

Supporting a Product Changes: A Practical Framework

For new releases, major updates, new products, and pivots

“If your product is changing, your support model needs to be designed—not improvised.”

Built from V1 product experience. Applicable to any major change.
Lala Mamedov | The Support Strategy

Design Principle

Pre-emption starts before Day One

- We are developing pre-emptive support
 - Define how issues will be resolved before they occur
 - Design knowledge capture into the workflow from the beginning
 - Define how product feedback will be generated and used
 - Assume friction will exist and design for it
-

Pre-emption is **not a later stage**.

It is a design choice from the start.

If you wait until after launch, you have slipped into the reactive more

The Four Questions

1

What is the support strategy?

2

What is the demand for support?

3

How will issues be resolved?

4

How do we pre-empt issues?

Start here – not with tools or staffing.

Strategy

Define the role of support.
Start with a mission statement:

What business is support in?

Examples:

We are in the business of helping our customers achieve the full benefit of using our product, so they can be more productive and successful.

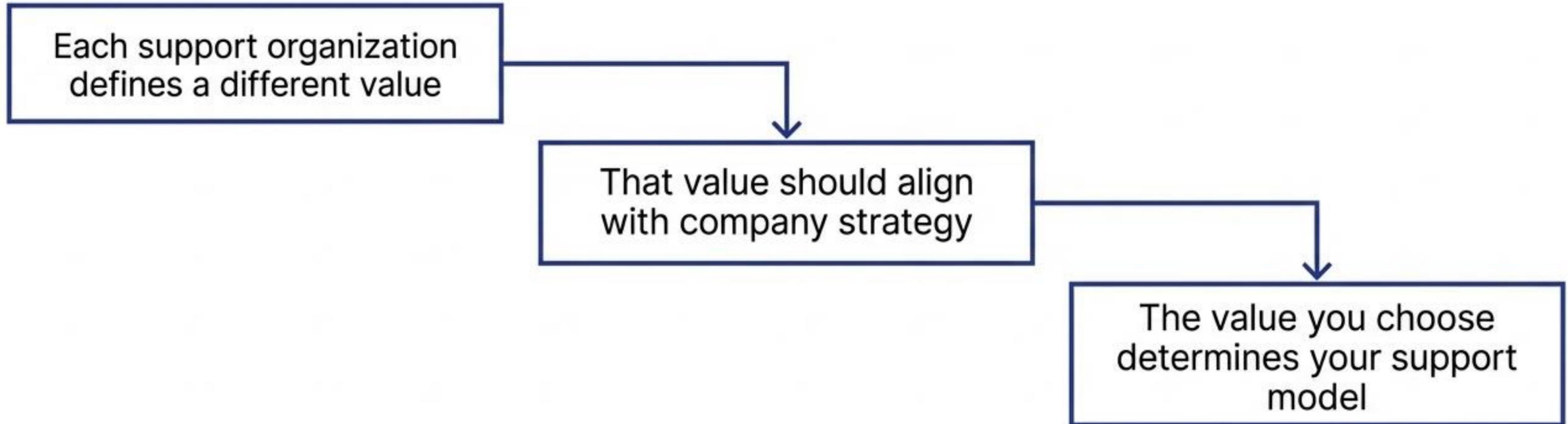
↳ **Focus: customer productivity**

We are in the business of helping customers receive and perceive the full value of the product.

↳ **Focus: product understanding and education**

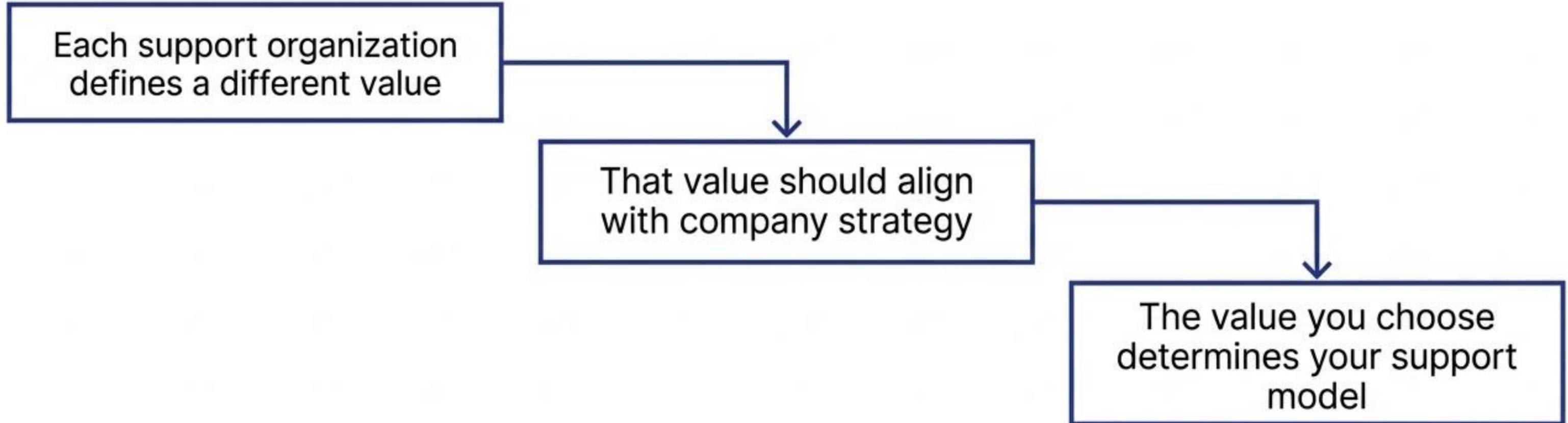
A clear mission statement grounds all subsequent support decisions.

Why this matters:



Support strategy is defined by the value you choose to deliver.

Why this matters:



Support strategy is defined by the value you choose to deliver.

Decide

- What support do you offer (free, paid, tiers)?
- Where do customers go for help (channels)?
- How fast can you respond (SLA)?
- What do customers expect?
- Who owns the customer experience?

CRITICAL DISTINCTION

**"Response time is not just what you can deliver—
it must align with customer expectations."**

Aligning capabilities with customer expectations is the core of the decision process.

Design for the right level

Do not go below
expectations

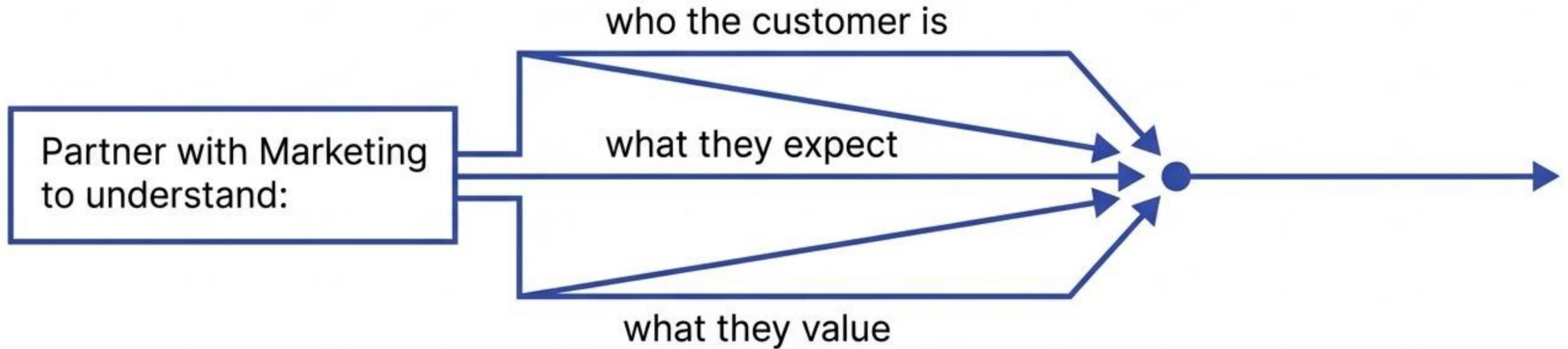
Do not over-deliver
without purpose

Align with:

customer profile
technical sophistication
use case criticality

Input matters

Customer expectations come from persona and positioning



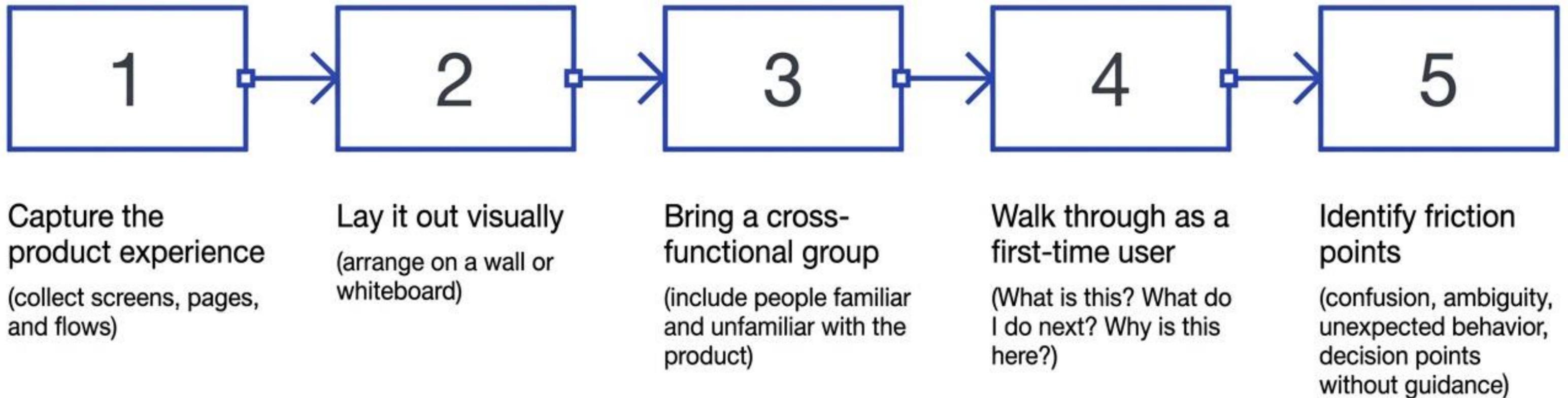
Support design balances capability with expectation.

Demand

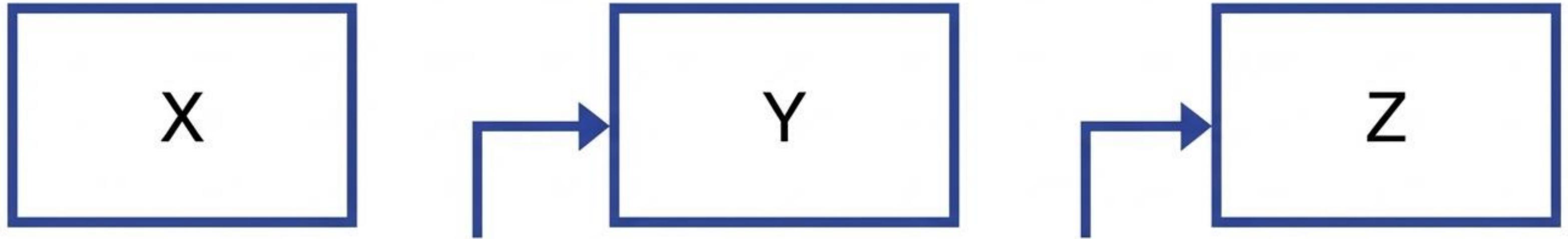
Start with the product journey map—not the customer journey.

Map how the product behaves in real use—not how customers move through it.

How to build a product journey map



Convert friction into use cases



User attempts X → cannot complete Y because Z

How to estimate demand

Use alpha experience to inform demand.
By beta, you are already behind.

How to estimate demand (continued)

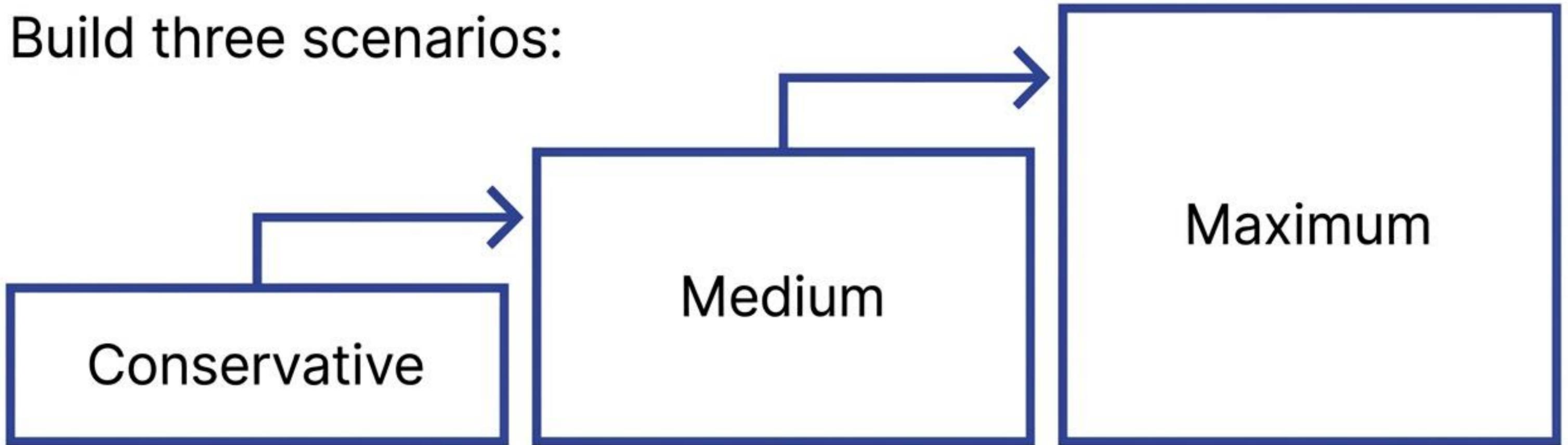
1	Define users and segments
2	Assign likelihood (high / medium / low)
3	Assign impact (blocking / degrading / minor)
4	Estimate contact behavior

Support demand is product friction expressed as volume.

Demand Modeling

Modeling demand under uncertainty

Build three scenarios:



This creates a working range without false precision.

How to estimate demand (continued)

Factor in self-service maturity

knowledge improves over time



more issues resolved without agents



assisted demand decreases

Self-service vs. deflection

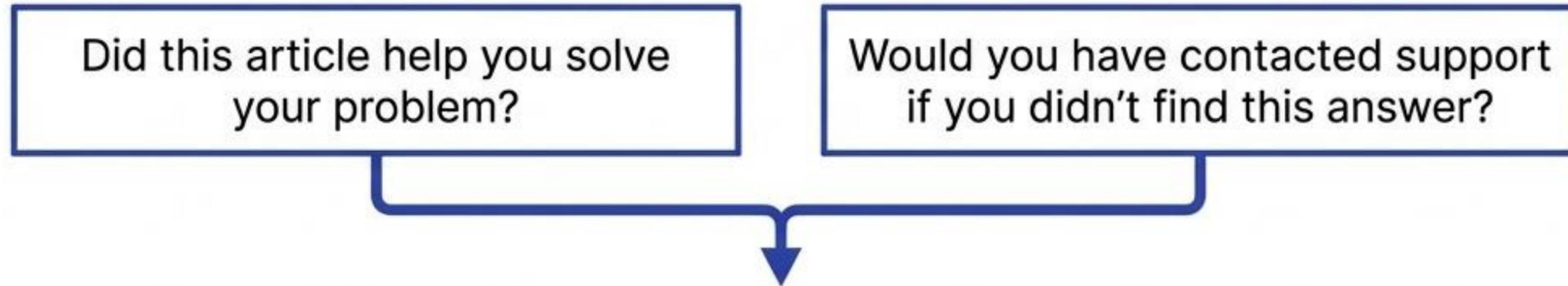
some users would contact support → true deflection

some would not → net-new value

some tolerate the issue

How to measure deflection

At the end of a knowledge base article, ask:



**Deflection = users who solved their issue
AND would have contacted support**

Why this matters

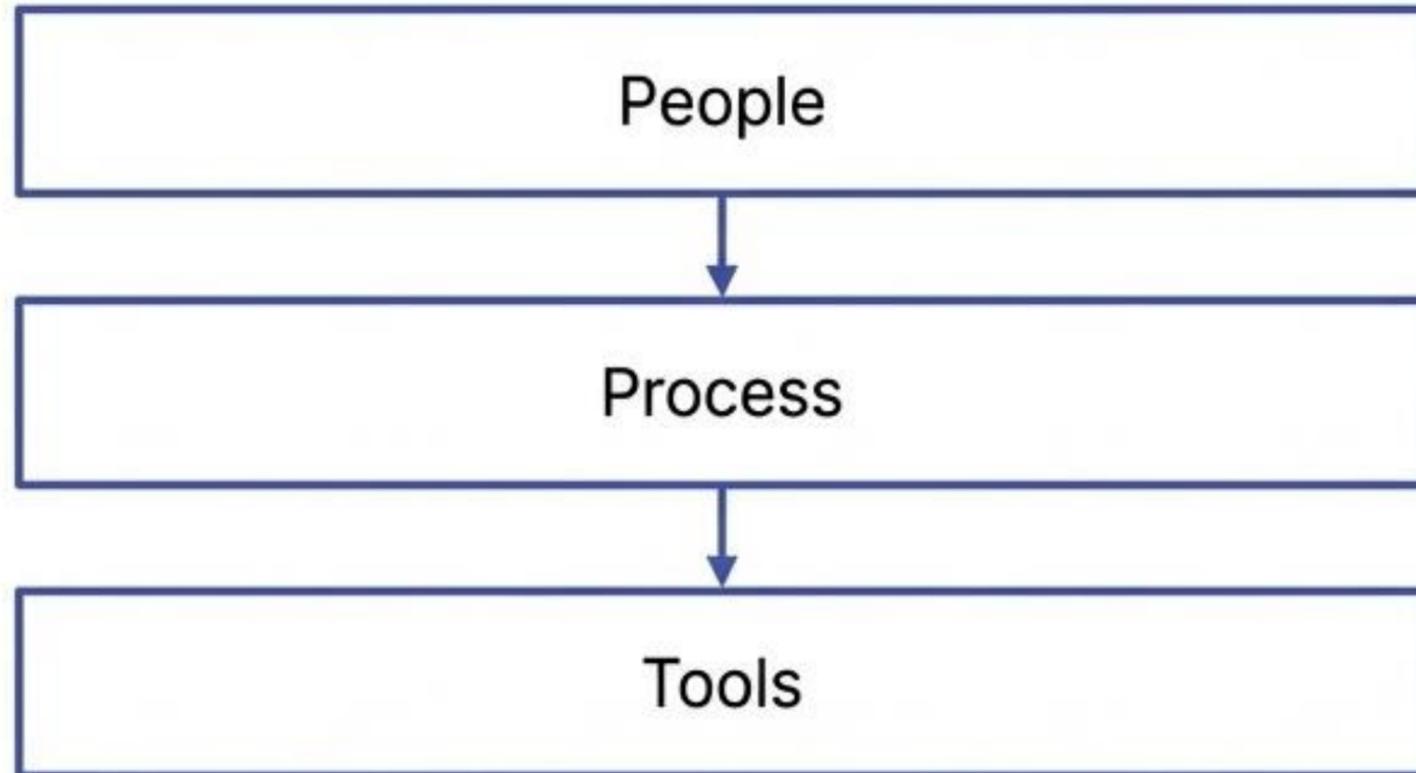
- supports users who never contact support
- reduces friction and frustration
- expands support coverage

"Self-service is both deflection and expanded support coverage."

Resolution — design for learning

“Early-stage support is not about efficiency—it’s about learning fast.”

Structure



Key distinction

Incident = customer problem	Issue = product problem

Support handles incidents. Product fixes issues.

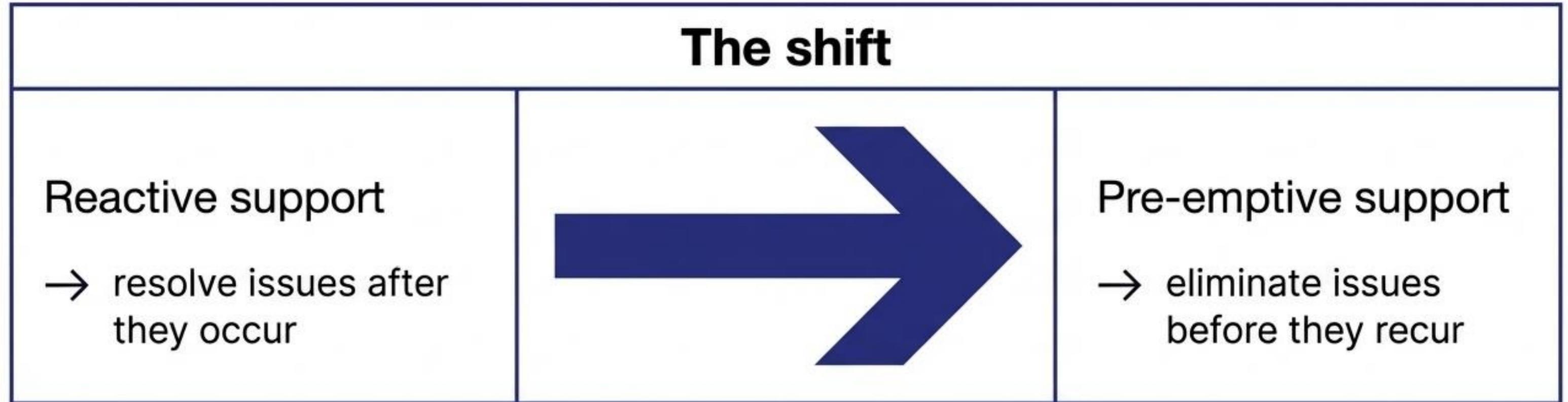
Feedback loop



Resolution generates knowledge—not just answers.

Pre-emption – the goal of support

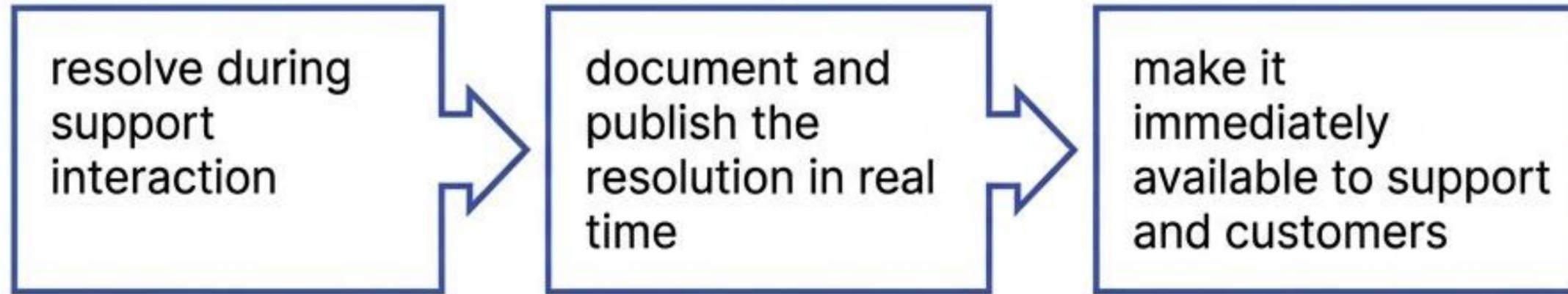
Move from resolving issues to pre-empting them.



**The goal is not to solve more cases.
The goal is to create fewer cases.**

Resolving new issues

New issues = no existing knowledge



Knowledge creation is part of the workflow—not a separate step.
(Aligned with Knowledge-Centered Service (KCS) practices)

What that knowledge does

1. Improves support capability

- shared knowledge
- faster, consistent resolution

2. Drives product improvement

- track recurrence and impact
- feed product roadmap
- eliminate root causes

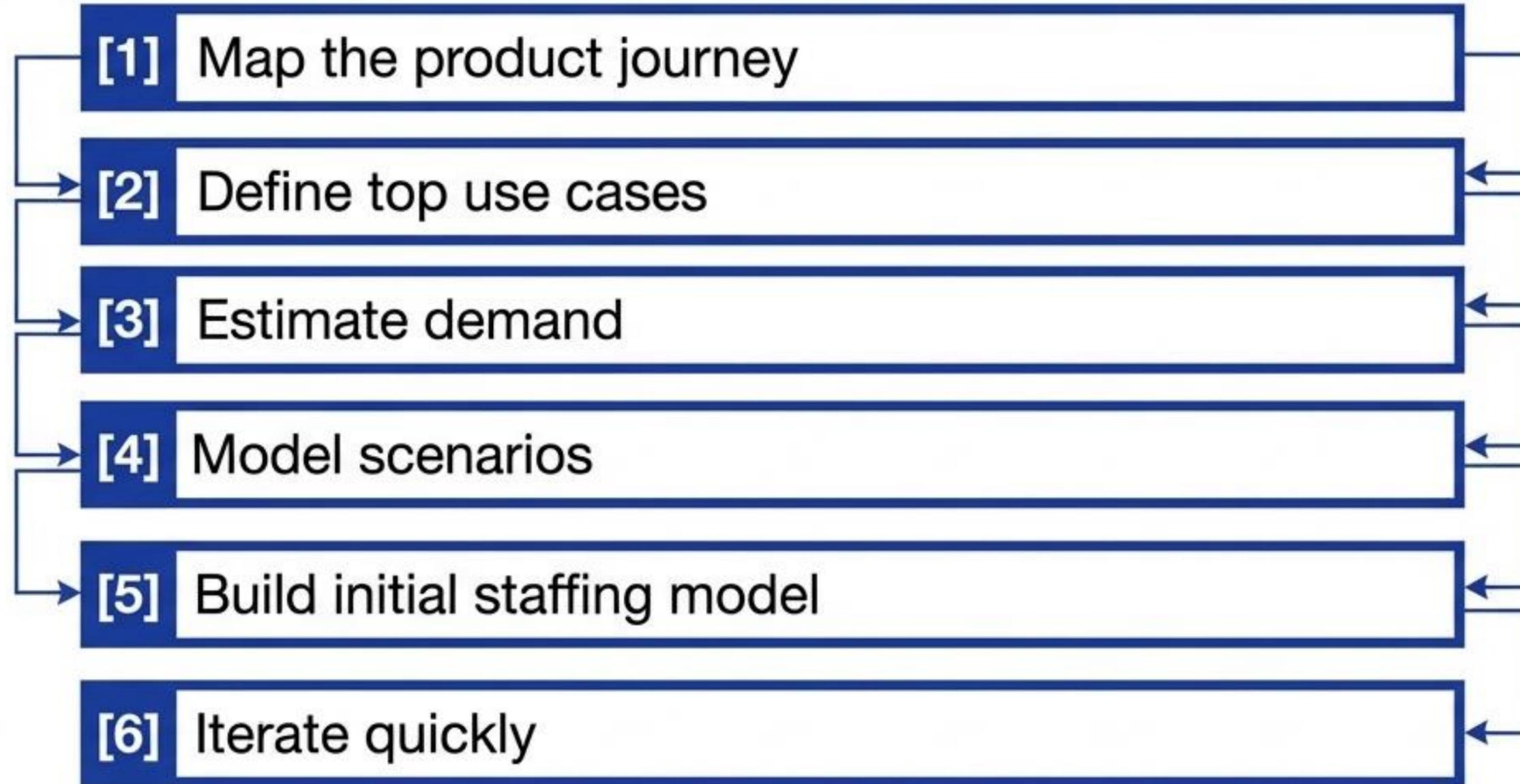
The learning loop



Guiding principle

“Never solve the same problem twice.”

How to apply



You are designing under uncertainty.

Takeaway

This framework helps you ask the questions
your support strategy depends on.

V1 support is a learning system—not a fixed design.
