CLEVERDIST
Industrial AI Solutions

IO Mission Identification Guide

Industrial Operator

How to identify high-value opportunities for industrial AI

Document: Methodology guide
Version: 1.0
Date: February 2026
Use: Internal and customers

Contents

1 Introduction **2**

2 What Is an IO Mission? **3**

3 How to Identify a Good Mission **4**
3.1 Step 1: Identify the pain point 4
3.2 Step 2: The intelligence test 4
3.3 Step 3: Levers and data 5
3.4 Step 4: Pilot readiness 5

4 Disqualification Criteria **7**

5 The Mission Card **8**

6 Next Steps **9**

1 Introduction

This guide has a practical objective: to help technical and operations teams identify specific use cases where IO's artificial intelligence can deliver measurable and sustainable value to their operations.

Industrial Operator (IO) is an AI platform designed for industrial and enterprise environments. IO does not replace existing systems: it sits on top of them, connecting to the data sources and systems already in place. Its function is to model the reasoning of process experts and apply it continuously, 24/7, with the ability to analyze more information and in less time than a human team could.

An IO use case is called a **mission**. A mission is a specific, recurring problem where AI delivers continuous value. Not every problem is a good mission. This guide provides a structured method to distinguish good opportunities from those that are not.

Fundamental principle: An IO mission must have a real, tangible pain point that justifies the operational cost of an intelligent system. If the benefit does not clearly outweigh the investment, the mission will not sustain itself over time.

2 What Is an IO Mission?

An IO mission is a specific operational problem that meets these characteristics:

- It has **sustained impact**: it occurs frequently or continuously, or it is an infrequent event with high impact whose permanent monitoring consumes disproportionate resources. What matters is that the combination of frequency and impact justifies continuous intelligent supervision.
- It requires **reasoning**: it cannot be solved with a fixed rule or a deterministic algorithm. It needs the ability to interpret context, weigh variables and make decisions.
- It has an **associated cost**: it consumes time, resources, affects quality, performance or safety.
- There is an **expert** who knows how to address it, but whose reasoning cannot be practically materialized on a continuous basis. It is not just that the expert cannot be present 24/7: even if present, the volume of information, the number of simultaneous variables, or the speed required make it infeasible for a human to provide that level of analysis in a sustained manner. Until the arrival of generative AI, this kind of intelligence was simply not automatable.

IO captures that expert reasoning and applies it continuously. It does not simply execute instructions: it interprets situations, analyzes real-time and historical data, processes text, images and voice, and generates well-founded recommendations or actions.

Combination of techniques

IO can combine different techniques as the mission requires:

- **Generative AI**: to interpret context, reason about complex situations and generate explanations.
- **Statistical optimization**: to explore scenarios and find optimal configurations.
- **Time series analysis**: to detect trends, anomalies and predict behaviors.
- **Unstructured information processing**: to read, interpret and extract information from documents, images and voice.

Operating modes

Each mission can operate in the mode that best fits the needs and level of confidence:

- **Supervised** (human-in-the-loop): AI analyzes, recommends and explains. The human decides and confirms.
- **Autonomous** (within policies): AI acts within explicit rules defined by the organization. Responsibility remains human, but execution is delegated.

Integration with existing systems

IO connects to systems already in place through standard interfaces. It does not require replacing or modifying current systems. Existing data and logic remain intact.

3 How to Identify a Good Mission

This is the core of the guide. For each opportunity under consideration, follow these four evaluation steps. Each step acts as a filter: if it is not passed, the opportunity is not a good mission, at least not at this time.

3.1 Step 1: Identify the pain point

The first requirement is that a real, describable problem exists.

Key questions:

1. Can the problem be described in concrete terms? "We want to improve" is not enough. It is necessary to be able to say *what* needs improving and *why* it is not being achieved today.
2. Can the cost of the problem be **quantified**? Even as an approximation, it must be expressed with concrete figures, not in the abstract. "We lose some time" is not sufficient. "We dedicate 40 hours/week to this task" or "we have 8% of product out of specification" is. Examples of metrics:
   - **Time**: personnel hours dedicated (e.g., X hours/week, Y full-time equivalents).
   - **Resources**: qualified personnel assigned to tasks below their capability.
   - **Quality**: percentage of product out of specification, number of rework cycles, complaints per period.
   - **Performance**: percentage of production below nominal capacity, number of stops.
   - **Energy**: excess consumption quantified (kWh, percentage above optimal).
   - **Safety**: number of incidents, frequency of risk situations.

   Optionally, an economic estimate of the cost can be included. It is not mandatory to share it, but having one helps evaluate the return on investment.
3. Does the potential benefit justify an investment in an intelligent system with recurring operational cost?

If the problem cannot be described or its cost cannot be quantified: the opportunity is not sufficiently defined. It needs to mature before moving forward. The cost does not have to be exact, but it must have a number attached to it.

3.2 Step 2: The intelligence test

Not every problem needs AI. This step determines whether the problem requires reasoning capabilities or whether it can be solved with simpler methods.

Key questions:

1. Can it be solved with a fixed rule or a deterministic algorithm?
   - *If yes*: it is not an IO mission. It is a conventional software development or an automation that is implemented once and maintained. It does not require a continuous AI service.
2. Does it require interpreting context, weighing multiple variables, or applying expert judgment?
   - *If yes*: it is a good candidate.
3. Is there an expert who would know how to act, but cannot be present 24/7?
   - *If yes*: it is a strong candidate. IO models exactly that reasoning.

4. Does it involve reasoning about unstructured information (documents, images, free-text records)?
    - *If yes*: it is also a valid candidate.

The ideal scenario: an expert knows how to solve the problem, but solving it requires analyzing more information than a human can process in real time, or maintaining continuous attention that is neither practical nor scalable. IO does not discover the solution. It captures the expert's reasoning and executes it at scale, uninterrupted.

3.3 Step 3: Levers and data

A well-identified problem also requires that there are means to act on it and data to feed that action.

Levers are the actions or decisions that can improve the situation:

- Adjust a process setpoint (speed, temperature, pressure).
- Change a schedule or work sequence.
- Issue a recommendation to the operator.
- Classify, prioritize or route information.
- Generate an alert for a specific condition.

Data are the required information sources:

- Real-time process signals (sensors, PLCs, SCADA).
- Historical data (databases, process historians).
- Enterprise systems (ERP, CMMS, MES, MOM).
- Documents (invoices, reports, manuals, records).

Key questions:

1. Can the specific levers be identified? What would be adjusted, decided or recommended?
2. Do the necessary data exist and are they accessible, or is there a realistic path to obtain them?
3. Can the data sources be connected through standard interfaces or through reasonable development?

If there are no clear levers: the problem may be real but not actionable for IO. If no data is available and there is no realistic path to obtain it: the mission is not viable at this time.

3.4 Step 4: Pilot readiness

Not every good mission is a good first pilot. A pilot must demonstrate value quickly and with controlled risk.

Criteria for a good pilot:

1. **Bounded scope**: it can be limited to one process, one line, one area, or one specific type of problem. Do not try to solve everything at once.
2. **Results measurable in weeks**: value can be observed within 2 to 3 months, not years.

3. **Available sponsor**: there is an expert or stakeholder willing to invest time in knowledge capture sessions and validation.
4. **Data already collected**: the necessary information is already being gathered, even if it is not being used for this purpose.
5. **Low integration complexity**: it does not require connecting more than 2–3 systems or solving network access problems that take months.

Signs that a mission is better as a second project (not a pilot):

- It requires integrating many systems simultaneously.
- It involves regulatory or security approvals that extend timelines.
- The benefit is high but the time to measure it is long.
- Technical complexity is high and the risk of delays is significant.

Practical advice: when there are multiple candidates, do not choose the most ambitious one. Choose the one that best combines demonstrable value and feasibility within 3 months. Once value is proven, you can scale to more complex missions with confidence.

4 Disqualification Criteria

Identifying good missions is just as important as quickly discarding those that are not. If an opportunity meets any of the following criteria, it is not a viable IO mission in its current form.

	Disqualification criterion	**Reason**
1	Pure automation	If the problem can be solved with fixed rules, conditionals or a deterministic algorithm, it is conventional software development. It is implemented once and maintained. It does not need a continuous AI service and does not justify a recurring operational cost.
2	Benefit does not justify the cost	IO is a service with operational cost (infrastructure, AI, support). If the savings or improvements generated do not clearly exceed that cost, the mission will not sustain itself. It does not have to be an exact calculation, but it must be evident that there is a significant return.
3	No expert knowledge exists	If no one in the organization knows how to improve the situation, not even in theory, AI will not be able to either. IO models the reasoning of an expert. If there is no reasoning to model, there is no mission.
4	No data and no path to obtain it	Without input data (signals, historical records, documents), the system has nothing to work with. If they do not exist today and there is no realistic plan to have them, the mission is not viable at this time.
5	Too complex for a first pilot	If the mission requires many systems, complex integrations or approvals that take months, it is better to save it for a later phase. This is not a permanent disqualification, but a deferral.

Important: criterion 5 does not mean the mission is bad. It means it is not the right one to start with. The best complex missions are tackled after proving value with a more bounded pilot.

5 The Mission Card

For each opportunity that passes the four evaluation steps, the following information should be documented. This card enables a structured presentation of the use case and facilitates joint evaluation with Cleverdist.

Field	What to document
Mission name	A short, descriptive name that identifies the use case.
Problem description	What is happening today that should not be? What needs improving and why is it not being achieved currently?
Pain point / Cost	The impact of the problem, **quantified**: hours/week, percentage of product out of specification, number of stops, energy consumption, safety incidents, etc. An exact calculation is not necessary, but concrete figures that dimension the problem are required. Optionally, an economic estimate if desired.
The expert	Who in the organization knows how to address this problem? Why can they not do it continuously? (e.g., cannot be present 24/7, the volume of information exceeds their analysis capacity, too many simultaneous variables).
The levers	What can be adjusted, decided or recommended to improve the situation? Examples: process setpoints, schedules, maintenance priorities, document classification.
Signals / Data	What relevant data exists for this mission? Where does it reside? (sensors, databases, enterprise systems, documents). Is it currently being collected?
Expected benefit	What would success look like? How would it be measured? (time reduction, cost savings, performance improvement, quality improvement, risk reduction).
Estimated complexity	Subjective assessment: Low / Medium / High. Consider the number of systems to integrate, data availability, required approvals and domain complexity.

It is recommended to complete this card for **2 to 4 candidates**. The information does not need to be exhaustive at this stage: the goal is to have a sufficient basis for a productive conversation. Technical details are refined in joint sessions.

6 Next Steps

Once the team has identified and documented the mission candidates, the process continues as follows:

1. **Document 2 to 4 candidates** following the mission card described in the previous section.
2. **Joint evaluation session** with Cleverdist: candidates are presented, analyzed together, and those with the highest value and feasibility as a pilot are identified.
3. **Pilot selection**: the mission that best combines demonstrable value, technical feasibility and reasonable timeline is chosen.
4. **Pilot proposal**: Cleverdist prepares a detailed proposal with scope, timeline, integration, deliverables and investment.
5. **Execution**: the goal is to reach measurable results within 8 to 12 weeks from kickoff.

The pilot is not the end: it is the beginning. Once value has been demonstrated in a bounded case, it can be scaled to more missions, more areas and more complexity. IO's architecture is designed to grow with the needs of the organization.