Modes of problem solving

It's ok to not know what the problem is when asking for help!

You might go back and forth between these modes

What is the real problem?

Useful when: Untangling issues, challenges or when something feels off
Do: Explore causes, not just symptoms. Stay in discovery, understand users
Don't: jump to solutions.

Typical activities

- User / stakeholder interviews, talk to people!
- Desk research, data analysis
- Journey / process mapping
- Observations, shadowing
- Question Banking

Useful methods

- 5 Whys
- Problem tree / fishbone diagram
- "What's really going on here?" reframing
- Systems mapping / context mapping

How might we solve it?

Useful when: you understand the problem and want to explore different technical and product options.
Do: Generate options and understand constraints.
Don't: start actually delivering the solution yet.

Typical activities

- Co-design workshops, bring people together!
- Sketching ideas, storyboards, simple mock-ups
- Talk to other people who've done similar
- Mapping risks, assumptions and constraints

Useful methods

- "How might we…?" statements
- Crazy 8s / rapid sketching
- Impact vs effort matrix
- Option appraisal grids

Try to solve it

Useful when: You have a clear sense of what the problem is and options to try out.
Do: Move from options to delivery. Show the thing.
Don't: give up if things do feel solved immediately

Typical activities

- Plan, build, deploy
- Communicate changes, adjust policies, train staff
- Launch a collaborative experiment

Useful methods

- Regular check ins, learning huddles
- Delivery frameworks, visual trackers
- A/B tests or simple "before / after" comparisons.

Learning should be integral and throughout. It underpins everything

What did we learn?

This isn't linear but modes help to keep the focus in the right place

Useful when: You have tried something
Do: Reflect and ask - Did we pick the right problem, choose the right approach, and deliver it in a fair, workable way?
Don't: make this an afterthought or think of this as a judgement

Typical activities

- Retrospectives and after-action reviews, User feedback, surveys, interviews, Metrics and outcome reviews Learning logs, reflection sessions, learning loops

Modes of problem solving

It's ok to not know what the problem is when asking for help!

You might go back and forth between these modes

What is the real problem?

Useful when: Untangling issues, challenges or when something feels off

Do: Explore causes, not just symptoms. Stay in discovery, understand users

Don't: jump to solutions.

How might we solve it?

Useful when: you understand the problem and want to explore different technical and product options.

Do: Generate options and understand constraints.

Don't: start actually delivering the solution yet.

Try to solve it

Useful when: You have a clear sense of what the problem is and options to try out.

Do: Move from options to delivery. Show the thing.

Don't: start here unless you've got a clear understanding of the problem

What did we learn?

Learning should be integral and throughout. It underpins everything

Useful when: You have tried something
Do: Reflect and ask - Did we pick the right problem, choose the right approach, and deliver it in a fair, workable way?
Don't: make this an afterthought or think of this as a judgement

Modes of problem solving

It's ok to not know what the problem is when asking for help!

You might go back and forth between these modes

What is the real problem?

Useful when: Untangling issues, challenges or when something feels off
Do: Explore causes, not just symptoms. Stay in discovery, understand users
Don't: jump to solutions.

Typical activities

- User / stakeholder interviews, talk to people!
- Desk research, data analysis
- Journey / process mapping
- Observations, shadowing
- Question Banking

Useful methods

- 5 Whys
- Problem tree / fishbone diagram
- "What's really going on here?" reframing
- Systems mapping / context mapping

How might we solve it?

Useful when: you understand the problem and want to explore different technical and product options.
Do: Generate options and understand constraints.
Don't: start actually delivering the solution yet.

Typical activities

- Co-design workshops, bring people together!
- Sketching ideas, storyboards, simple mock-ups
- Talk to other people who've done similar
- Mapping risks, assumptions and constraints

Useful methods

- "How might we...?" statements
- Crazy 8s / rapid sketching
- Impact vs effort matrix
- Option appraisal grids

Try to solve it

Useful when: You have a clear sense of what the problem is and options to try out.
Do: Move from options to delivery. Show the thing.
Don't: give up if things do feel solved immediately

Typical activities

- Plan, build, deploy
- Communicate changes, adjust policies, train staff
- Launch a collaborative experiment

Useful methods

- Regular check ins, learning huddles
- Delivery frameworks, visual trackers
- A/B tests or simple "before / after" comparisons.

Learning should be integral and throughout. It underpins everything

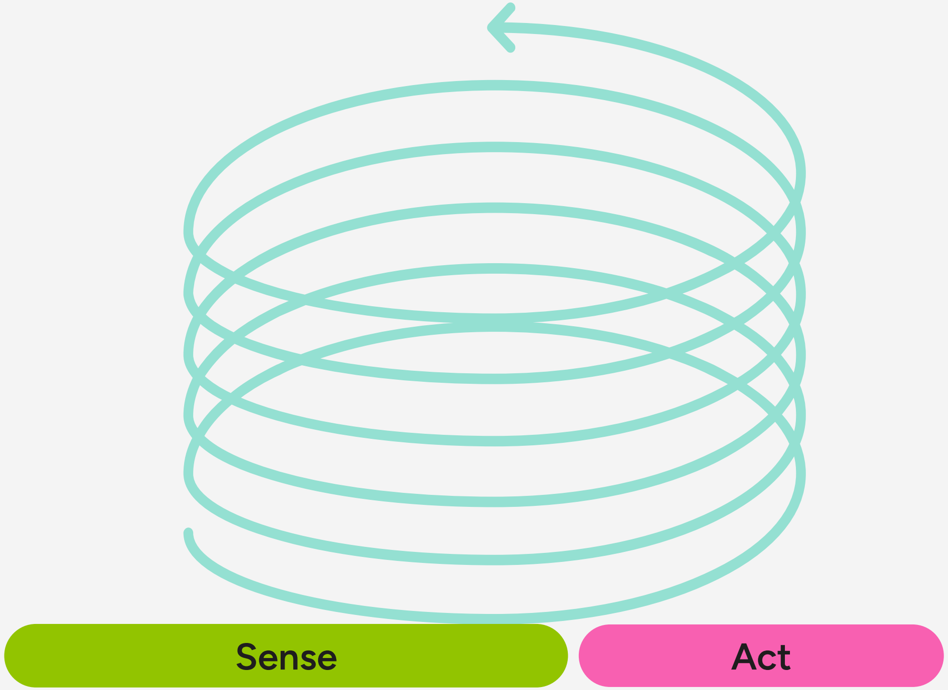
What did we learn?

Useful when: You have tried something
Do: Reflect and ask - Did we pick the right problem, choose the right approach, and deliver it in a fair, workable way?
Don't: make this an afterthought or think of this as a judgement

Typical activities

- Retrospectives and after-action reviews, User feedback, surveys, interviews, Metrics and outcome reviews Learning logs, reflection sessions, learning loops

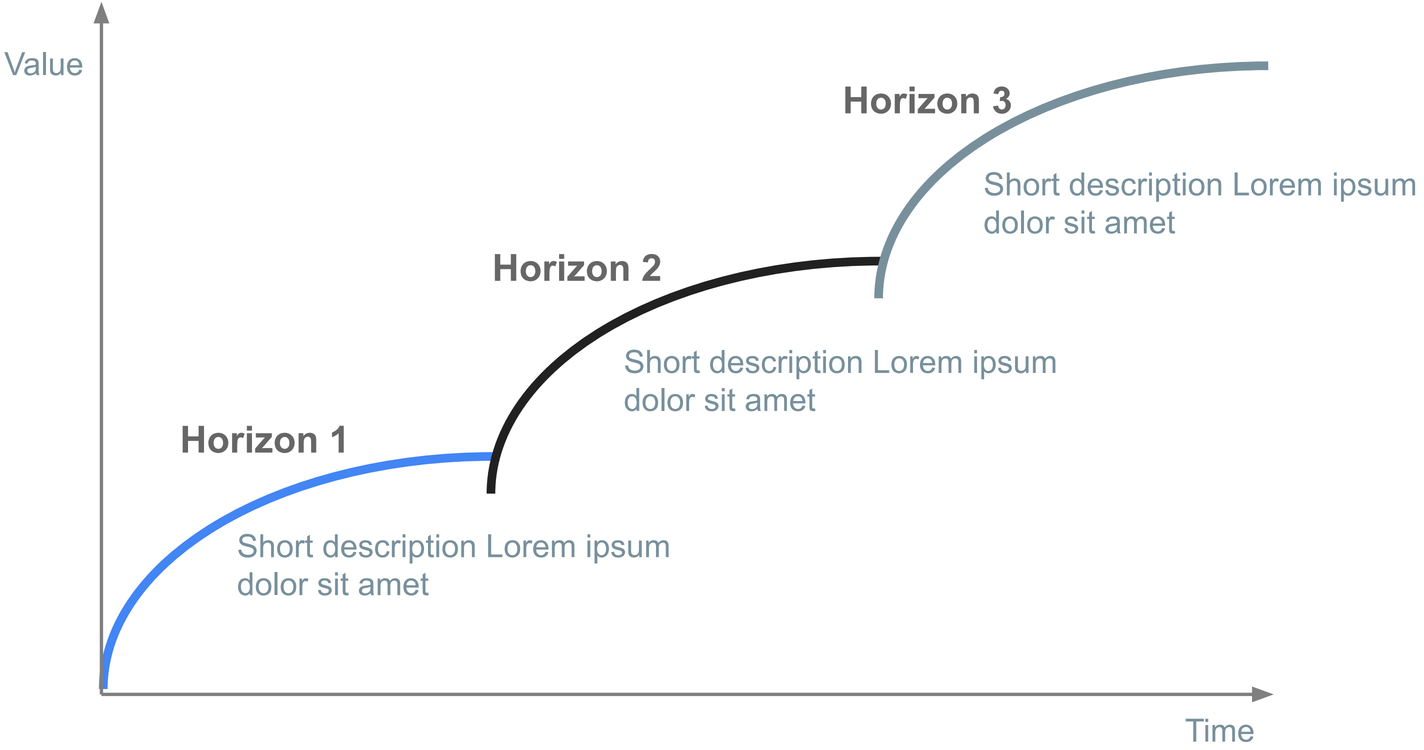
This isn't linear but modes help to keep the focus in the right place



Sense

Act

Horizons framework



Value

Horizon 3

Short description Lorem ipsum dolor sit amet

Horizon 2

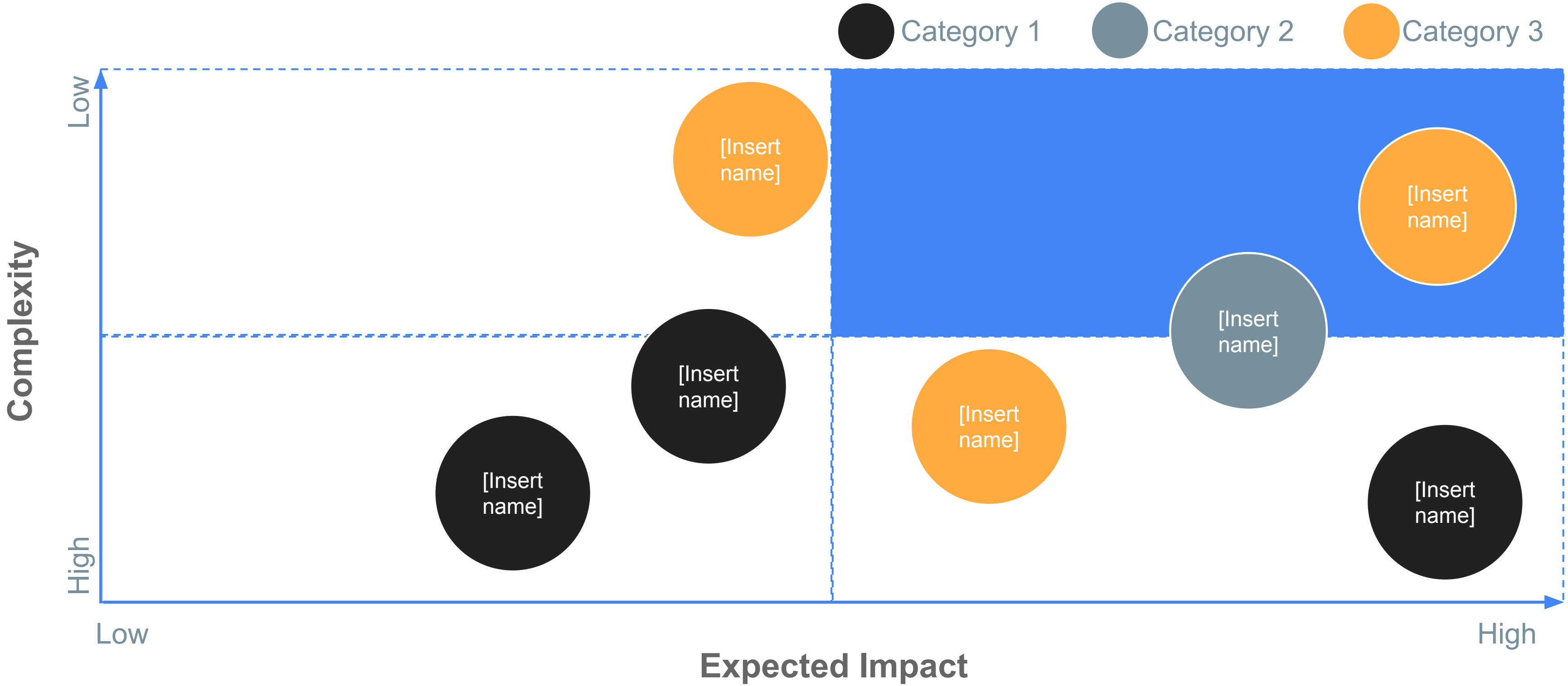
Short description Lorem ipsum dolor sit amet

Horizon 1

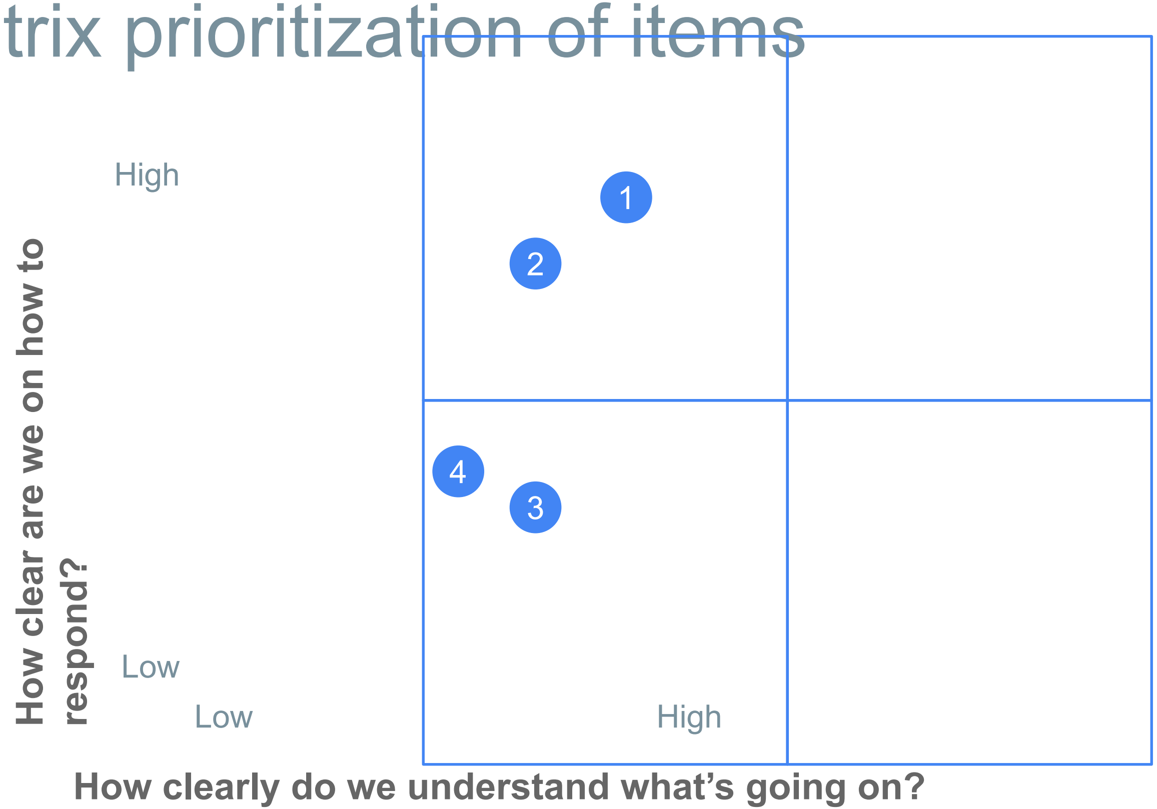
Short description Lorem ipsum dolor sit amet

Time

4 box prioritization framework



Matrix prioritization of items

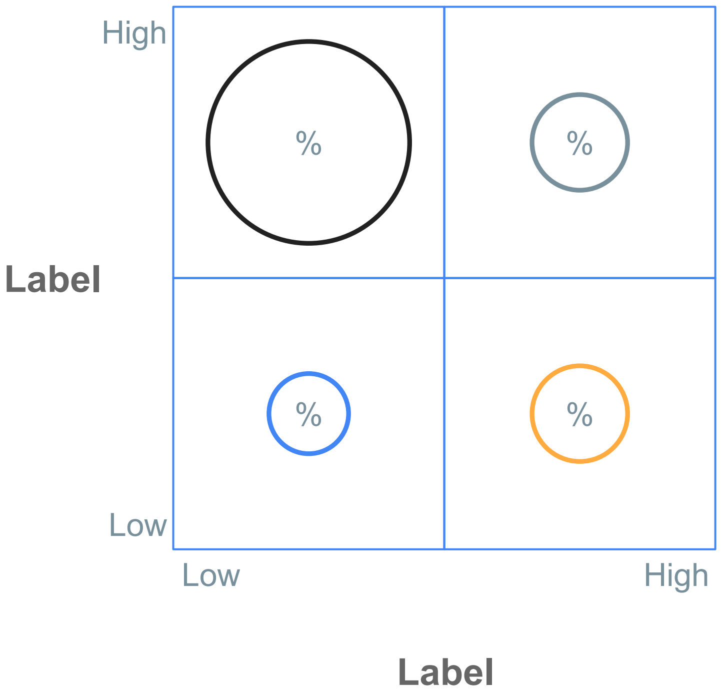


High
1
2
4
3
How clear are we on how to respond?
Low
Low
High
How clearly do we understand what's going on?

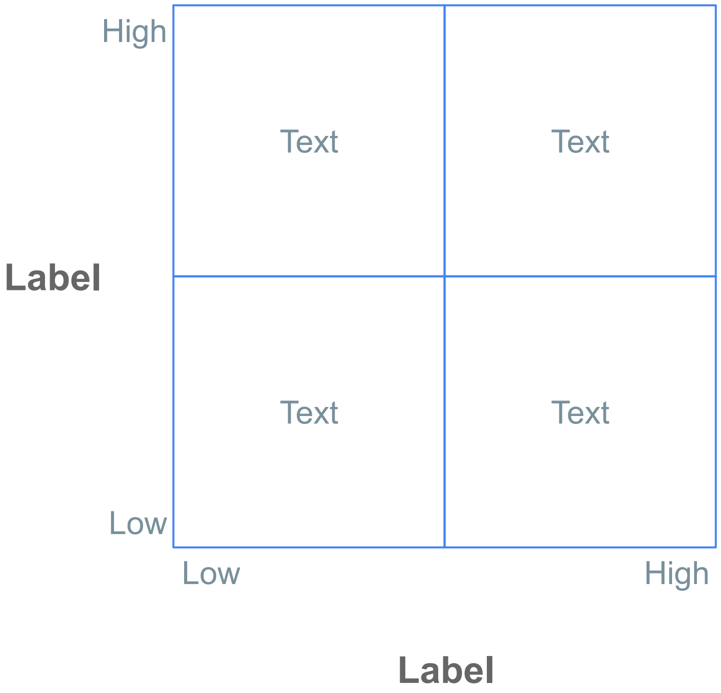
Prioritized

- 1 Description
- 2 Description

1 Matrix with proportions



1 Matrix



High

Text

Text

Label

Text

Text

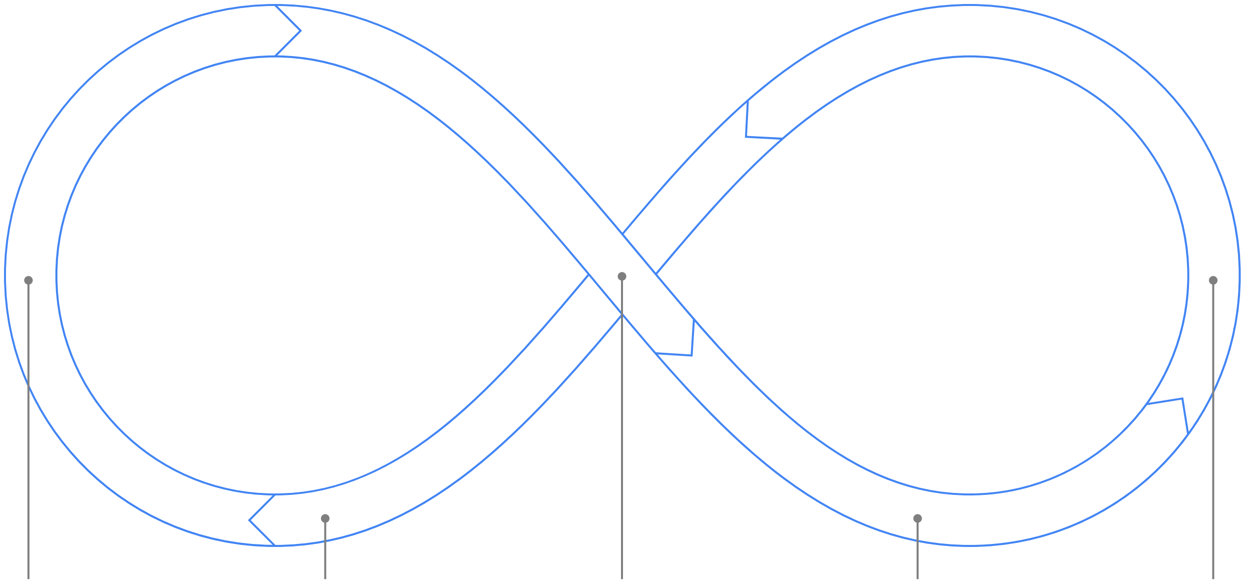
Low

Low

High

Label

5 step infinite loop process



Heading

Description

Heading

Description

Heading

Description

Heading

Description

Heading

Description