessible?	Data missing or hard to access	Accessible with effort	Available and accessible	define
Error tolerance	How limited and detectable are the consequences of an error?	Serious consequences, hard to detect	Limited consequences, detectable with review	Minor consequences, easy to correct	define
Measurability	Can the output be assessed unambiguously?	Only subjectively	With agreed standards	Clearly right or wrong	define
Case volume	Will enough cases occur during the pilot period?	Rarely	Regularly	Frequently	define
Ownership	Is there a process owner and a participating team?	Open	Named, but rarely available	Named and available	define

The matrix does not replace a decision. But it makes visible why a task was chosen, and it shows weaknesses that deserve particular attention during the pilot. A task with high value and low measurability can still be the right one if it is clarified in advance how the results will be assessed.

Limiting the scope

Limit the scope deliberately. A pilot covering one process step in one team yields clearer insights than one that tries to change several processes at once. State explicitly what is out of scope: further document types, further locations, further languages, adjacent process steps.

This boundary protects the pilot from growing quietly. It also makes communication easier. Anyone who asks why an obvious extension is missing gets a clear answer: it has been deliberately excluded and can be examined after the decision.

WORKSHEET: TASK PROFILE

Field	Content
Name of the task	…
Use case the task belongs to	…
Current process in a few sentences	…
Inputs (form, source, system)	…
Output (form, recipient)	…
Who works with the output next	…
Frequency and fluctuation of volume	…
Consequences of a faulty output	…
Steps for AI, rules and people	…
Out of scope	…
Process owner	…
Participating team	…
Score in the matrix	…
Open questions	…

At a glance

- A use case is not yet a task. Describe inputs, output and the people involved precisely enough for an outsider to understand.
- Choose the task for which the pilot can deliver a clear answer, not necessarily the one with the greatest conceivable value.

- For each step, check whether a model, a rule or a person is best suited to it.
- Write down what is out of scope.

CHAPTER 03

Decision question and criteria

From AI idea to pilot. A guide

Before the pilot begins, it is clear which question it should answer and how the answer will be measured. That sounds obvious. In practice, however, criteria are often only formulated after the first results and then unconsciously adjusted to fit them.

This chapter describes how to formulate the decision question, record the current process as a point of comparison and set success and stop criteria that can be tested.

The one question

A pilot answers one question. It is addressed to the person or body that decides at the end, and it can be answered with the three possible outcomes: scale, rework or deliberately stop.

A good decision question names the task, the benchmark for comparison and the conditions of operation. An example: “Can the solution identify missing mandatory information in property damage notifications reliably enough to reduce review effort compared with today, without more incomplete cases entering processing, and can it be operated in the existing system?”

Weak questions give themselves away by being open-ended. “What can AI do in the claims process?” cannot be answered. “Does the model work?” is a technical question that belongs in a proof of concept. “Are the users satisfied?” picks out one aspect and ignores effort and risk.

Write the question down and agree it with the person who will decide. If that person does not recognise the question as their own, they will not follow the answer at the end either.

Recording the current process

Without a starting point, there is nothing to compare against. So before the pilot begins, record how the task is done today. This record is called the baseline. It does not have to be perfect, but it must be measured in the same way as the solution will be later.

Typical baseline measures are:

- **Effort per case:** the working time a person spends on one case.
- **Lead time:** the time from receipt to finished output, including waiting times.
- **Error rate:** the share of cases in which the output later has to be corrected.
- **Rework:** queries, follow-up requests and corrections in downstream steps.
- **Volume:** the number of cases per unit of time and its fluctuation.

Three approaches are suitable for recording it. First, analysing existing system data, such as timestamps in a ticketing or case management system. Second, a short, supported time recording over a defined period. Third, a retrospective review of a sample of completed cases to

determine the error rate.

Make sure the baseline reflects the same mix of cases that will occur in the pilot. If the baseline is measured in a quiet period and the pilot runs at a peak, the comparison is distorted.

The baseline has a second benefit. It shows how good the current process actually is. Sometimes it turns out that people, too, disagree with one another on the task more often than assumed. That is an important insight for the question of what quality can realistically be expected from the solution.

Formulating success criteria

Success criteria are set before the start. They should be testable, much like acceptance criteria in software development, and phrased in the language of the business. A good criterion describes what is measured, how it is measured, what value should be reached and where the data comes from.

Output quality

How often is the output correct, complete and usable? Measure quality against the current process, not against an abstract ideal. Distinguish between types of error. A missed mandatory case usually weighs more heavily than an unnecessary flag. Define how each type of error is weighted.

Review effort

How much time does human review of the outputs take? This criterion is often underestimated. A solution whose outputs must all be checked saves less time than expected. So measure the total effort per case, including review and correction.

Acceptance

Do people work with the solution, and do they trust it to an appropriate degree? Acceptance can be captured through short, repeated surveys, but also through observation: how often is the output adopted, how often discarded, how often bypassed? The two together give a more reliable picture than a survey alone.

Operation and cost

Can the solution be operated in the intended environment, and is the expected ongoing effort reasonable in relation to the value? This criterion can only be partly measured in a pilot. At a minimum, record the consumption per case, the support effort and the open prerequisites for operation.

Stop criteria

Stop criteria are just as important as success criteria. They define the conditions under which the pilot ends early. Without them, a pilot keeps running even when it is clear early on that it cannot answer its question.

Typical stop conditions are:

- The data needed is not available, or not in sufficient quality.
- Errors of a critical type occur and are not reliably detected.
- Privacy, information security or legal requirements prohibit the intended processing.
- The case volume is not sufficient to test the criteria.
- The people taking part are no longer available.

A stop is not a defeat. It answers the decision question earlier than planned and saves effort. For it to actually happen, it must be clear in advance who establishes it and who decides on it.

Sample catalogue

The following table shows how a criteria catalogue can be structured. The thresholds are deliberately left open. They depend on the task, the baseline and your organisation’s appetite for risk, and they are set jointly before the start.

Criterion	Measurement	Threshold	Source
Detection rate for critical cases	Share of cases with missing mandatory information that the solution detects, tested against a reference set	define, at least at baseline level	Reference set, business review
Unnecessary flags	Share of flags that turn out to be unfounded on review	define	Review log
Total effort per case	Working time including review and correction, compared with the baseline	define	Time recording, system data
Lead time	Time from receipt to the request for missing information	define	Timestamps in the case management system
Adoption of outputs	Share of outputs adopted without change or with minor changes	define	Solution log
Acceptance	Assessment by the people taking part in repeated short surveys	define	Survey

Criterion	Measurement	Threshold	Source
Consumption per case	Cost of model calls and infrastructure per case	define	Billing, monitoring
Stop: critical errors	Critical errors not detected during review	define, any occurrence triggers a review of whether to stop	Sample, feedback from downstream units
Stop: data availability	Share of cases in which required data is missing	define	Solution log

Agreeing and fixing the criteria

The criteria are agreed with the process owner, the decision-maker and representatives of IT, information security and privacy. They are then fixed, with a date and a version.

Changes during the pilot are possible, but not in passing. Anyone who wants to change a threshold gives a reason, documents it and obtains the decision-maker's approval. That way, it remains traceable what the pilot is measured against at the end.

At a glance

- Formulate a decision question that names the task, the benchmark for comparison and the conditions of operation.
- Record the current process in the same way you will later measure the solution.
- Criteria describe measurement, threshold and source. Set the thresholds before the start.
- Stop criteria are just as binding as success criteria.
- Changes to criteria are justified, documented and deliberately approved.

CHAPTER 04

Data, law and security

From AI idea to pilot. A guide

A pilot is only as meaningful as the cases it works with. At the same time, it only holds up if it is clear whether this data may be used and how it is protected. Both questions belong at the start.

Anyone who raises questions about information security and privacy only at the end risks a pilot stalling shortly before it goes live. This chapter describes what should be clarified before the start and gives an overview of the risk logic of the EU AI Act. The legal assessment of the individual case remains with your legal experts and your data protection officer.

The data check

Before the start, check three things: whether the data is available, whether it has the necessary quality and whether you may use it for this purpose.

Availability

Where is the data stored, in which system and in what form? Who can approve access, and how long does that usually take? Can the data be retrieved automatically, or are manual exports needed? Manual intermediate steps are acceptable in a pilot, but they must be recorded as an open prerequisite for operation.

Quality

Is the data complete, current and consistent? What share of documents is poorly legible, has missing fields or contains contradictory information? Draw a sample of real cases and look at it together with the people who work with them today. They know the quirks of the data better than any documentation.

Rights of use

May the data be processed for this purpose? This applies to personal data as well as to business information classified as confidential, data from contractual partners and content to which third parties hold rights. Also check whether contracts with customers or suppliers restrict disclosure to external services.

Checkpoint	Guiding question	Result	Responsible
Source	Which systems does the data come from?	…	…
Access	Who approves access, and has it been requested for the pilot?	…	…
Form	Is the data structured, in documents or in free text?	…	…

Checkpoint	Guiding question	Result	Responsible
Completeness	How often is information missing that the task requires?	…	…
Representativeness	Does the sample reflect the actual mix of cases?	…	…
Personal data	Does the data contain personal or particularly sensitive information?	…	…
Confidentiality	How is the data classified under the internal classification scheme?	…	…
Contractual restrictions	Are there contracts that restrict processing by third parties?	…	…
Retention	How long may inputs, outputs and logs be stored?	…	…

Privacy

As soon as personal data is processed, the GDPR applies. For a pilot, the following questions in particular must be clarified, together with your privacy organisation:

- **Purpose and legal basis:** On what basis is the data processed, and does the original purpose of collection cover the new processing?
- **Data minimisation:** Does the solution really need all the information, or can fields be omitted, shortened or pseudonymised?
- **Data processing agreements:** If an external service is used, the corresponding contracts are required. Also clarify whether the vendor stores inputs or uses them for its own purposes.
- **Processing location:** Where is the data processed and stored, and is it transferred to third countries?
- **Impact assessment:** If there is likely to be a high risk to the people concerned, a data protection impact assessment may be required.
- **Automated decisions:** If the solution makes decisions with legal or similarly significant effects on people, special requirements apply. Human review is then not only a question of quality.
- **Data subject rights and transparency:** How are the people concerned informed, and how are requests for access or erasure fulfilled, including for logs and intermediate outputs?

Sensitive data is often not needed for the preparation. Anonymised or purpose-built examples are enough to clarify the task, the process and the technical connection. Real data is only used once the processing has been approved.

Information security

An AI solution is a new system with new interfaces. It is subject to the same requirements as any other application, but it raises some questions of its own. At a minimum, clarify the following with your information security team:

- **Vendor and operating model:** Is a model used through an external service, run in your own cloud environment or operated on premises? What commitments does the vendor make on storage, access and security?
- **Access and permissions:** Which systems and data does the solution access? Does it receive only the rights it needs for the task? This applies especially when the solution calls tools or changes data.
- **Inputs from untrusted sources:** Emails, uploaded documents and web content can contain instructions that lead the model into unwanted behaviour. Define how such inputs are handled and which actions are excluded without approval.
- **Outputs:** Can the solution pass confidential content to people who are not allowed to see it? How do you prevent outputs from leaving the organisation unreviewed?
- **Logging:** What is logged, where and for how long? Logs are needed for traceability, but they often contain data that needs protection themselves.
- **Integration into existing processes:** How is the solution connected to identity management, vulnerability management and incident response?

In a pilot, individual requirements may be implemented in simplified form if the risk is limited and the simplification is documented. What matters is that the requirements for operation are known and feed into the evaluation.

The EU AI Act at a glance

The EU AI Act, Regulation (EU) 2024/1689 as amended, ties obligations to an organisation's role and the type and risk of its application. Providers, for example, develop AI systems and place them on the market; deployers use them under their own authority. The following overview provides initial orientation. Your legal experts assess the classification and applicable obligations in each case.

As of 2 October 2026: According to the European Commission's implementation timeline, prohibitions on certain AI practices have applied since 2 February 2025. Transparency rules generally apply from 2 August 2026, with transitional provisions for certain existing offerings.

High-risk rules apply from 2 December 2027 to systems covered by Annex III and from 2 August 2028 to systems embedded in products covered by Annex I. This selection of key dates does not replace an assessment of the relevant transitional provisions and exceptions. Source: European Commission, AI Act Service Desk, “Timeline for the Implementation of the EU AI Act”, accessed on 2 October 2026.

Risk class	Characteristics	Consequence at a glance
Prohibited practices	Applications that the legislator classifies as unacceptable, such as certain forms of manipulation or the scoring of people based on social behaviour	Use is not permitted
High risk	Applications in defined areas, such as certain uses in human resources, access to essential services or as a safety component of products	Extensive requirements, including for risk management, data quality, documentation, logging and human oversight
Transparency obligations	Applications that interact with people or generate content	Disclosure that AI is involved, or labelling of generated content
Minimal risk	Most other applications	No specific obligations from the risk logic; general requirements continue to apply

Three practical steps follow from this for the pilot. First, add the planned application to an inventory of AI applications, if one exists, or create the first entry. Second, make an initial classification by role and risk class and have your legal experts confirm it. Third, take the resulting requirements into account in the design of the pilot. Requirements for logging and human oversight are hard to add later.

Whatever the legal classification, the people involved need to understand the system's limitations, sources of error and human oversight. Train them on the specific task before the pilot begins. Your legal experts should check current law to determine which training and competence requirements apply to your role and application.

Involve them early

Privacy, information security and legal are not asked for approval at the end. They are asked to contribute at the start. There is a practical reason for this. Their requirements determine the architecture, the choice of vendor and the control points. If they are raised late, these decisions have to be made again.

A short joint meeting before the start has proven useful. The task, the data flow and the intended vendors are presented, open questions are collected and responsibilities for resolving them are agreed. The result is a list of approvals that must be in place before real data is used, and a list of requirements that only need to be met for operation.

> Information security and privacy belong at the start. Questions raised late hold a pilot up exactly when it is due to go live.

At a glance

- Check the availability, quality and rights of use of the data before you finalise the criteria.
- Clarify purpose, legal basis, data processing agreements and processing location with your privacy organisation.
- Treat inputs from untrusted sources and the solution’s permissions as security questions in their own right.
- Classify the application early according to the risk classes of the EU AI Act. The legal assessment is made by your legal experts.
- Work with anonymised or purpose-built examples during the preparation.

CHAPTER 05

People and accountability

From AI idea to pilot. A guide

Human oversight is part of the process, not a safeguard added afterwards. It is designed together with the solution and shapes risk and value alike. Too much oversight eats up the time saved. Too little lets errors pass unnoticed.

This chapter describes three basic patterns of human oversight, how to choose the right one and how to assign accountability clearly in the pilot.

Three patterns of oversight

A person reviews every output

The solution produces a proposal, and a person reviews it and approves, changes or discards it. No output is used further without approval. This pattern is suitable for the start of a pilot, for tasks where errors have serious consequences and for all steps that have an external effect. The value then lies in faster preparation, not in dropping the review.

A person reviews a sample

The outputs are used further, and a defined share is reviewed afterwards. This pattern is suitable when errors have limited consequences, can be corrected later and quality has been stable over a longer period. The sample must be drawn at random and be large enough to detect changes.

A person reviews exceptions

The solution handles clear-cut cases on its own and routes uncertain, unusual or irregular cases to a person. This pattern promises the greatest value, but it requires the most reliable detection of exceptions. It assumes that the solution indicates its uncertainty reliably or that clear rules define when a case is routed on.

Pattern	Suitable when	Value comes from	What to watch for
Every output	Serious consequences of errors, external effect, early phase	Faster preparation, more consistent outputs	Review effort can eat up the time saved. Habituation leads to superficial review.
Sample	Consequences of errors limited and correctable, quality proven stable	No more review of each case	The sample must be random and sufficient. Errors have an effect until they are discovered.
Exceptions	Clear-cut cases easy to separate, uncertainty reliably detectable	People handle only the difficult cases	Misclassified exceptions go unnoticed. Routing rules need their own review.

Choosing the right pattern

Three questions are decisive: How serious are the consequences of an error? Can it be reversed? How quickly is it noticed? Reading steps such as classifying or summarising can usually be designed more freely than steps that change data, trigger payments or communicate externally.

As a rule, start the pilot with the stricter pattern. That creates the data needed to decide later whether a lighter pattern is justifiable. Oversight can be relaxed step by step when evaluations show stable quality. Define the conditions for this in advance.

A note on effectiveness: anyone who reviews mostly correct proposals over a longer period reviews less carefully over time. This habituation is human, and appeals do not fix it. What helps are clear review points instead of a general approval, an indication of the places where the solution is uncertain and occasional known test cases that make it possible to check how attentive the review itself is.

Defining accountability

For every output, it must be clear who is accountable for it. The solution carries no accountability. Accountability remains with the person or unit that approves the output or whose process uses it. Clarify three questions before the start:

- Which outputs may be used without review, and which may not?
- Who is accountable if a faulty output goes unnoticed?
- How do corrections feed back so that the solution can be improved?

The answers belong in the pilot profile. They also matter because they continue to apply in operation. Accountability that remains unclear in the pilot is rarely settled cleanly in operation after the fact.

Roles in the pilot

The following overview assigns the most important tasks to roles. It follows the familiar RACI logic: R stands for carrying out the task, A for final accountability, C for being consulted before a decision, I for being informed afterwards. Each task has exactly one role with A.

Task	Decision-maker	Process owner	Users	Project lead	IT and architec-ture	Information security	Privacy
Define decision question	A	R	C	R	C	I	I

Task	Decision-maker	Process owner	Users	Project lead	IT and architec-ture	Information security	Privacy
Define criteria and thresholds	A	R	C	R	C	C	C
Approve data and access	I	A	I	R	R	C	C
Define oversight pattern	C	A	C	R	C	C	C
Review and cor-rect outputs	I	A	R	I	I	I	I
Document observations	I	C	R	A	R	I	I
Establish a stop	A	R	C	R	C	C	C
Prepare evaluation	C	C	C	A	R	C	C
Make decision	A	R	I	C	C	C	C

Depending on the organisation, roles can be combined. What matters is that each role is filled by a person who is available for the duration of the pilot. A small, stable group works more reliably than a large, changing one.

Feeding corrections back

Every human correction is information about the solution. It shows where the solution is wrong, which cases are difficult and where instructions, data or rules need to be adjusted. This information is lost if corrections only end up in the output.

So define how corrections are recorded: with the original output, the corrected version and a short category for the error. If this is built into the interface, it takes a few seconds per case, and it provides the most important basis for evaluation and improvement.

> Human oversight is designed together with the process. It shapes risk and value alike.

At a glance

- Choose between reviewing every output, reviewing a sample and reviewing exceptions based on the consequences of errors, their reversibility and how easily they are detected.
- Start with the stricter pattern and define in advance under which conditions oversight will be relaxed.
- Accountability for outputs always lies with people. Record with whom.
- Record corrections in a structured way. They are the most important basis for evaluation and improvement.

CHAPTER 06

Running the pilot

From AI idea to pilot. A guide

The pilot runs for a fixed period, with real cases and with the people who do the task today. The preparation from the previous chapters determines what is tested. This chapter describes how the pilot run is organised so that reliable results are available at the end.

The team

The core team consists of the process owner, two to a few users, a project lead and the technical implementation team. Information security, privacy and the decision-maker are not involved continuously, but they are brought in at fixed points.

Agree with the users' managers how much time the users will spend on the pilot. A pilot that is supposed to run alongside full day-to-day work gets the little time that is left over and delivers correspondingly thin results.

Time frame and phases

The period is set before the start. It should be long enough for sufficient real cases to pass through to test the criteria, and short enough to keep attention and participation up. How many cases are enough depends on the criteria. Anyone who wants to detect rare, critical errors needs more cases than someone comparing an average processing time. An extension is possible, but it requires a deliberate decision and a reason.

A structure of four phases has proven useful:

Phase	Goal	Result	Time frame
Preparation	Clarify task, criteria, data, approvals and reference set, build the solution	Pilot profile, criteria catalogue, approvals, solution ready for use	A few weeks, depending on approvals and integration
Parallel run	The solution runs alongside the current process, outputs are compared but not used	Comparison with the current process without risk to operations	A few days to a few weeks
Supervised live run	Outputs are used in the process under the agreed oversight pattern	Measurements of quality, review effort and acceptance under real conditions	A few weeks
Evaluation	Compare results with the criteria, prepare the decision paper	Decision paper and decision meeting	A few weeks

Whether a parallel run is necessary depends on the risk. For tasks where errors have serious consequences, the entire pilot can run in parallel. For tasks with minor consequences, the supervised live run can start early, as long as every output is reviewed.

Real and difficult test cases

Use real, representative cases, including the difficult and incomplete ones. Carefully selected examples make a solution look better than it will be in everyday use.

Put together a reference set before the start. This is a fixed collection of cases with a known correct output, determined by subject matter experts. It is used to test the solution consistently before and during the pilot, for example after every change to instructions or the model. The reference set should contain:

- **ordinary cases** in the mix in which they occur in everyday work,
- **borderline cases** that even experienced people have to think about,
- **incomplete and faulty inputs**, such as poorly legible documents or missing fields,
- **out-of-scope cases**, where the solution should recognise that it is not responsible,
- **critical cases**, where an error would have serious consequences,
- **inputs that could lead the solution into unwanted behaviour**, if the task processes content from untrusted sources.

The people who do the task today know best which cases are difficult. Ask them specifically to contribute such cases. The reference set is not changed quietly during the pilot. New cases go into a supplement so that comparisons over time remain possible.

Feedback and checkpoints

Agree fixed, short checkpoints at regular intervals. Three questions are examined there: Are the criteria still within reach? Has a stop condition been met? Are there observations that suggest an adjustment?

In addition to the meetings, there needs to be a simple way to give feedback as you go. Users should be able to record a notable observation in a few seconds, directly in the solution or in a shared log. Feedback that is only collected from memory at the checkpoint is incomplete.

If the solution is improved during the pilot, for example through changed instructions, test every change against the reference set first. That makes visible whether an improvement in one place leads to deterioration in another.

Documentation

Document as you go, not at the end. Three logs are usually enough:

- **Output log:** input, output of the solution, review decision, correction and error category for each case. It is kept automatically wherever possible.
- **Observation log:** notable, unexpected or recurring observations, including those that fit none of the criteria.
- **Change log:** every change to the solution, instructions, model, data, oversight pattern, scope or criteria, with date, reason and approval.

These logs are the basis of the evaluation. They are also a rehearsal for operation, where traceability is needed permanently.

Changes to scope

Changes to scope are allowed, but not in passing. If the task is extended or changed during the pilot, those responsible decide this deliberately and adjust the criteria. Otherwise the pilot ends up measuring something other than what was planned.

An example: the users suggest including notifications from another line of business alongside property damage, because the solution could help there too. The suggestion is reasonable. It is recorded in the observation log and noted for the decision. It is only included if the decision-maker agrees and the criteria, reference set and time frame are adjusted.

Communication

A pilot is watched, including by people who are not involved. Expectations form quickly, in both directions. So communicate in advance and factually: which task is being tested, why, for how long, who is involved and that it ends with a decision, which may also be a stop.

For the users taking part, one question is particularly important: what does the pilot mean for their work? Answer it openly. People who fear making themselves redundant by taking part do not give honest feedback. If employee representative bodies such as the works council are affected, involve them early.

At a glance

- Set the time frame, the phases and the users’ time before the start.
- Test with a fixed reference set that contains ordinary, difficult, incomplete and critical cases.
- Hold fixed checkpoints and provide a simple way to give feedback as you go.

- Keep the output, observation and change logs as you go.
- Change the scope only deliberately, and then adjust the criteria.

CHAPTER 07

Evaluating and deciding

From AI idea to pilot. A guide

The pilot ends with a decision meeting, not a presentation. It is clear in advance who decides and on what basis. The results are set against the agreed criteria, including the observations that do not fit any of them.

This chapter describes how to prepare the evaluation, which three outcomes are possible and how to run the decision meeting so that it actually leads to a resolution.

Evaluating against the criteria

Present each criterion from the catalogue with the measured value, the agreed threshold and a short assessment. Use three levels for this: met, partly met, not met. A “partly” assessment needs a reason, for example that the threshold was reached only for some of the cases.

Pay attention to four points:

- **Comparability:** measured values of the solution are compared with the baseline, which was recorded in the same way. Differences in the mix of cases or the period are named.
- **Breakdown:** averages hide problems. Break down the results by type of case, especially critical cases and borderline cases.
- **Development over time:** did quality change over the course of the pilot, for example after changes to the solution? Did acceptance rise or fall?
- **Uncertainty:** with few cases, measured values are uncertain. Say so instead of feigning precision.

Observations outside the criteria

Not everything important fits into a criteria catalogue. Perhaps it turned out that an upstream step is the actual cause of rework. Perhaps users used the solution for a subtask that was not planned. Perhaps the effort for an approval in operation is much higher than expected.

These observations belong in the evaluation, in a section of their own. They do not replace the criteria and do not justify reinterpreting them after the fact. They can, however, influence the decision, for example whether reworking makes more sense than scaling.

The three possible outcomes

Three outcomes are possible, and each is equally valid if it is well founded.

Scale

The criteria are met, and the prerequisites for operation can be put in place. Scaling does not mean using the solution everywhere at once. It means commissioning the move into operation, with a budget, ownership and a rollout plan. Open prerequisites for operation are named as conditions.

Rework

The approach is sound, but the task, the data or the checkpoints need adjusting. This leads to a new, tightly scoped pilot or an extension with revised criteria. Reworking is only a meaningful outcome if it names specifically what changes and which question is to be answered afterward. Reworking without a clear change is a disguised continuation.

Deliberately stop

Value, risk and effort do not fit together. The reasons and lessons are recorded so that they inform future assessments. This includes the conditions under which a new review would make sense, such as better data quality or new technical possibilities.

Deliberately stopping, in particular, needs the backing of leadership. Without it, pilots tend to be extended rather than ended.

> A pilot ended with a clear rationale is worth more than one that carries on without a decision.

The decision meeting

The decision meeting is a working meeting with a small group. It should include the decision-maker, the process owner, the project lead and, depending on the open points, representatives of IT, information security and privacy.

The decision paper is circulated a few days in advance. The meeting itself is not for presenting the results but for clarifying open questions and reaching a resolution. It ends with the chosen outcome, the rationale, the next steps with owners and, where applicable, the conditions attached to the outcome.

The decision paper follows a fixed structure:

Section	Content	Guiding question
Decision question	The question defined before the start	What was the pilot meant to answer?

Section	Content	Guiding question
Summary	Recommendation in a few sentences	Which outcome is recommended, and why?
Criteria	Measured value, threshold and assessment for each criterion	Were the criteria met?
Observations	Important insights outside the criteria	What else came to light?
Risks	Open risks relating to quality, security, privacy and law	What could go wrong in operation?
Prerequisites for operation	Open points for operation, main types of cost	What is needed to operate the solution?
Options	Scale, rework, deliberately stop, each with its consequences	What does each option mean?
Resolution	Chosen outcome, conditions, next steps	What is decided, and who does what?

A version to fill in can be found in the appendix.

After the decision

Record the resolution in writing and share it with everyone involved, including the users. They contributed time and experience and should know what follows from it.

Also secure the groundwork: reference set, logs, criteria catalogue and evaluation. If you scale, they form the starting point for operation. If you stop, they are the basis for reviewing the question again later with less effort.

At a glance

- Present each criterion with measured value, threshold and assessment, and break the results down by type of case.
- Observations outside the criteria belong in the evaluation, but they do not justify reinterpreting the criteria after the fact.
- Scaling, reworking and deliberately stopping are equally valid if they are well founded.
- The decision meeting is for reaching a resolution, not for a presentation.

CHAPTER 08

From pilot to operation

From AI idea to pilot. A guide

The move into operation does not start after the pilot but in its planning. In the pilot, many things are done by hand that must work reliably and permanently in operation: retrieving data, checking quality, approving changes, supporting users.

This chapter describes the prerequisites for operation, the technical control layer around the solution, the allocation of ownership, the types of running cost and a step-by-step rollout.

Prerequisites for operation

If you intend to scale, clarify the following points during the pilot:

- business and technical ownership for ongoing operation,
- integration with existing systems instead of manual workarounds,
- regular checks that output quality remains stable,
- rules on who approves changes to the solution,
- training and support for the people who work with the solution,
- a realistic view of the running costs.

In addition, there are the requirements from information security, privacy and, where applicable, the EU AI Act that were simplified or deferred in the pilot. The list from the joint meeting at the start is now worked through.

The AI harness

By AI harness, we mean the technical control layer around models and agents. It connects organisational policies with controls and evidence during operation. Its effectiveness depends on the implementation and the use cases tested. Here, the harness comprises five building blocks.

Guardrails

Guardrails check inputs, outputs and planned actions. Depending on their purpose, they combine fixed rules, permission checks and model-based assessments. They help limit unauthorised data sharing or actions. Access rights and approval requirements are enforced outside the language model. Test effectiveness against normal and deliberately manipulated cases, and document remaining limitations.

Evaluations

Evaluations check the quality of the solution systematically against a fixed set of cases. The reference set from the pilot is their starting point. Before every change to the model, instructions or data, an evaluation is run, and the change is only approved if quality does not fall below

the agreed threshold. In operation, the reference set is extended with new, typical and difficult cases.

Logging

Logging records which input led to which output, with which version of the model and instructions and with which human review decision. It makes it possible to trace individual cases and to provide evidence. The scope and retention period are agreed with privacy and information security.

Monitoring

Monitoring observes the solution continuously: quality in samples, share of corrections, share of escalated exceptions, response times, errors, consumption. It shows when quality changes gradually, for example because the inputs change or a vendor updates its model. For each metric, it is defined at which value who is notified.

Change process

Models, instructions and configurations are versioned. Every change follows a fixed path: proposal, evaluation, approval, introduction, observation. It is also defined how to switch back to an earlier version and how to pause the solution and continue the process by hand.

Building block	Purpose	Guiding question for operation
Guardrails	Control inputs, outputs and actions	Which operations need to be checked, blocked or approved?
Evaluations	Check quality before changes	How do we recognise that a change makes quality worse?
Logging	Trace cases and provide evidence	Can we explain how a specific output came about?
Monitoring	Detect changes during ongoing operation	Do we notice a loss of quality before it causes harm?
Change process	Introduce and roll back changes in a controlled way	Who approves changes, and how do we return to the last good version?

The harness does not have to be complete before first use. Its scope depends on the risk of the application. Logging, a fallback and an evaluation before changes, however, belong to every solution in operation.

> The prerequisites for operation belong in the planning of the pilot, not at its end.

Ownership in operation

In operation, the solution needs two fixed owners. Business ownership lies with the business unit, usually with the process owner from the pilot. This person decides on scope, oversight pattern and quality thresholds. Technical ownership lies with IT or a product team. It operates the solution, the harness and the integrations.

Also define who approves changes, who responds to anomalies in monitoring, who collects user feedback and who maintains the solution in the inventory of AI applications. A solution without clear operational ownership loses quality over time without anyone noticing.

Running costs

The running costs of an AI solution are made up differently from those of conventional software. One part depends directly on the case volume, another on the effort needed to maintain quality. The following overview lists the usual types of cost. Determine their amount from the measured values of the pilot and your vendors' offers.

Type of cost	Driver	Note
Model usage	Case volume, length of inputs and outputs, choice of model	Extrapolate from the consumption per case in the pilot, allow for fluctuations
Infrastructure	Hosting, storage, logs, network	The retention period of the logs affects storage needs
Integration and maintenance	Number and stability of the interfaces	Changes to connected systems create follow-up effort
Harness and monitoring	Scope of guardrails, evaluations and metrics	Align with the risk of the application
Human review	Oversight pattern, share of exceptions, correction effort	Often the largest running item, and measurable in the pilot
Quality maintenance	Extending the reference set, adjusting instructions, changing models	Effort arises even without new features

Type of cost	Driver	Note
Training and support	Number of users, staff turnover, changes to the process	New employees need an introduction
Governance and evidence	Inventory, reviews, documentation	Scope varies with the risk class

Rollout

Introduce the solution step by step. Start with the team from the pilot, extend to further teams or locations and check after each step whether the metrics remain stable. Every extension of scope, for example to new types of case or languages, is treated like a change: with evaluation and approval.

For every step, the fallback is known and tested. The people who work with the solution are trained, know their review task and know how to report anomalies. The users from the pilot are valuable points of contact for new teams.

Also schedule a fixed date on which the solution is assessed again after an initial phase of operation. It checks whether value, cost and risk match the expectations from the decision, and whether the oversight pattern can be adjusted.

At a glance

- Clarify the prerequisites for operation during the pilot, not afterward.
- The AI harness connects controls, checks and evidence during operation. Its effectiveness and limitations need to be tested.
- The solution needs business ownership and technical ownership.
- Running costs also arise from human review and quality maintenance, not only from model usage.
- Introduce the solution step by step, with a tested fallback.

Appendix

From AI idea to pilot. A guide

The appendix summarises the steps of this guide in checklists and templates. The templates can be used as they are or transferred into existing project tools.

Checklists

Choosing the task

- [ ] The task is described so that outsiders can understand it without asking.
- [ ] Inputs, output and the people who work with it next are named.
- [ ] For each step, it has been checked whether a model, a rule or a person is best suited to it.
- [ ] The task has been scored in the assessment matrix, and the weighting was set in advance.
- [ ] The scope is limited, and what is out of scope is recorded in writing.
- [ ] A process owner and a participating team are named.

Defining the question and criteria

- [ ] The decision question is recorded in writing and agreed with the decision-maker.
- [ ] The baseline of the current process has been recorded in the same way as the later measurement.
- [ ] Success criteria for quality, review effort, acceptance and operation are described with measurement, threshold and source.
- [ ] Stop criteria are defined, as is who establishes and who decides on a stop.
- [ ] The criteria catalogue is fixed with a date and a version.

Clarifying data, law and security

- [ ] Availability, quality and usage rights of the data have been checked on a sample.
- [ ] Purpose, legal basis, data processing agreement and processing location have been clarified with privacy.
- [ ] Vendor, access rights, handling of untrusted inputs and logging have been agreed with information security.
- [ ] The application is recorded in the inventory and provisionally classified against the risk classes of the EU AI Act, and the legal assessment has been commissioned.
- [ ] Approvals are in place before real data is used.
- [ ] Requirements that only need to be met for operation are listed.

Defining the role of people and ownership

- [ ] The oversight pattern has been chosen, and the choice is justified.

- [ ] The conditions under which control may be relaxed are defined.
- [ ] For each output, it is clear who is accountable for it.
- [ ] The roles are filled, and for each task there is exactly one role with final accountability.
- [ ] Corrections are recorded in a structured way with an error category.

Running the pilot

- [ ] Time frame and phases are defined, and the users' time is agreed.
- [ ] The reference set contains ordinary, difficult, incomplete, off-topic and critical cases.
- [ ] Checkpoints are scheduled.
- [ ] The output log, observation log and change log are kept up to date.
- [ ] Every change to the solution is tested against the reference set in advance.
- [ ] Those involved and the affected units are informed, and the works council is involved where required.

Evaluating and deciding

- [ ] Each criterion is presented with measured value, threshold and assessment.
- [ ] The results are broken down by type of case.
- [ ] Observations outside the criteria are recorded in a section of their own.
- [ ] The decision paper was circulated before the meeting.
- [ ] The resolution with rationale, conditions and next steps is recorded in writing and shared with everyone involved.

Moving into operation

- [ ] Business and technical ownership for operation are named.
- [ ] Manual workarounds have been replaced by integrations with existing systems.
- [ ] Guardrails, evaluations, logging, monitoring and change process are implemented in line with the risk.
- [ ] The fallback is described and tested.
- [ ] The types of running cost are recorded and included in the budget.
- [ ] Users are trained, and a date for the reassessment is set.

Templates

Pilot profile

Field	Content
Name of the pilot	
Task	
Decision question	
Scope	
Out of scope	
Decision-maker	
Process owner	
Project lead	
Users	
IT and architecture	
Information security	
Privacy	
Data sources and approvals	
Oversight pattern	
Accountability for outputs	
Time frame and phases	
Checkpoints	
Decision meeting	
Open prerequisites for operation	

Criteria catalogue

No.	Criterion	Type	Measurement	Baseline	Threshold	Source	Owner
1		Success					
2		Success					
3		Success					
4		Success					
5		Success					
6		Stop					
7		Stop					
8		Stop					

Version: ... Fixed on: ... Approved by: ...

Decision paper

Section	Content
Pilot	
Decision question	
Recommendation in a few sentences	
Criteria: met	
Criteria: partly met, with reason	
Criteria: not met	
Observations outside the criteria	
Open risks	
Open prerequisites for operation	
Types of running cost	
Consequences of scaling	

Section	Content
Consequences of reworking	
Consequences of deliberately stopping	
Resolution	
Conditions	
Next steps and owners	
Date and signature of the decision-maker	

Glossary

Term	Meaning
Agent	AI system that breaks a task down into substeps, calls tools and systems to carry them out and adjusts its approach based on intermediate results.
AI harness	Technical control layer around models and agents, with guardrails, evaluations, logging, monitoring, approvals and versioning.
Baseline	Record of the current process against which the results of the pilot are measured.
Oversight pattern	Defined form of human review: every output, a sample or exceptions.
Decision question	The one question the pilot is meant to answer for the decision-maker.
EU AI Act	Regulation (EU) 2024/1689 on artificial intelligence, as amended. Obligations and application dates depend on the role, application and transitional provisions. Chapter 4 provides a dated overview.
Evaluation	Systematic check of the quality of a solution against a fixed set of cases, especially before changes.
Guardrail	Technical control for inputs, outputs or actions, using rules, permission checks or model-based assessments. Effectiveness and limitations need to be tested.
Monitoring	Continuous observation of the quality, behaviour and consumption of a solution in operation.
Parallel run	Phase in which the solution runs alongside the current process and its outputs are compared but not used.

Term	Meaning
Pilot	Time-limited trial of a solution in the actual process with real cases and criteria set in advance, leading to a decision.
Proof of concept	Test of whether something is technically possible.
Reference set	Fixed collection of cases with a known correct result, against which the solution is tested consistently.
Stop criterion	Condition set in advance under which a pilot ends early.
Success criterion	Testable requirement with measurement, threshold and source, set before the start.



Let’s talk about your plans.

Optimentum supports companies in strategy, technology and transformation. Tell us which decision is on your agenda.

office@optimentum.eu
optimentum.eu

Optimentum GmbH · Tegetthoffstraße 3/10, 1010 Vienna, Austria · FN 686237y, Commercial Court of Vienna