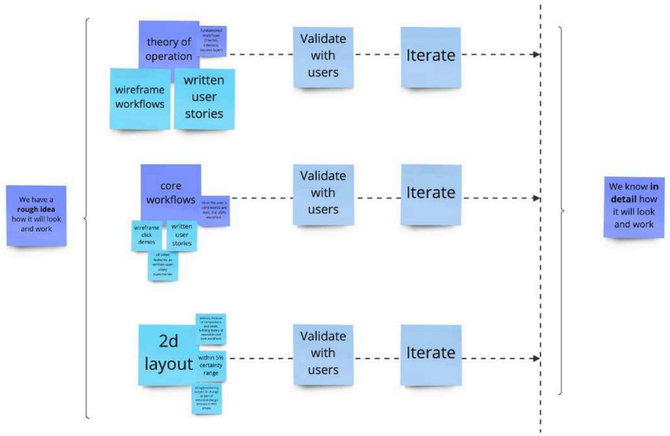
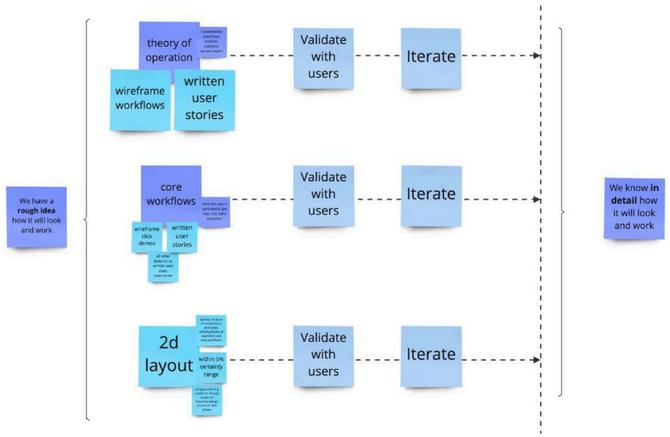
HW/SW Product/UX Design @ Native Instruments

Ant Orant 2024

(Product context) MASCHINE is an all-in-one music production and performance system



1. Product concept ideation roadmap

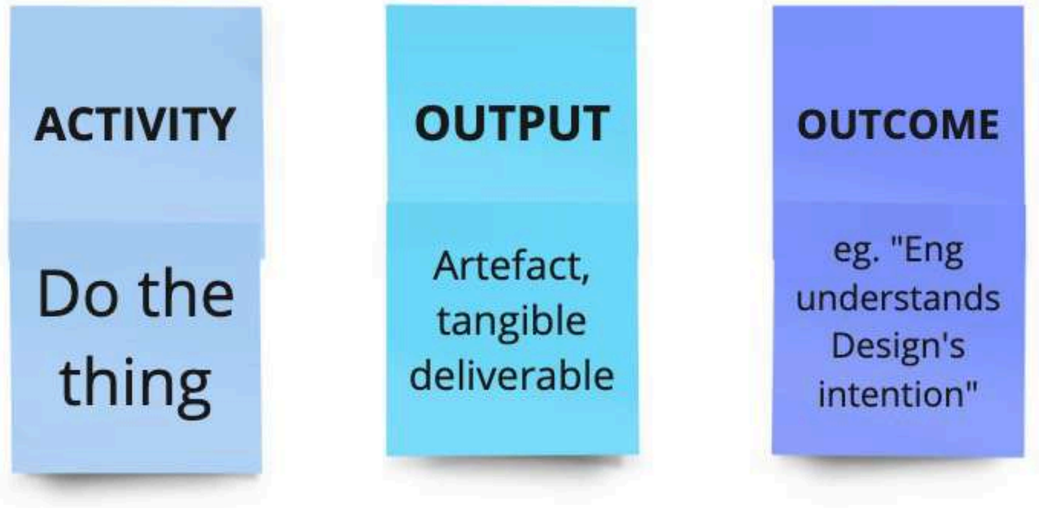


With the Product team concluding an improvised, engineering-centric ideation phase, about to begin work on the next product, the designers needed to define and sequence the key milestones in order to deliver a strong concept within a much tighter deadline.

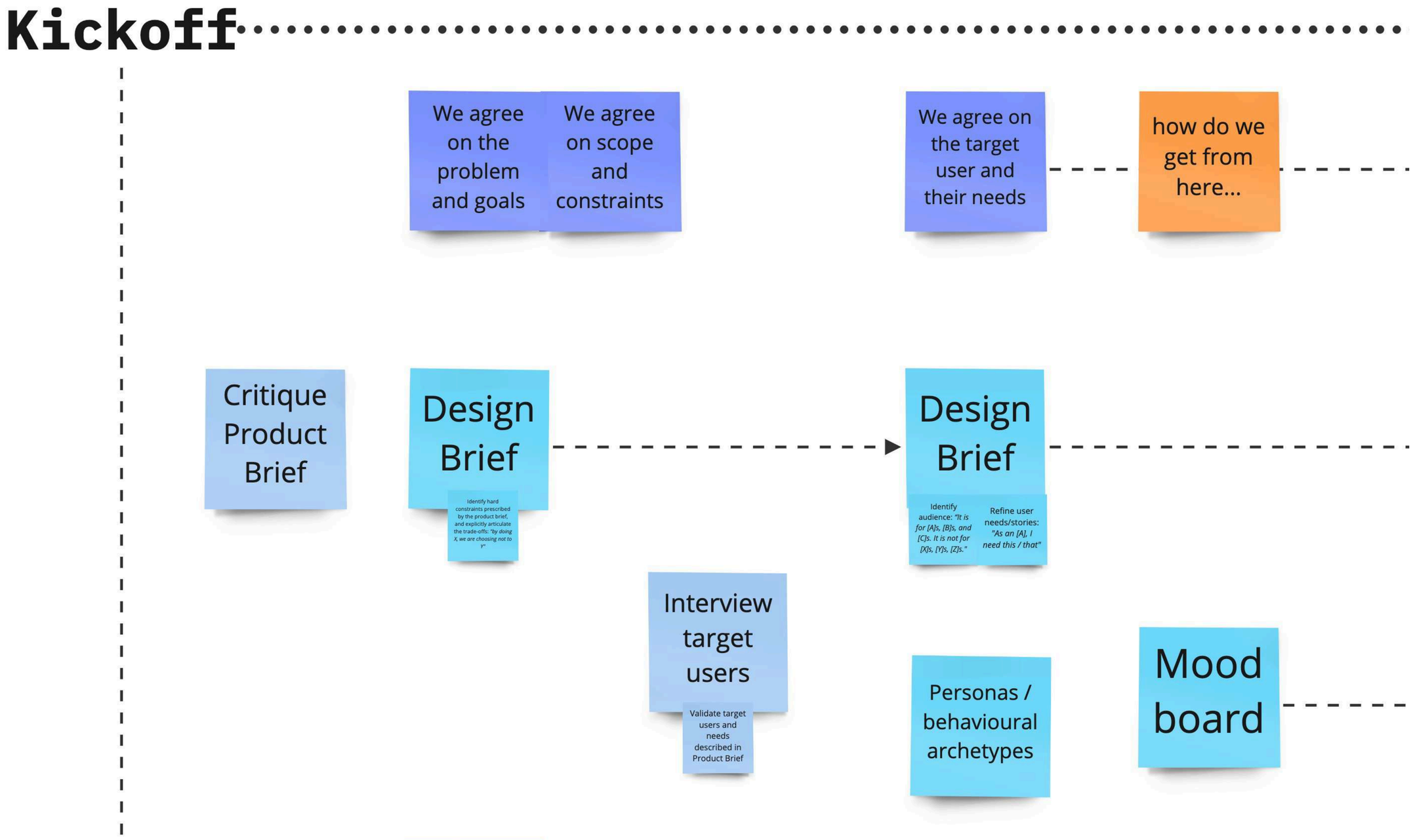
I reviewed assets and communication from the previous cycle, identified the core artefacts and activities, then grouped and ordered them chronologically.

The Design team entered the next cycle with clear priorities and realistic goals.

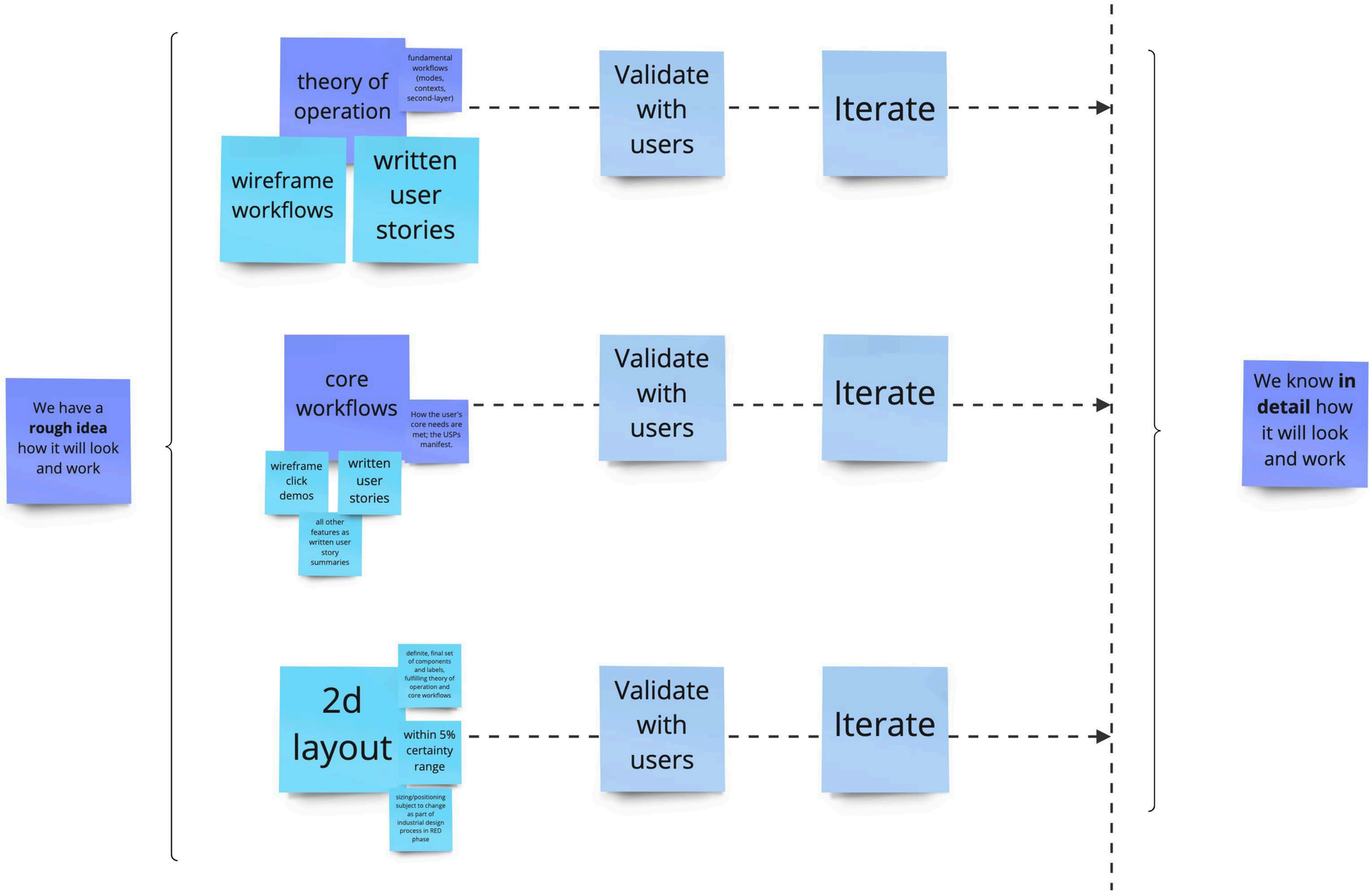
“Outcomes over output”



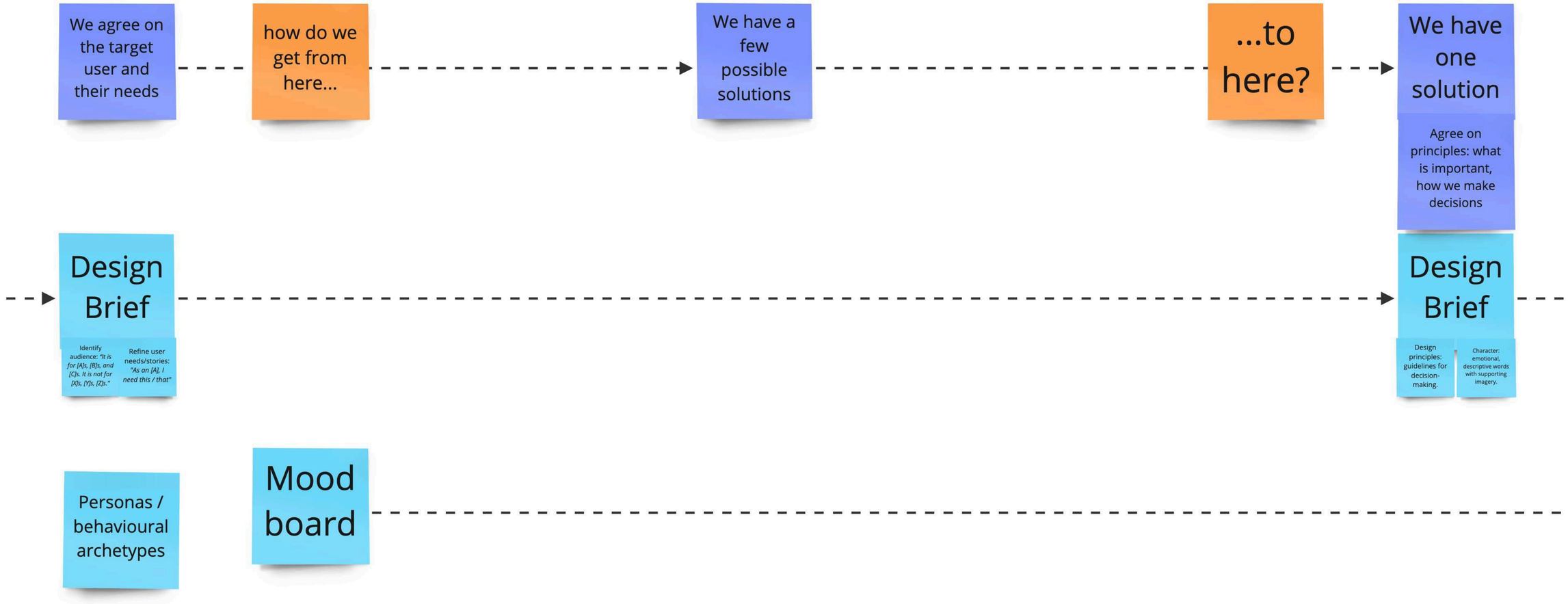
Structuring the initial phase



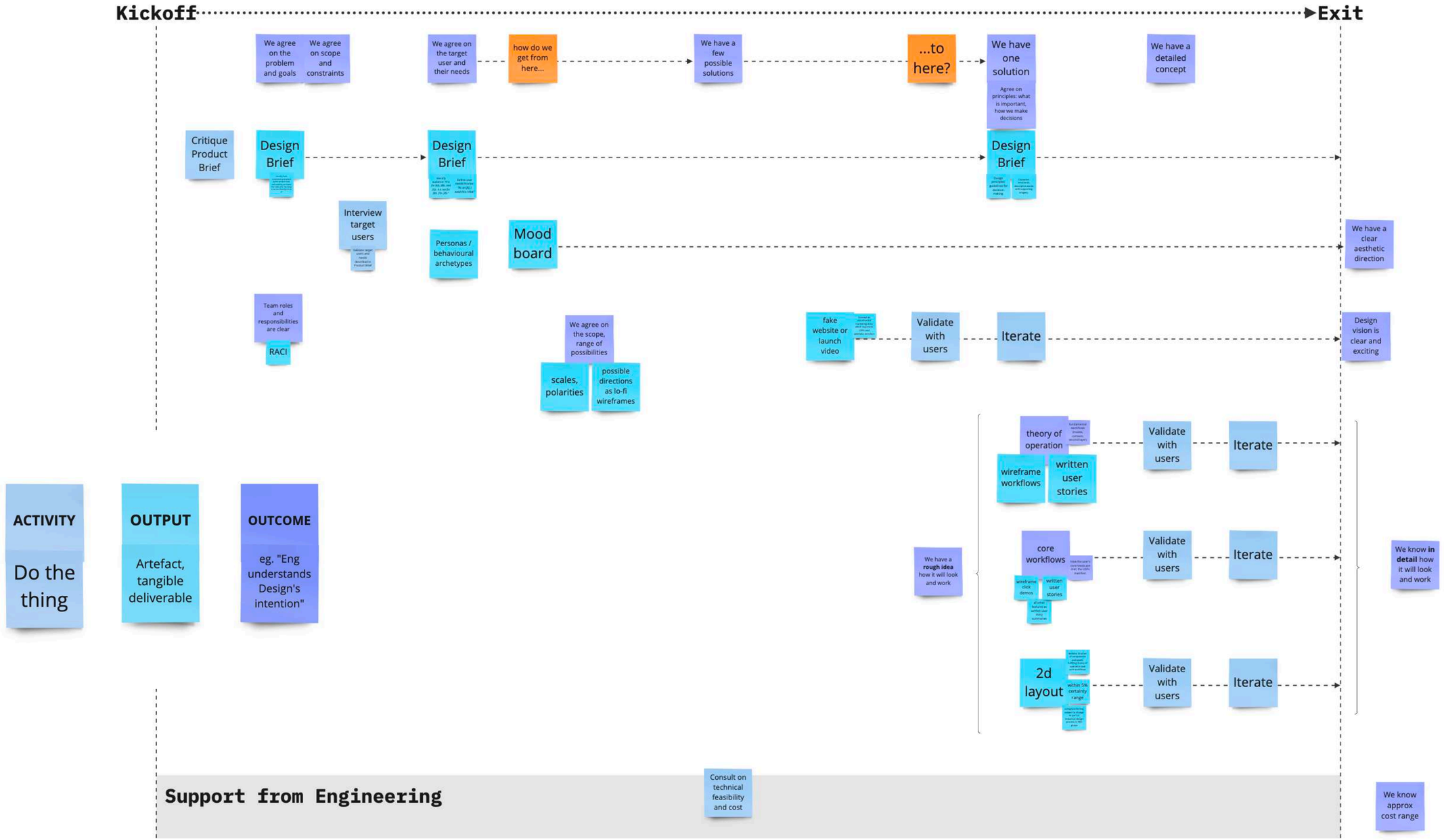
Iterative shaping of core concept

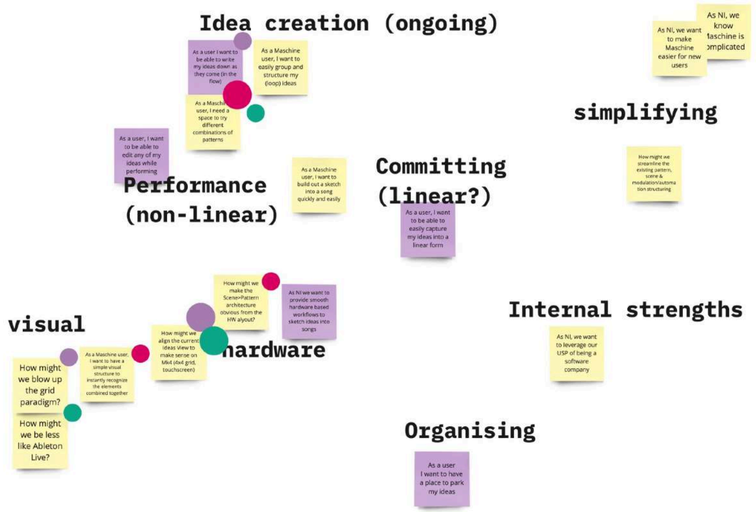


Mapping conceptual development

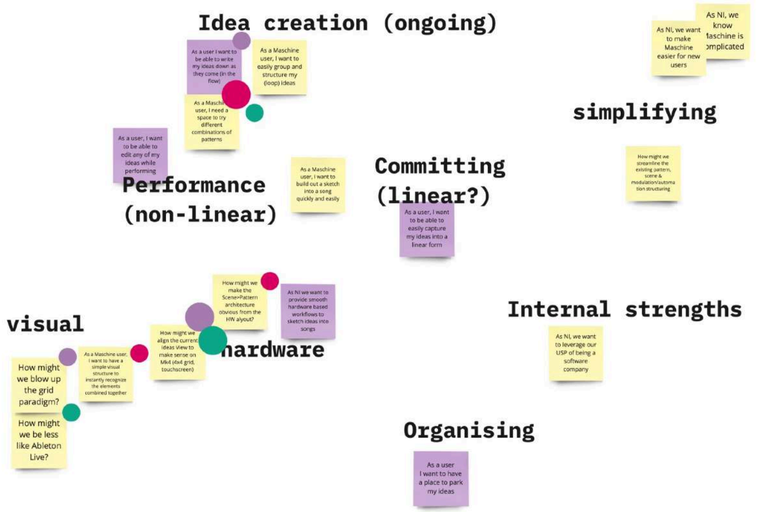


The entire concept ideation phase





2. “Ideas” view exploration sprint

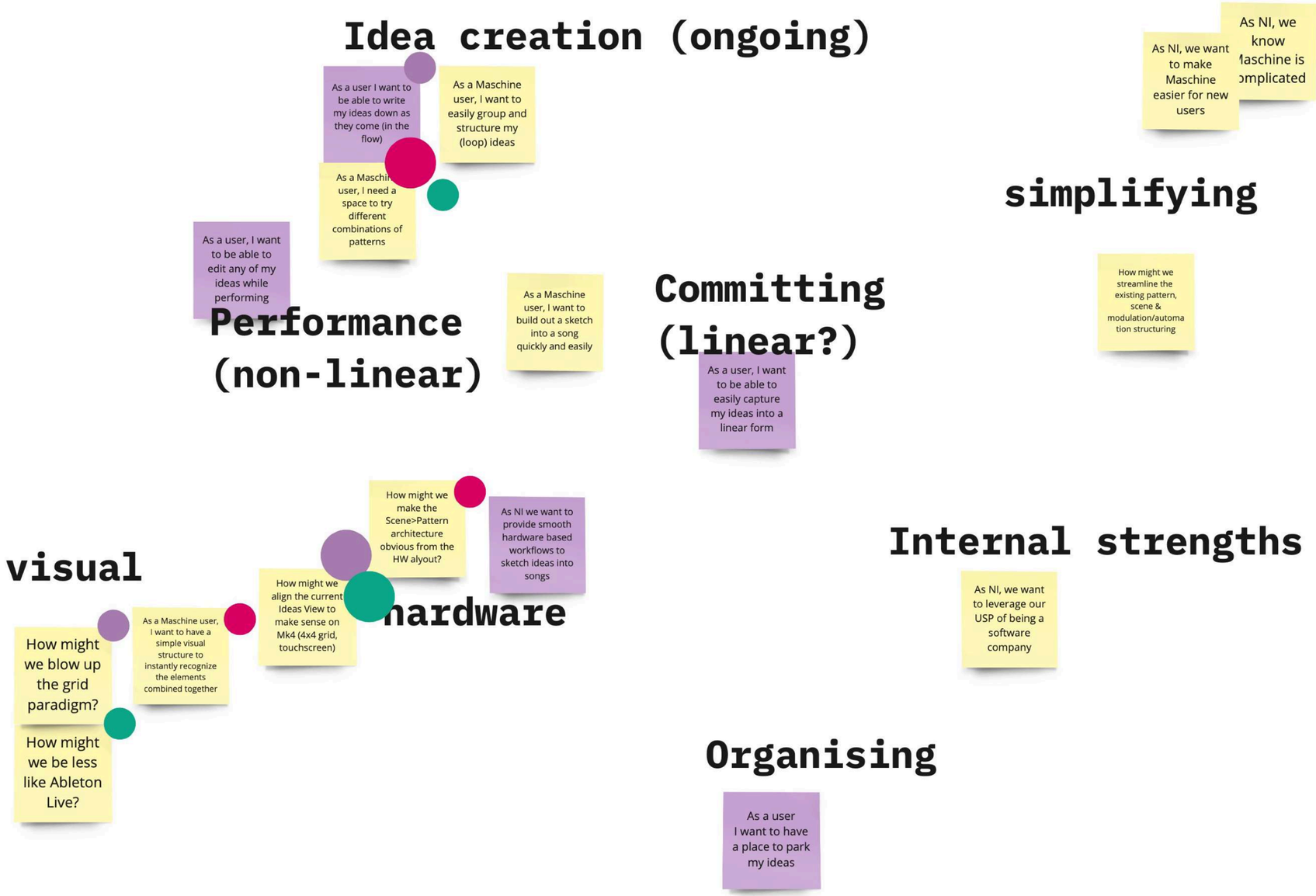


Approaching ideation exit with a long wish-list of features and limited engineering resources, the team informally identified the “Ideas” feature as a critical UX improvement, but with a lack of clarity around scope, effort and feasibility.

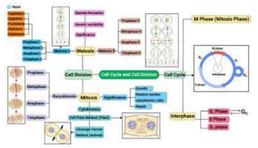
I led a two-day workshop with key product team members, with the goal of capturing and structuring the various concerns, advocacy, and technical considerations, then quickly exploring the range of possible changes.

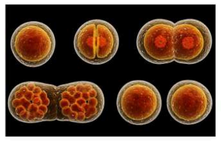
The problem context and detail were documented in a shareable format, with potential solutions shaped and scoped, prioritised by confidence and effort, leading to a clear decision.

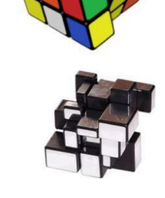
Problem space exploration

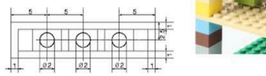


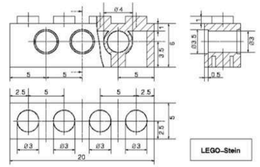
Exploration, benchmarking, rapid ideation

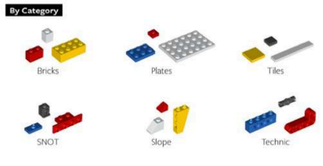


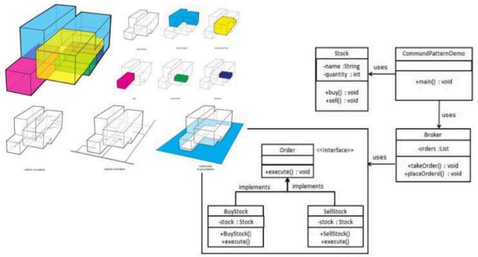


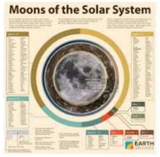


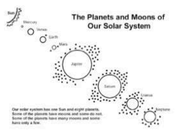


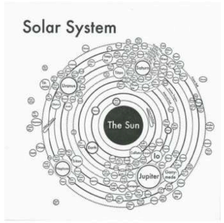


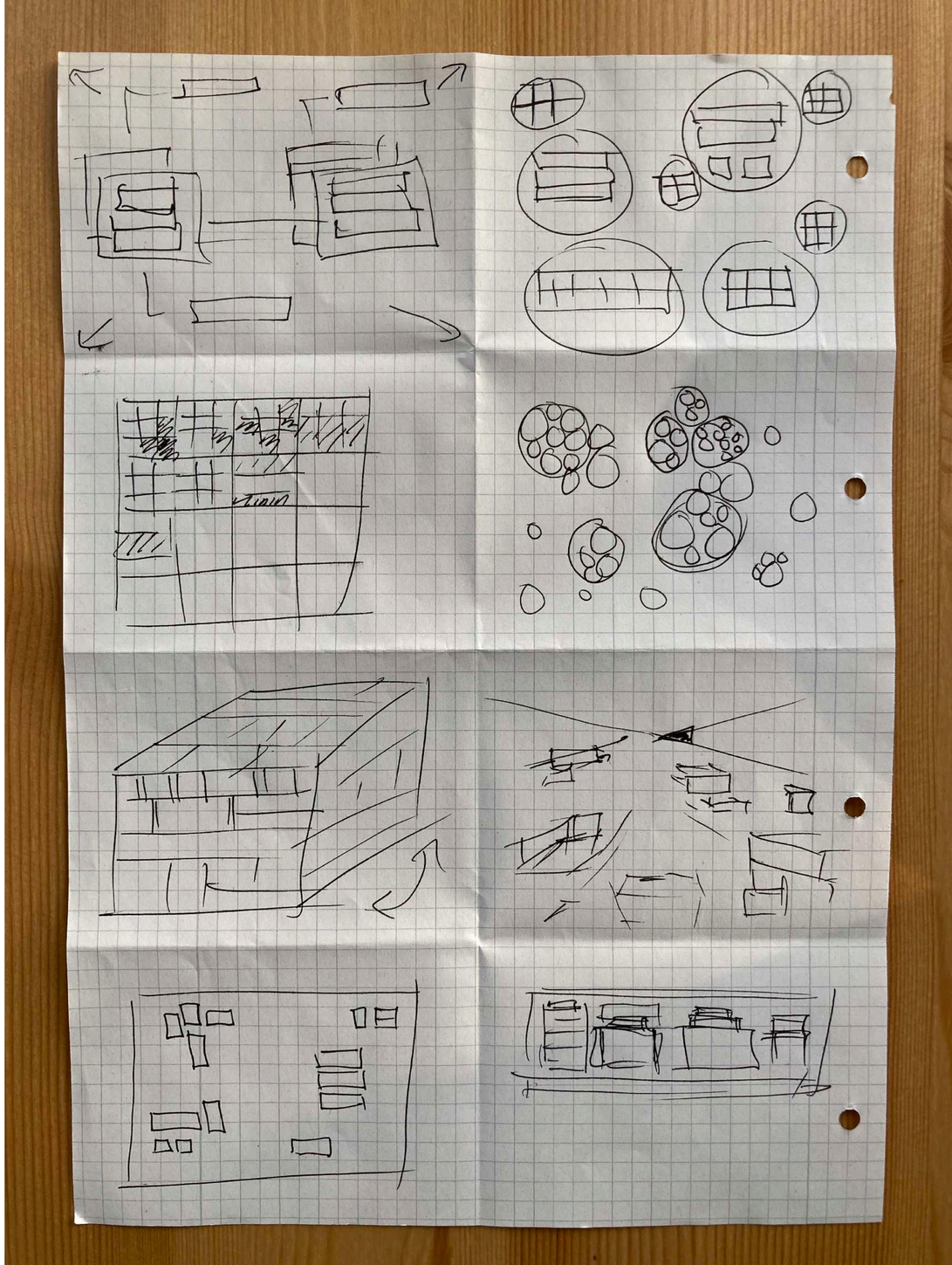




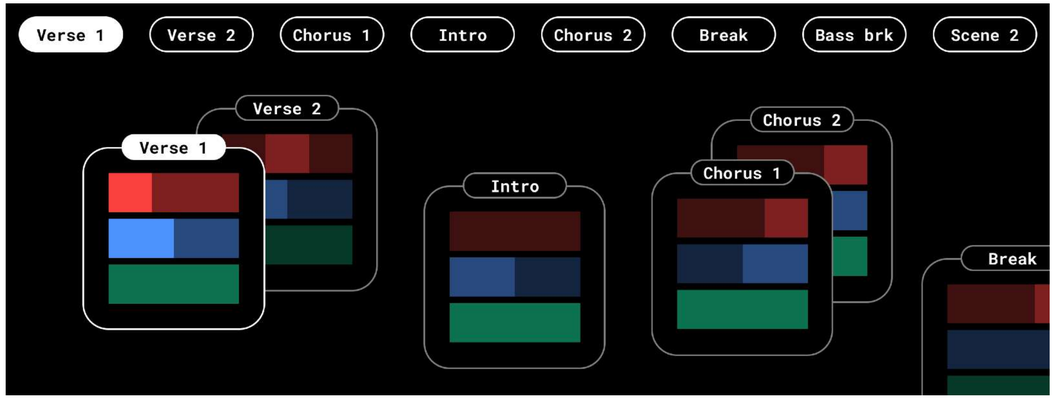


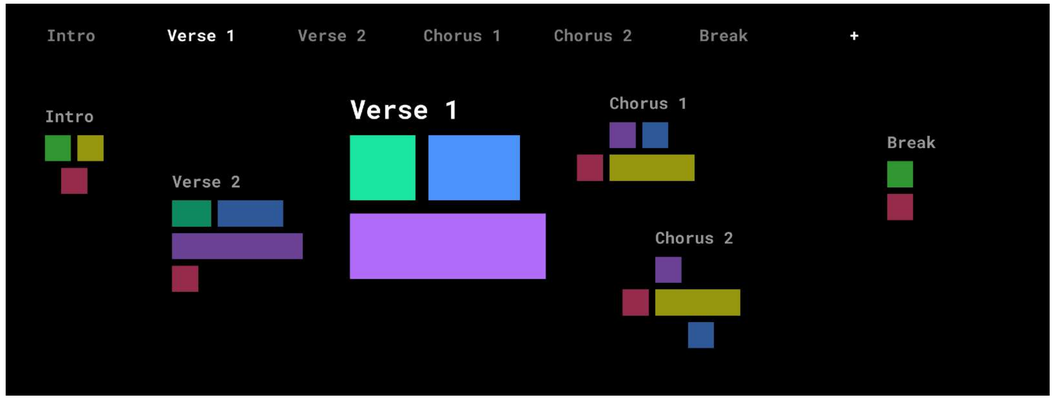


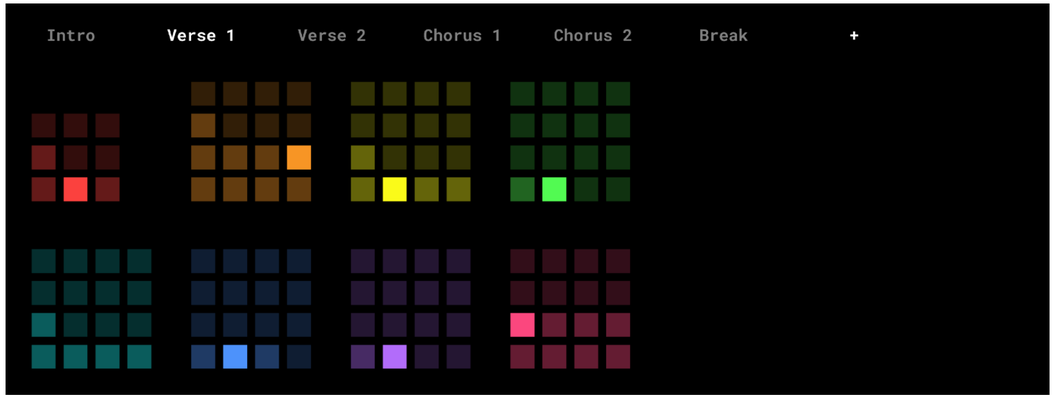


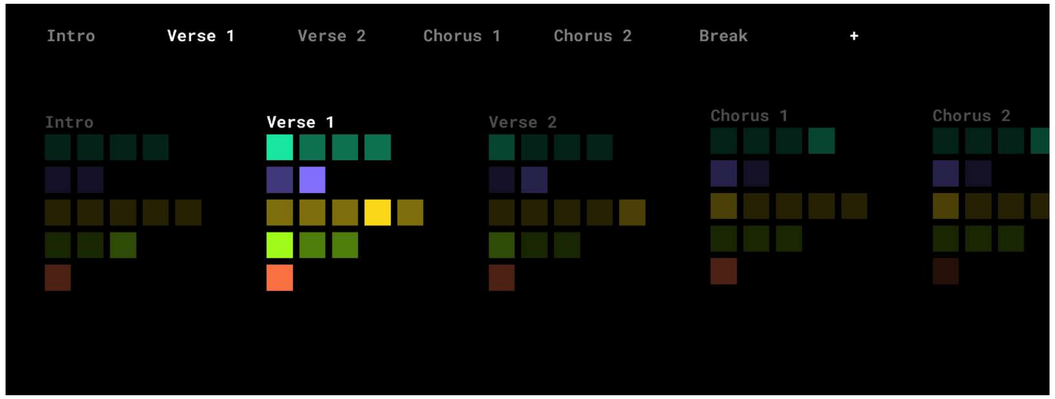


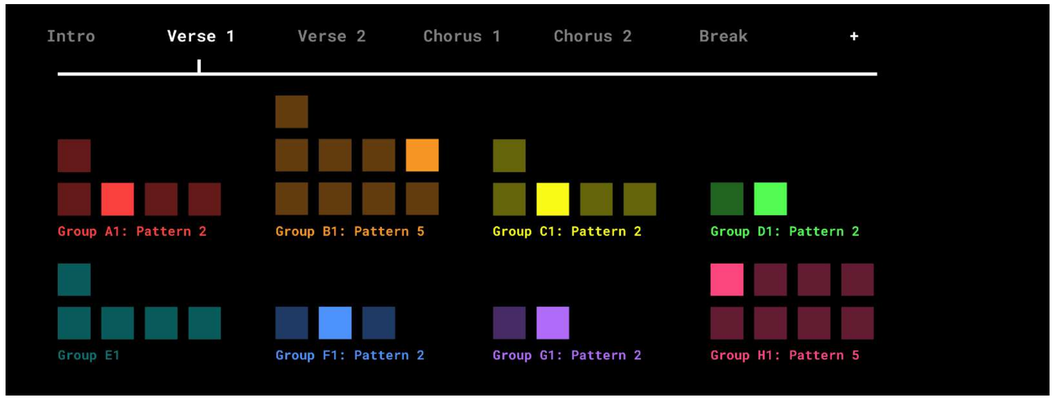
UI exploration







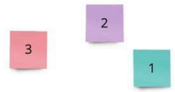
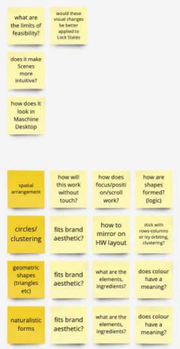
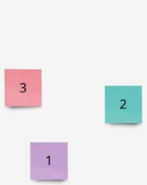




Outcomes scoped, prioritised by team confidence and effort

Ideas View Design Sprint

Outcomes

	User benefit	Next steps / action items	Effort (h/d/w) (time needed for one person to build a detailed pitch or testable prototype)	Confidence 1-3 "I think this will be a significant improvement to the product"
Small improvements we can test now			Adrien Ant Marcus	Adrien Ant Marcus
Change the axes of the Ideas view? Groups are horizontal rows, Scenes are columns 			 [1.5d]	 [7]
Take Scenes *slightly* off the grid? 			 [1d]	 [6]
Significant improvements requiring deeper exploration				
Distinguish Scenes visually from Groups and Patterns: Take Scenes *further* off the grid? 			 [1.5w]	 [6]
Distinguish Scenes visually from Groups and Patterns: Take Scenes *more into* the grid? 			 [1w]	 [4]

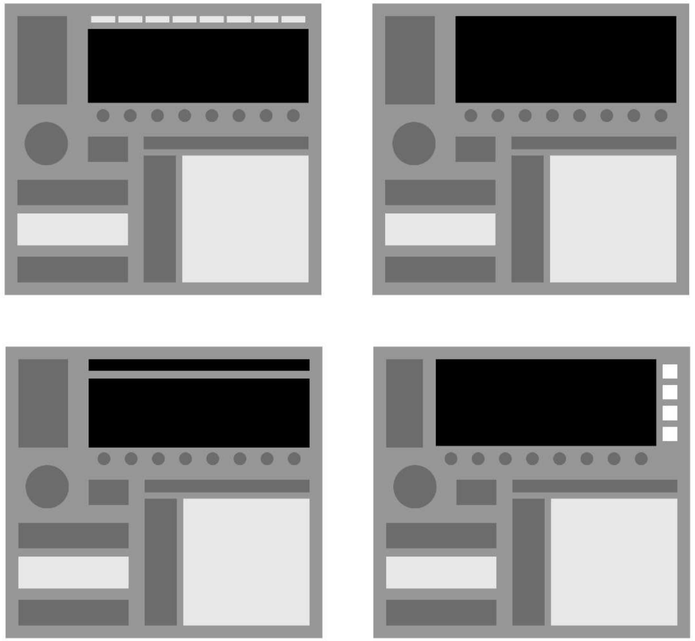
Let's try these quickly

3. Lean feature/workflow discovery: Context menu / OPTION button

With the Product team close to finalising a concept, committing to a new hardware layout with significant changes to core interactions, an engineer (and power user) strongly advocated for a change to the navigation and organisation of features which would drastically simplify the workflow architecture and improve discoverability.

The topic appeared abstract to people with less hands-on experience using the product. I took on the task of channeling the engineer’s advocacy, validating the perceived problem and shaping the proposed solution, aligning the potential value with the essential product story.

We turned disparate conversations into impactful, shareable artefacts, and convinced the wider team to commit to the feature.



One-page summary of problem and background

What?

The Context Menu is a third layer of access to functionality (after direct and SHIFT)

Conceptually, it's similar to a right-click on a computer, or a long-press on an iPhone: it gives me access to functionality which isn't directly accessible from the main interface.

Background

Maschine has two layers of functionality: the main/top layer "direct access", and the Shift layer.

Generally, each hardware button corresponds to...

- a *view* (Ideas, Settings, Scene, etc)
- a *mode* (Pitch, Perform, Keyboard, Step, etc)
- or a *global action* (Play, Rec, etc)

For every View and Mode there are specific actions or settings. These are accessible on the eight soft buttons above the screen.

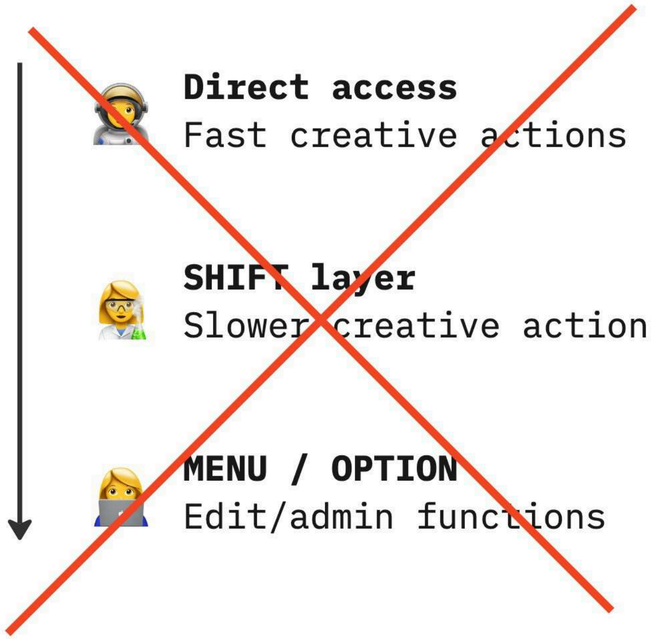
Why introduce a Context Menu?

1. Some hardware buttons are only used for global settings (eg. Swing). If only set or adjusted once or twice per project, they don't require direct access - they'd more logically live under project settings. These **hardware buttons could be removed or assigned to more creative functions**.

2. Some views could logically be merged into other views (eg. Channel: Input/Output into Mixer view, and Groove into Plugin view). This would **simplify the architecture** (less views) and **free up hardware buttons** for more creative functions.

3. We only have maximum 16 slots per View/Mode for contextual functionality.
   (ie. screen buttons: 8 direct access, 8 shift layer).
   Both creative/flow actions and more admin functions exist in these slots, and the placement is inconsistent. A context menu would be the default location for all non-creative/non-flow actions, opening up the **soft buttons for only creative/flow actions**, with consistent placement.

Aligning interaction design principles with workflow architecture

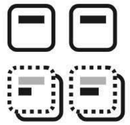


Direct access
Fast creative actions

SHIFT layer
Slower creative actions

MENU / OPTION
Edit/admin functions

Direct access controls
Core actions and workflows



SHIFT layer
Alternative button-specific functionality

OPTION
All other functionality, organised by View

Summary of options for interaction, hardware implications

How do I open the Context Menu?

Option 1:
Press a [MENU] button
(we add a new button)

where should the button be?

means heavy commitment to Context Menu feature.

muscle memory

Option 2:
Press a combination of (existing) buttons

SHIFT + Select?

saves us from heavy commitment to Context Menu feature

muscle memory

natural progression from Shift layer

should be close for one-hand operation

Option 3:
Touch the screen

Two/three-finger tap?
Swipe from edge/corner?

could be an additional way, as well as button(s)

relies on touchscreen

One-page summary of user story, commitments and unknowns

Proposal 2023-05-27

In a nutshell / as a user:

- I press SHIFT to access secondary functionality of hardware buttons and screen buttons.
- I press OPTION to access further functionality and settings relating to the current View.

Decision / commitments at this point:

- We keep the SHIFT layer and button.
- We call the context menu OPTION.
- The OPTION layer functionality is View-specific (not element specific).
- We **do** add a new hardware button for OPTION, but add a SHIFT function to a nearby button (propose: Select)

Known open questions / work to do / rabbit-holes:

- UI (at design-system level) for OPTION layer.
- Structure of OPTION list - vertical/horizontal, groups, nested, etc.
- Use/role of touch interactions for OPTION layer.
- How does current focus (eg. Scene, Pattern) respond to or influence OPTION-layer functionality.
- Make second-layer functionality more visible/discoverable in general, across the design system (concerns Kontrol also).
- Press-and-hold might also be a viable interaction to activate the OPTION layer.
- We still need to list the extended functionality and settings for each View.
- We are not yet sure exactly what the consequent structural changes will be.

Thanks!

Detail/case studies on request
→ ant @ ajo.design