Show the concept *before it hardens.*

Plan three customer checks while the design can still change. Walk through the concept, test the controls and bring the revision back before release.

How I used it at Motorola Solutions.

At Motorola Solutions I led design for CommandCentral Notify from early concepts through release. We took early concepts to public-safety customers before committing to the interface, so we could ask about the workflow while there were still alternatives.

We followed customer demonstrations with remote usability testing and card sorting, then brought the work back to customers. The methods served different purposes. A walkthrough lets a customer explain how they would do the work. A usability session shows whether someone can find and use the controls. Card sorting shows where people expect related controls to live.

When to use it.

Good fit

The workflow is new, the users know the setting better than the team, and the structure can still change.

Start elsewhere

The interface is already built and only details can change. Run usability sessions on the controls instead of a concept walkthrough.

The steps.

Change the question as the concept matures

Concept	→	Controls	→	Revision
Walk through the workflow with customers.		Test tasks and use card sorting to check grouping.		Return to customers to check the revised flow.

1. Prepare first concepts. We used story mapping, desk research, personas and competitive analysis.
2. Walk the concepts through with customers while alternatives exist. Ask where they would define a scenario and what they need to see.
3. Run remote usability sessions to see whether people can find and use the controls.
4. Use card sorting to see where people expect related controls to live.
5. Revise, then return to customers before release.

Making the decision.

What to check.

Ask a different question at each point.

At concept, can an operator recognise the workflow and explain how it would fit their work? At controls, can they find the parts they need, and do the groups make sense? At revision, does the interface still hold up when customers see it again?

When to invite people in

Early concepts let the team change the structure. Once development is implementing the flow, changes have a different cost. Decide the date before building starts.

Worked example.

Practice scenario

A new alert workflow asks an operator to choose recipients, channels and a message. In a walkthrough, an operator says they would prepare the area before an incident, not while sending.

The decision

Let a saved scenario carry the area, so the operator reviews recipients, channels and message at the moment of use.

Then test whether people can find saved scenarios and whether the controls are grouped the way they expect.

When it gets complicated.

Customers disagree about the workflow.

Record where they differ by role. Look for the part everyone does the same way and design that first.

Usability results contradict the walkthrough.

Keep the two pieces of evidence separate. Check whether they describe the same role, task and conditions. Use a targeted follow-up to find out whether the gap is about the workflow or how the prototype presents it.

The team wants to wait until the design is polished.

Bring the smallest artefact that lets customers walk through the task. Tell them which structure is open and ask them to narrate a real situation.

See the research sequence in Notify →

Worksheet.

Plan the next round.

Name the point you are at, the method that fits it and the change you could still make if people struggle.

Filled-in example

Point	Concept.
Method	Customer walkthrough.
Question	Where would an operator define a scenario?
What to watch	What they set up before the event and what they check at the moment of use.
Change we could still make	Move the area into a saved scenario.
Return date	Before release.

Use it on your project

After each round, record what you learned and which part of the structure it changed.

Open decision and research method

What you will watch people do

What you could change and when you'll check it again