Make discovery concrete.

Replace broad requirements with observed workflows, representative inputs and a clear acceptance boundary.

Discovery preparation

Bring	Why it matters
A recent request	Shows the actual sequence of work
Forms and spreadsheets	Reveals information and validation rules
A difficult exception	Exposes judgement, authority and recovery needs
The process owner	Can resolve priorities and acceptance questions

Follow one item all the way through

Observe how the request enters, where it waits and how completion is recorded. Do not stop at the team boundary if another system owns the result.

Write testable acceptance criteria

Describe an input, an action and an observable outcome. Include the failure cases that would make the change unsafe or unusable.

Begin with a task people already perform

Discovery should establish what the business is trying to change and why the current process is difficult. Ask someone to walk through a recent example using the actual sequence of documents, systems and decisions. Record where information is copied, where work waits and which exceptions require experienced staff to intervene.

Separate the symptom from the underlying constraint. A slow approval process might be caused by missing information rather than an old interface. A proposed AI assistant might depend on documents that have no agreed owner. These findings can change the proposed solution substantially. Discovery creates value when it prevents the team from building the wrong thing, not when it merely confirms the initial feature list.

Test the assumptions that affect scope

Identify the uncertainties that could change cost, feasibility or delivery order. These might include access to an external API, the quality of historical records or a requirement to work without reliable connectivity. Investigate the highest-impact uncertainty with a small, bounded exercise.

For an invoice workflow, inspect a representative sample before estimating extraction and validation work. Include unusual layouts, missing references and disputed records. For a portal, confirm that the source system can supply the fields users need within the required access model. Record what the exercise establishes and what remains unknown. A prototype should answer a question rather than become an unreviewed production system.

Produce a usable delivery starting point

The discovery outcome should include the problem statement, users, workflow boundaries, dependencies and a prioritised first increment. Define acceptance examples and identify decisions still needed from the business. Where an estimate is provided, state its assumptions and the conditions that could change it.

Include a recommendation on what to defer. Not every observed problem belongs in the first release. Explain how the proposed sequence reduces risk or provides feedback earlier. Agree the next decision, its owner and the information needed to make it. Discovery is complete when the team can begin a sensible piece of delivery with shared expectations, rather than when a workshop calendar has been exhausted.

Is discovery a commitment to build?

No. Its purpose is to make the next decision better informed. The result may support a smaller change, an existing product or a decision not to proceed.

Read the current resource online

https://cobnex-review.rgcsekaraa.workers.dev/resources/domains/