WOMEN IN ENERGY 2026 · BREAKOUT SESSION

Operating Effectively in Complex Environments

A Technical Perspective on Discovery, Risk, and Collaboration

Khushboo Nigam

Principal Cloud Architect | Oracle

My Background

EDUCATION
M.S. Telecommunications Engineering
University of Maryland, College Park

WORK
Enterprise cloud architecture, technical discovery & adoption

FOCUS
Observability, reliability & practical technical decisions

Connect with me on LinkedIn



The approach I use in my work

Shaped by eight years of working with different enterprise environments

01 **DISCOVER**

Understand current systems, tools, and what information is available

02 **DEFINE THE NEED**

Identify the real challenge and the outcome the team needs

03 **APPLY JUDGMENT**

Compare feasible options, constraints, risks, and tradeoffs

04 **SUPPORT ADOPTION**

Explain the choice, help teams start, and follow through

The original request is a starting point and not always the full problem.

What can others borrow from this approach?

Different systems and responsibilities. Familiar decision challenges.

WHAT I SEE ACROSS ENTERPRISE TEAMS	A HABIT THAT TRANSFERS
No one begins with the full context	**Look beyond your immediate task**
Requests often arrive pre-framed	**Ask which outcome matters**
Real constraints shape the decision	**Make tradeoffs explicit**
A conclusion still needs adoption	**Help others understand and act**

The transferable value is a way of reasoning.

A framework shaped by my everyday work

Four connected habits from initial discovery to a usable result

1

UNDERSTAND

Discover the current environment

What exists?

→

2

CLARIFY

Find the need behind the request

What matters?

→

3

EVALUATE

Apply judgment to options and tradeoffs

What fits?

→

4

ENABLE

Help others use and sustain the result

Can they act?

These are connected questions, not a rigid sequence.

01 UNDERSTAND

Understand the current system

CURRENT STATE

What is in place today?

DEPENDENCIES

What feeds into or depends on it?

PEOPLE

Who operates or uses it?

EVIDENCE

What do we know vs. assume?

TAKEAWAY • Know the current state before designing the next state.

A simple scenario: a dashboard request

Imagine a team asks for a shared dashboard to track open technical issues.

Before building it, what do we need to understand?

CURRENT REPORTING

Teams already send weekly issue updates

INPUTS

Issue status comes from multiple groups

USERS

Who needs updates, and how often?

GAPS

Where are delays or blind spots today?

Understand establishes the situation; it does not decide the solution yet.

02 CLARIFY

Four questions to define the result that actually matters

01 What problem needs to be resolved?

02 Which decision or action is affected?

03 What should change if we address it?

04 How will we know it worked?

A clearly stated outcome gives us something meaningful to evaluate.

CLARIFY: return to the dashboard request

The requested deliverable is not the same as the result we need.

INITIAL REQUEST

“Build a dashboard for open technical issues.”

→

POSSIBLE REAL NEED

“We cannot reliably see unresolved blockers in time.”

→

USEFUL OUTCOME

“Teams identify urgent issues early enough to act.”

A dashboard may be the answer but now we know what it must improve.

03 EVALUATE

Possible ≠ Practical ≠ Sustainable

01

POSSIBLE

Can it work?

Technical feasibility

02

PRACTICAL

Does it fit the context?

Constraints • resources

03

SUSTAINABLE

Will it keep working?

Ownership • operations

TAKEAWAY • Technical feasibility is only the first test.

Evaluate options - not just the first workable answer

1	**FIT**	Does this option solve the defined need?
2	**CONSTRAINTS**	Can we support it with available resources?
3	**ALTERNATIVES**	Is there an existing or simpler path?
4	**TRADEOFFS**	What benefit comes with what cost or risk?
5	**SUSTAINABILITY**	Will it remain useful and maintainable?

EVALUATE: which path best solves the issue?

For the shared technical-issues dashboard, compare realistic approaches.

NEW DASHBOARD

Build a shared view of open blockers

OR

IMPROVE UPDATES

Make existing status updates more timely

OR

TARGETED ALERTS

Flag urgent blockers when they change

Compare fit, effort, and ongoing support before choosing a path.

04 ENABLE

Turn a sound decision into a usable result

DECISION

A sound technical conclusion

→

UNDERSTANDING

Reasoning, evidence and limitations

→

ACTION

A clear path for the next team

Enablement means others can understand, adopt, and act on the result.

Follow through until others can move forward

01

CHECK UPTAKE

Are people able to use it?

02

REMOVE BLOCKERS

What still needs resolution?

03

VERIFY OUTCOME

Is the intended result emerging?

A handoff matters when the next person can take action.

ENABLE: make the decision usable for the team

Whichever approach is chosen, it must help people act on urgent blockers.

SHARED DEFINITIONS

Agree what counts as urgent or unresolved

→

CLEAR OWNERSHIP

Who updates status and who responds?

→

CHECK THE OUTCOME

Can teams spot issues and act sooner?

Enablement succeeds when the next team can use the result independently.

Where AI fits into my technical workflow

It helps me handle diverse requests—but I still verify the answer.

EXPLORE

Review use cases and scan product documentation for possible approaches

VALIDATE

Check official sources, recent changes, and test capabilities directly before advising

SHARE

Turn verified learning into useful explanations, notes, articles, and technical guidance

AI accelerates my research; responsibility for the recommendation remains with me.

Make your technical work visible and reusable

CAPTURE WHAT YOU LEARN

- Problems and solutions
- Technical insights
- Decisions and tradeoffs
- Lessons worth sharing

→

TURN IT INTO SOMETHING USEFUL

GUIDE	Knowledge articles or guides
DEMO	Walkthroughs or demos
TEAM SESSION	Peer learning and sharing
ARTICLE / TALK	Technical articles or talks

TAKEAWAY • Make your impact visible by sharing what others can learn.

The work behind the visible result

ASK

Sharpen the real question

CONNECT

Bring missing context

ANTICIPATE

Surface risk early

ENABLE

Help others take action

Not every important technical contribution looks like a finished build.

TAKE THIS BACK TO WORK

Better decisions. More usable outcomes.

UNDERSTAND	CLARIFY	EVALUATE	ENABLE
See the wider system	Name the real need	Test options and risks	Make the result usable

Share the reasoning—so others can build on the result.

AUDIENCE APPLICATION / 5 MINUTES

Try the four steps on a decision from your work

UNDERSTAND

What system or dependency am I overlooking?

CLARIFY

What outcome are we really seeking?

EVALUATE

Which risk or tradeoff needs attention?

ENABLE

Who needs to understand or act next?

Choose a real example • Pair up • Share one insight

WOMEN IN ENERGY 2026

Questions & discussion

Thank you for joining the conversation.

Khushboo Nigam | Principal Cloud Architect, Oracle

Your feedback matters

