



sili_auto

HMI design process

CES2020 demo case study

Hi there



Michał Jasiński
UX/UI/Visual designer



Mateusz Skoczylas
HMI UX/UI designer

Design process of creating CES2020 demo

What worked, what surprised us.

siili_auto

What will you learn?

- Design process
- Design challenges
- Issues

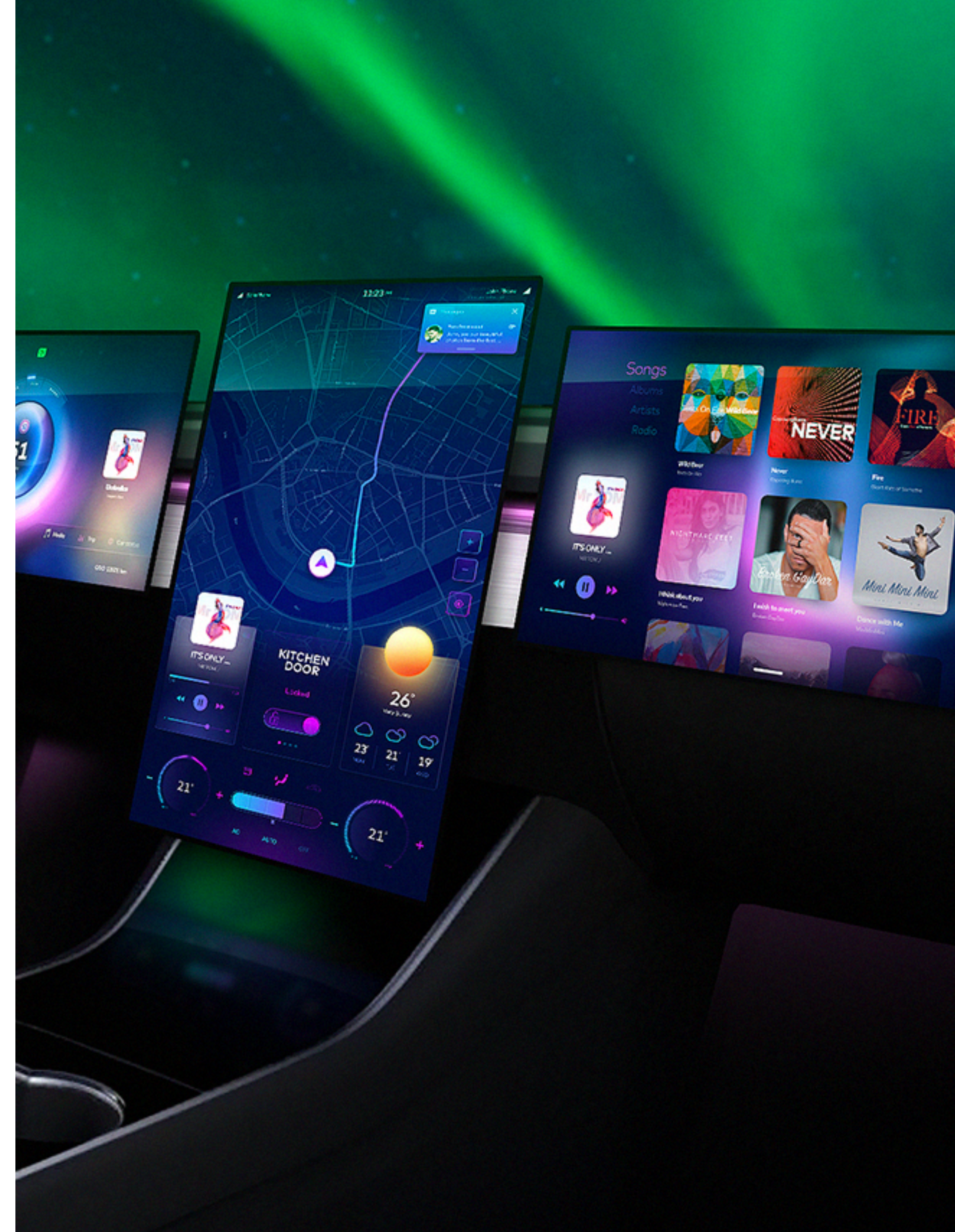


Background

- **We're at the end of 2019**
- **Bigger and bigger screens are coming to cars**
- **Several manufacturers are considering adding a 3rd passenger screen**
- **Android Automotive seems ready to be used in car manufacturing**

What is the goal of the project?

Let's show the world that Qt is ready to build a car interface consisting of three large screens and supporting Android Automotive.



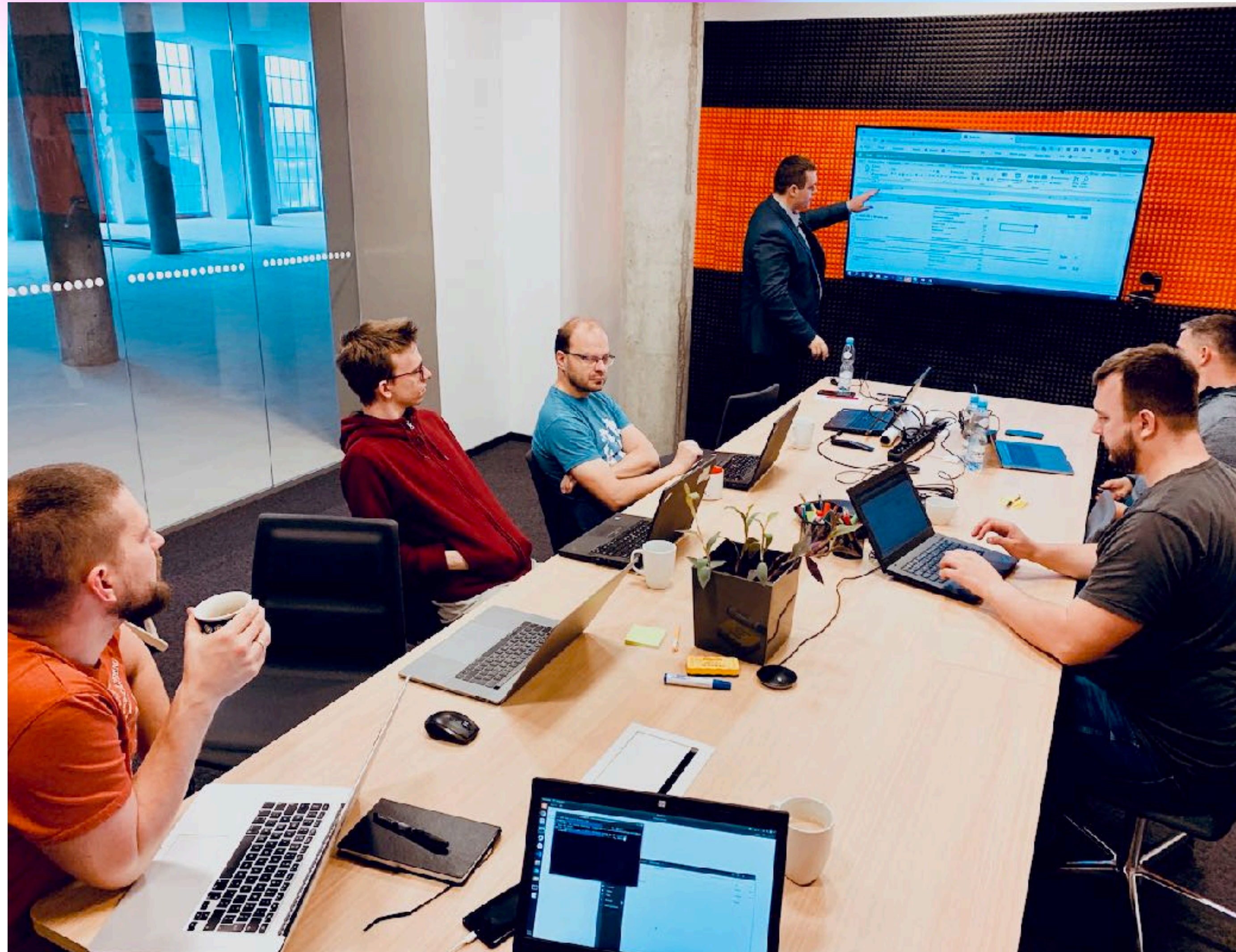
We need the plan

- What is our goal?
- What do we need to accomplish?
- Who do we need?
- How much time do we have?
- What do we need?
- How will we do it?
- ...



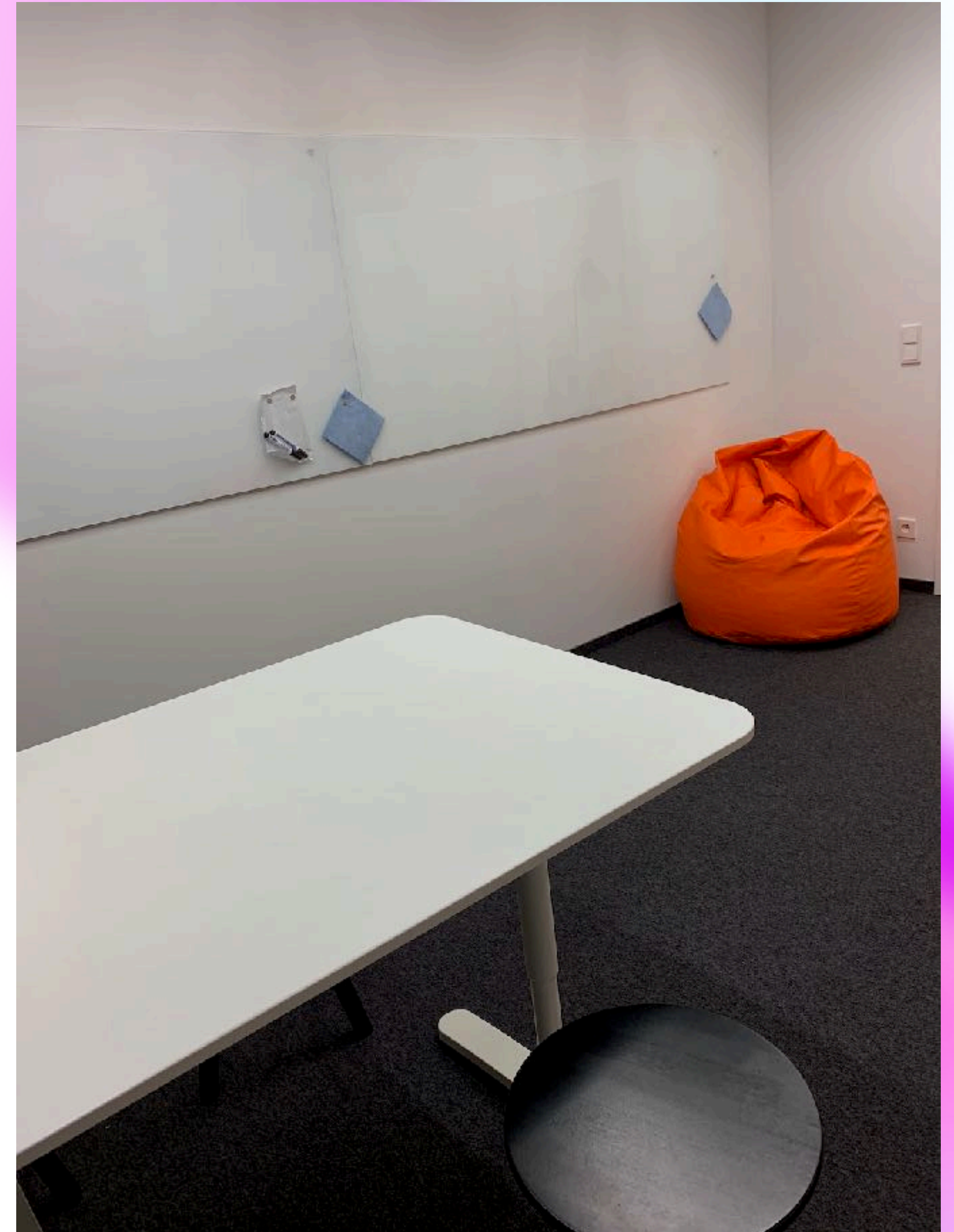
Avengers - The team

- **Antti Aaltonen** / UX Director from Qt (product owner)
- **Marcin Ostrowski** / PM
- **Mateusz Skoczylas** / UX/UI designer
- **Michał Jasiński** / UX/UI designer
- **Oskar Lewandowski** / Tech leader
- **Tomasz Jankowski, Sebastian Mateja, Aleksander Grobicki** / developers



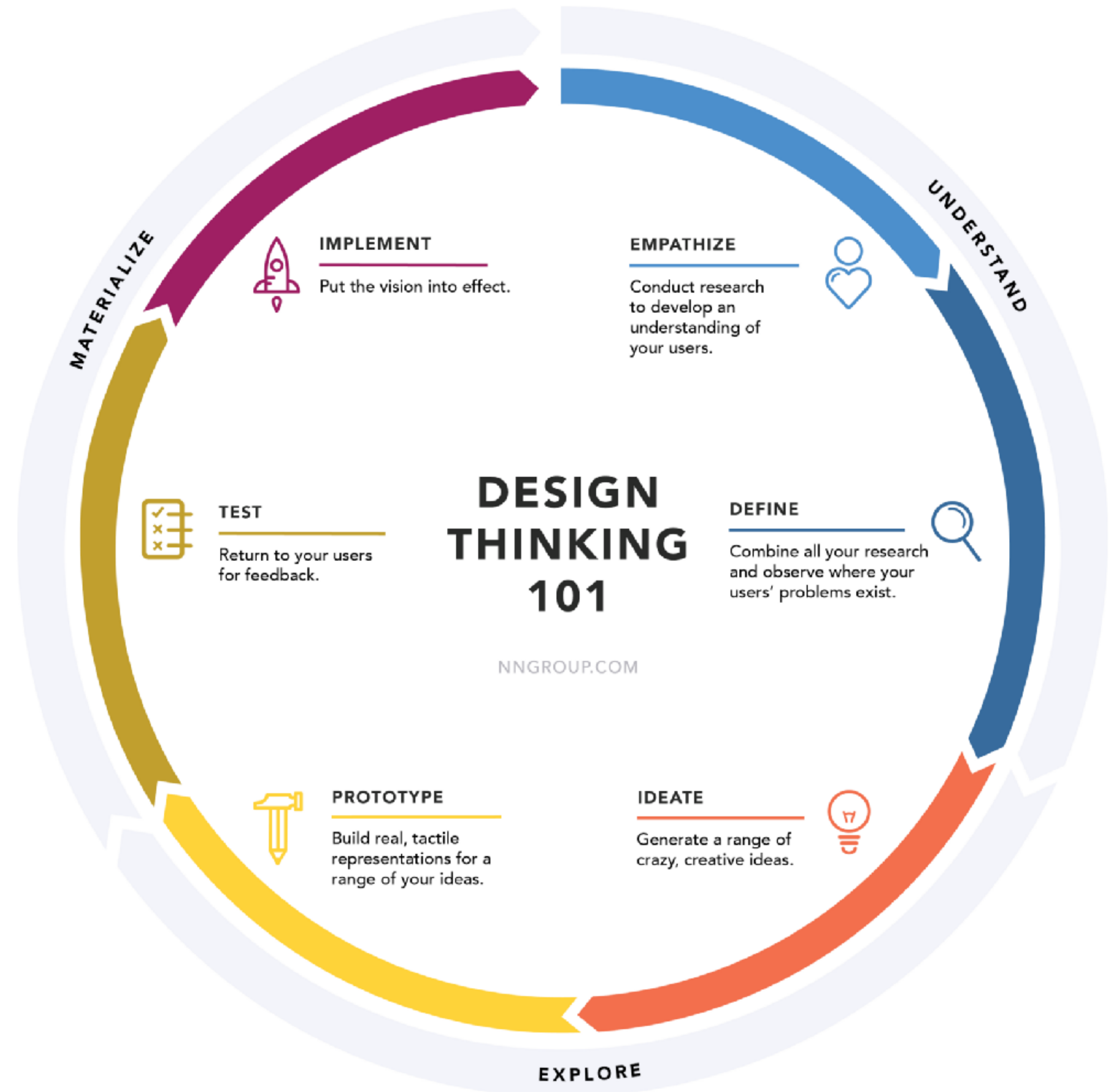
The workshop

Once we built the team, the questions that were asked had to be answered.



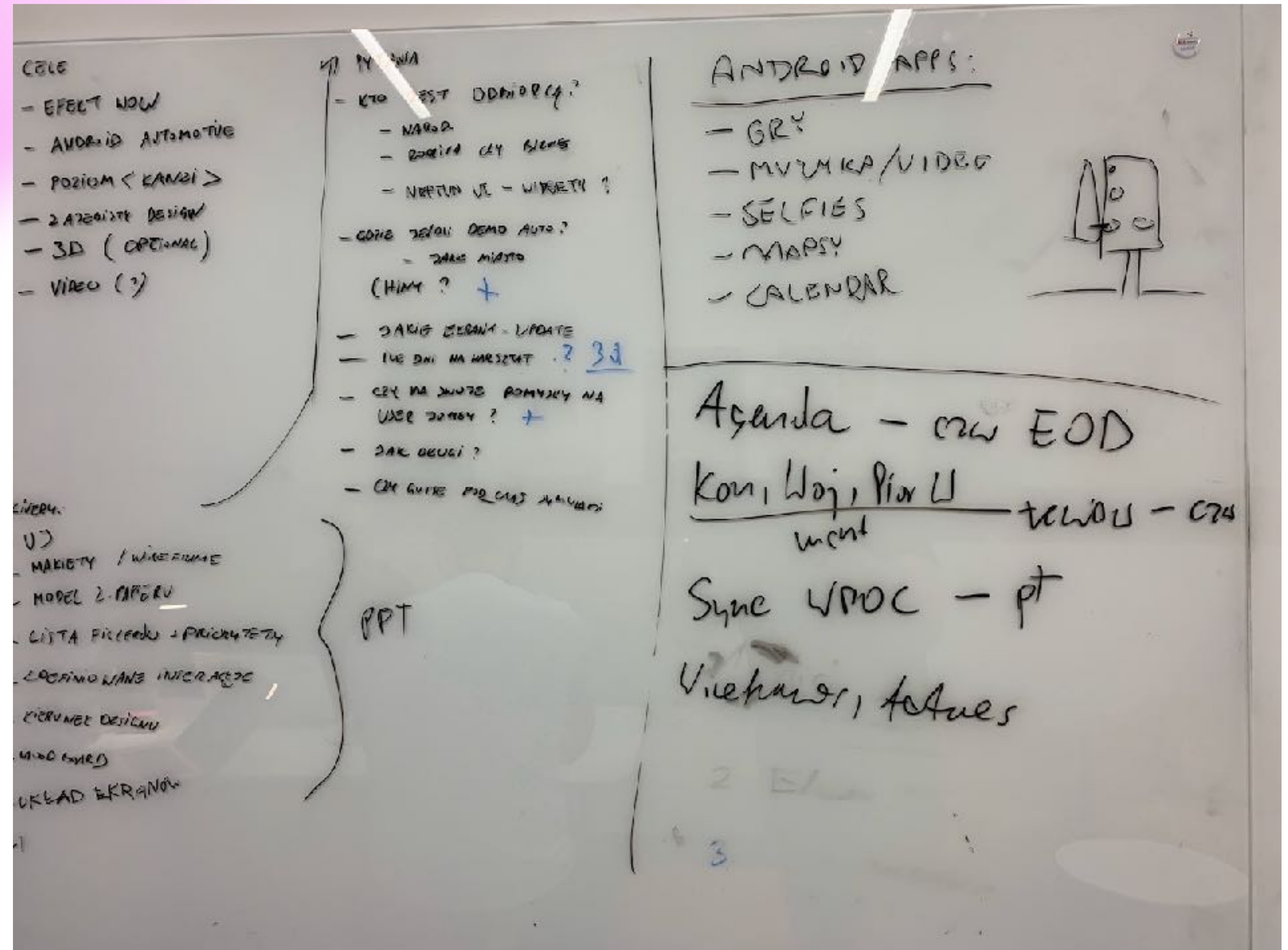
Design process

Design Thinking method.



What is important to us?

Determine technical features we need to present.



The main demo character - persona

..., no - two personas.



"Mary"



Behaviors

- Has a housecleaner
- Buys take-away 3 nights/wk
- Frequently feels overwhelmed when she "forgets" something

Demographics

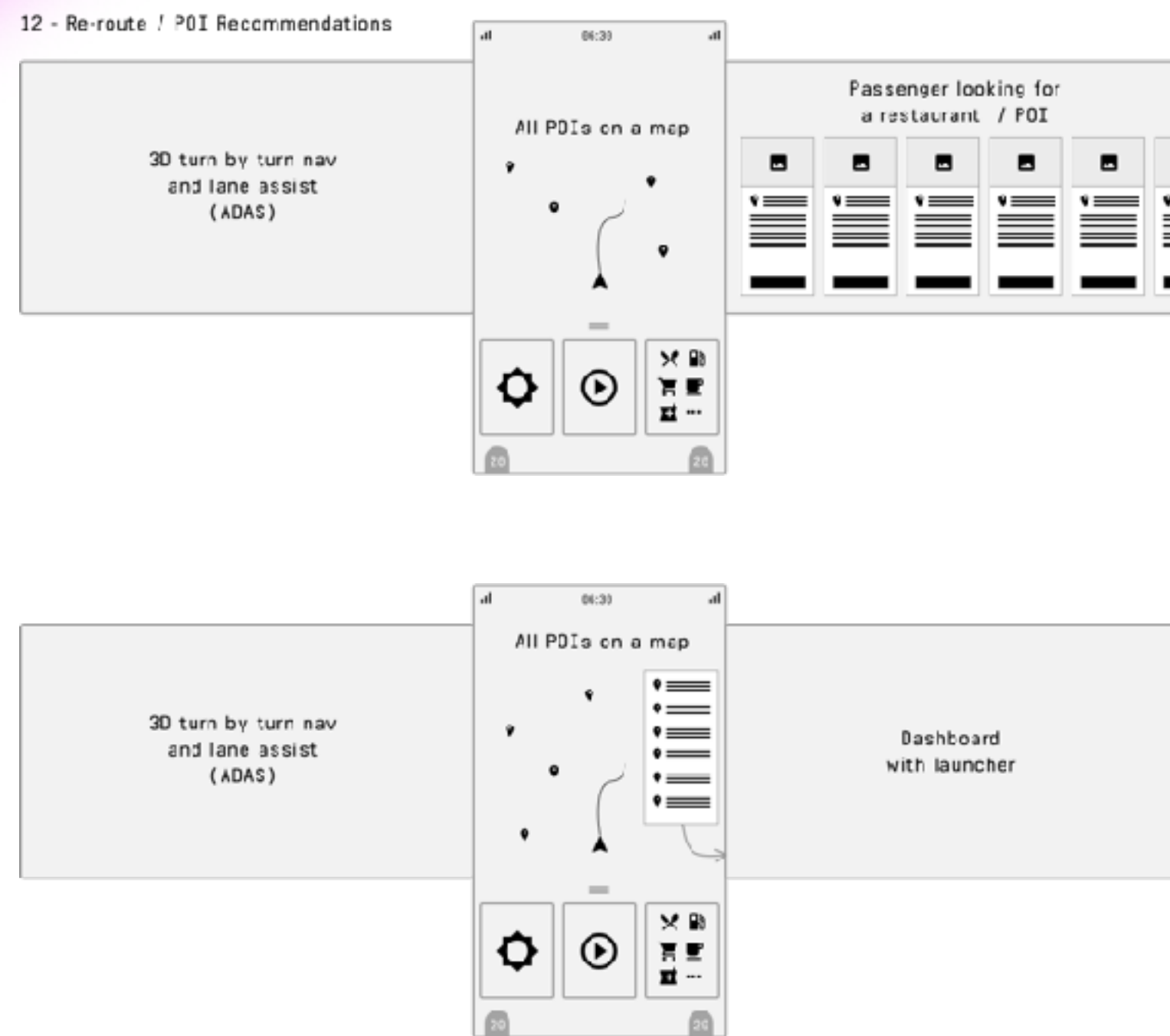
- Working mom
- 34 years old
- Lives in Reading, works in London
- Married, 2 kids
- Household 125k/yr

Needs & Goals

- Help! Running errands, managing kids, keeping things running
- Time for her girlfriends
- To feel like she "has it sorted"
- "To clone herself"

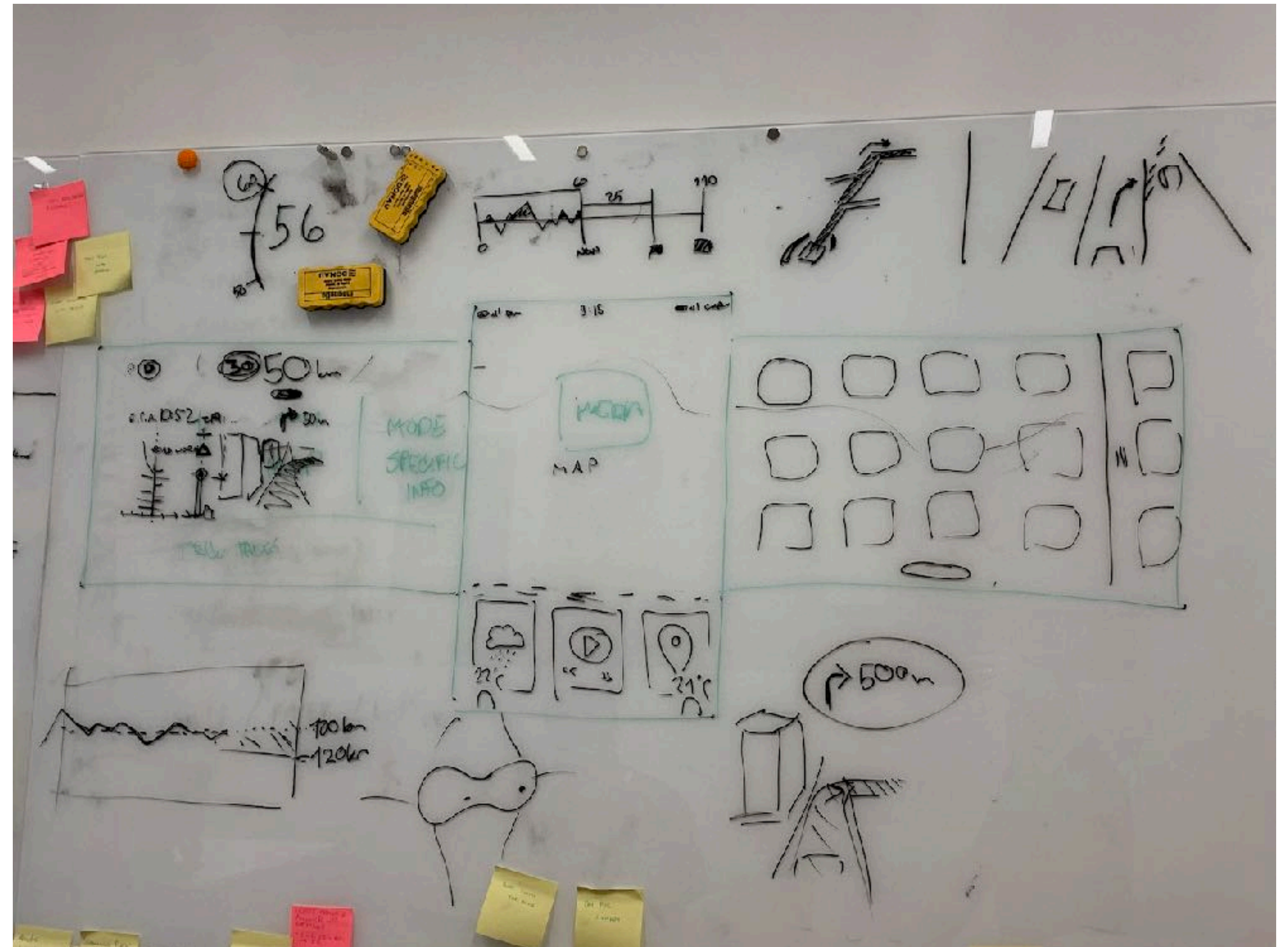
The story(board)

Then we created drawings - our story board.



Mapping features

Building the backlog.



What limitations do we have?

- We had a strict deadline.
- We could not count on a larger team.
- We want to present all important features.



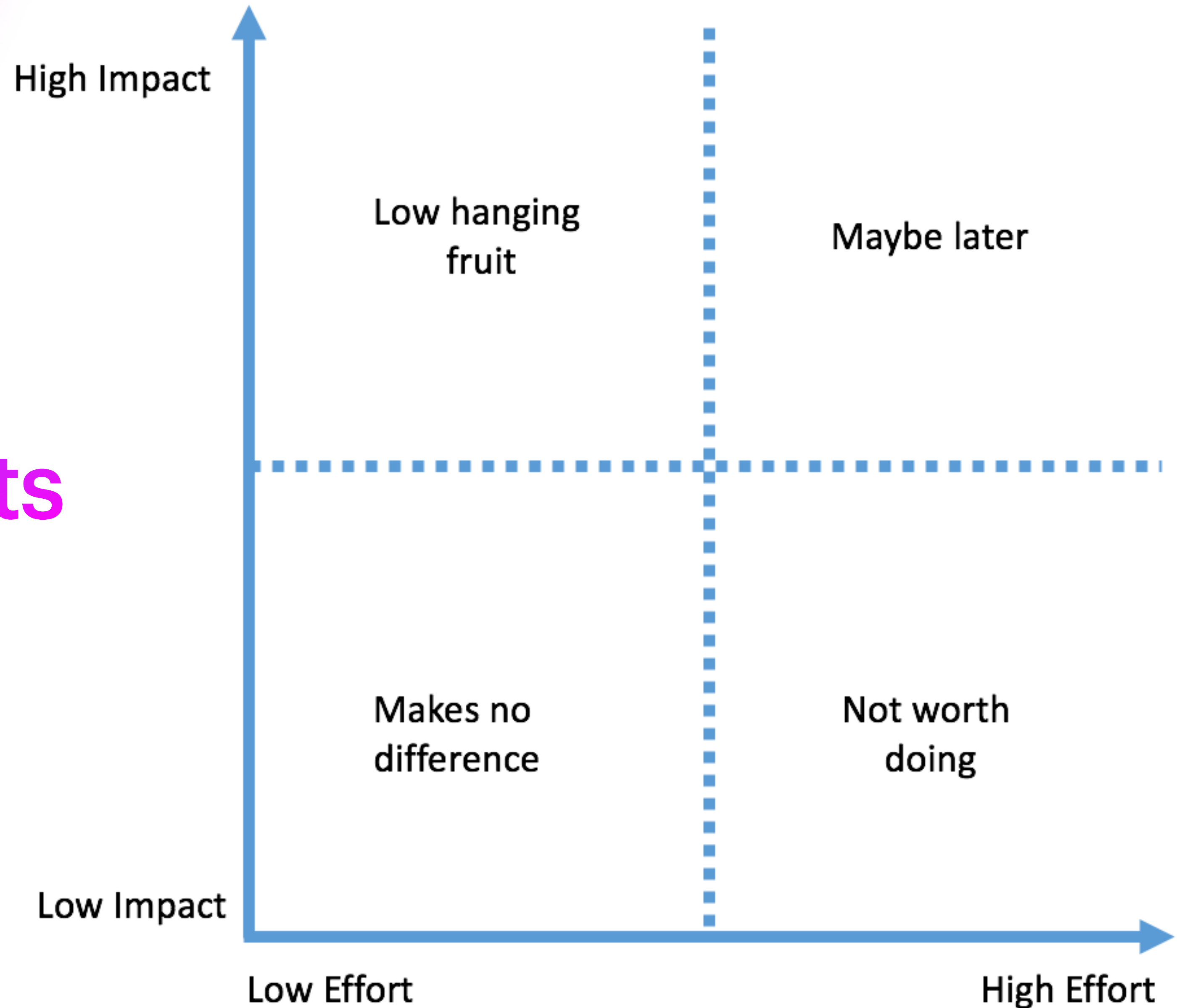
So what can we realistically do?

How do we choose what we must, what we can, and what is worth doing?



Looking for low-hanging fruits

Impact Effort Matrix



Consultation with developers - early estimates and rapid prototyping

**Talk, ask, create rapid
prototypes and test.**

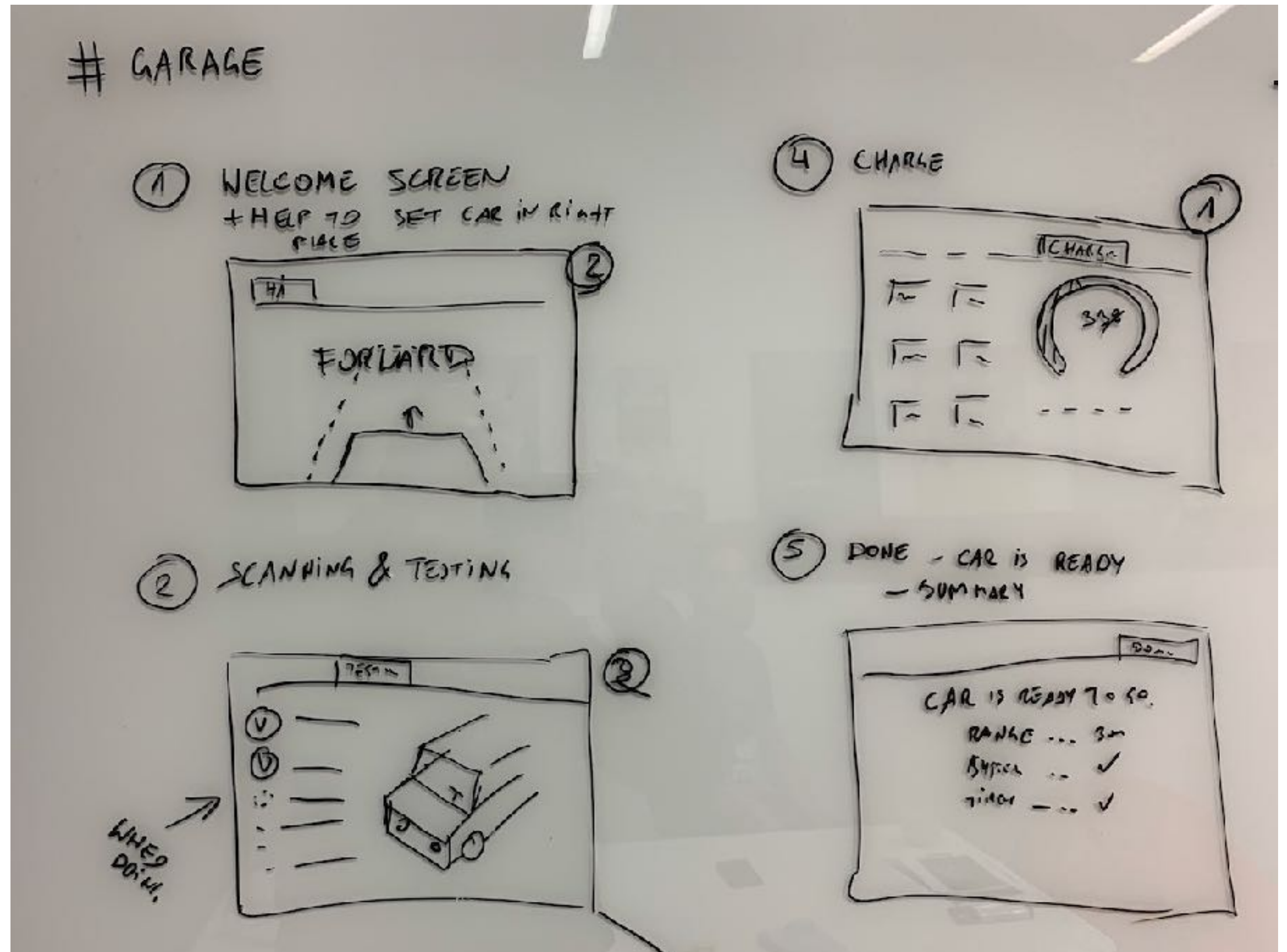
Summary and get to work

Learn to work under the press of time. Use it as an advantage.



Wireframes, drawings, prototypes

We drew screens, superimposed them on a cardboard model, and tested our hypotheses.



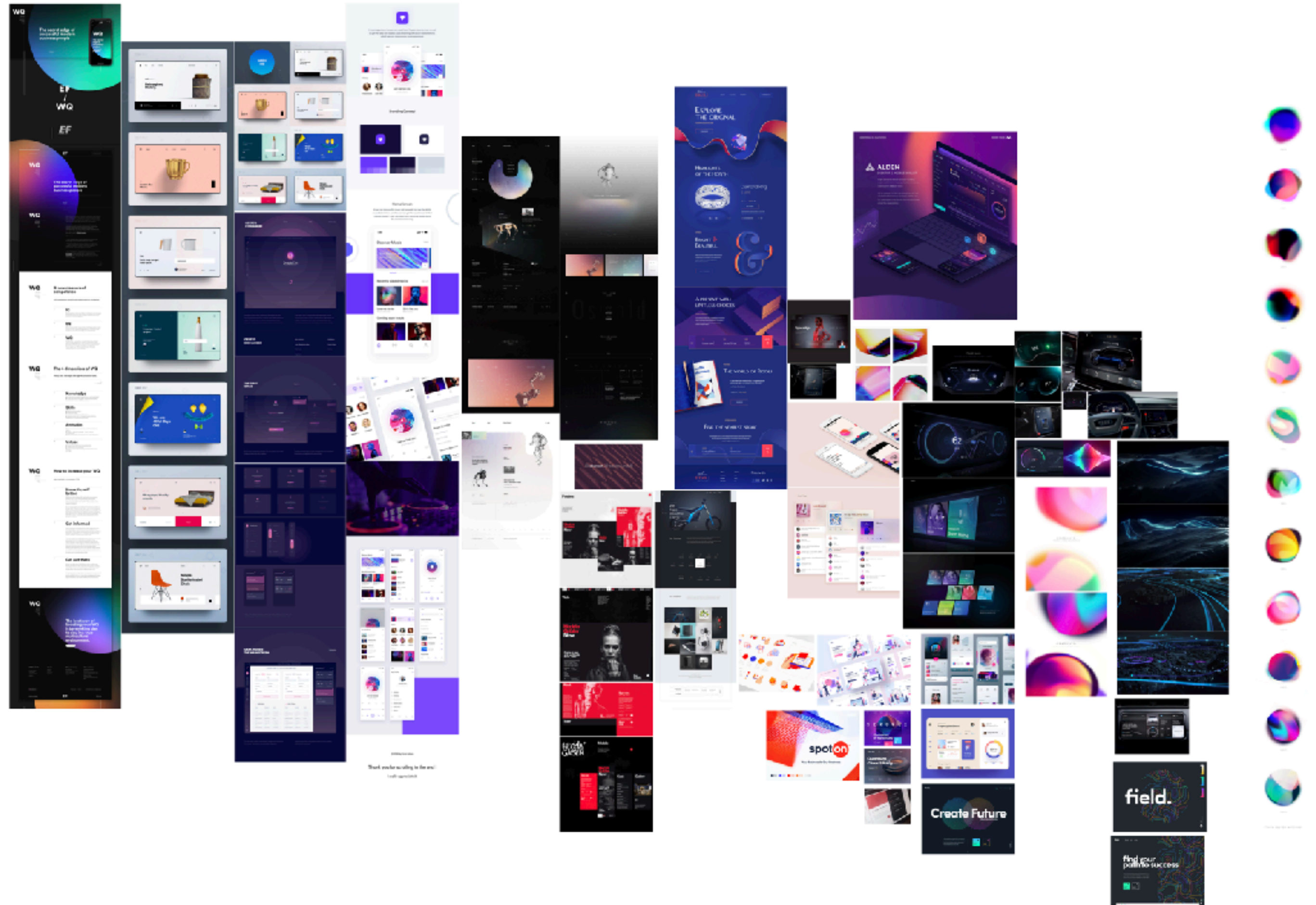
The end of the workshop, planning

Summarize all the
knowledge you have
gathered and plan
further work.



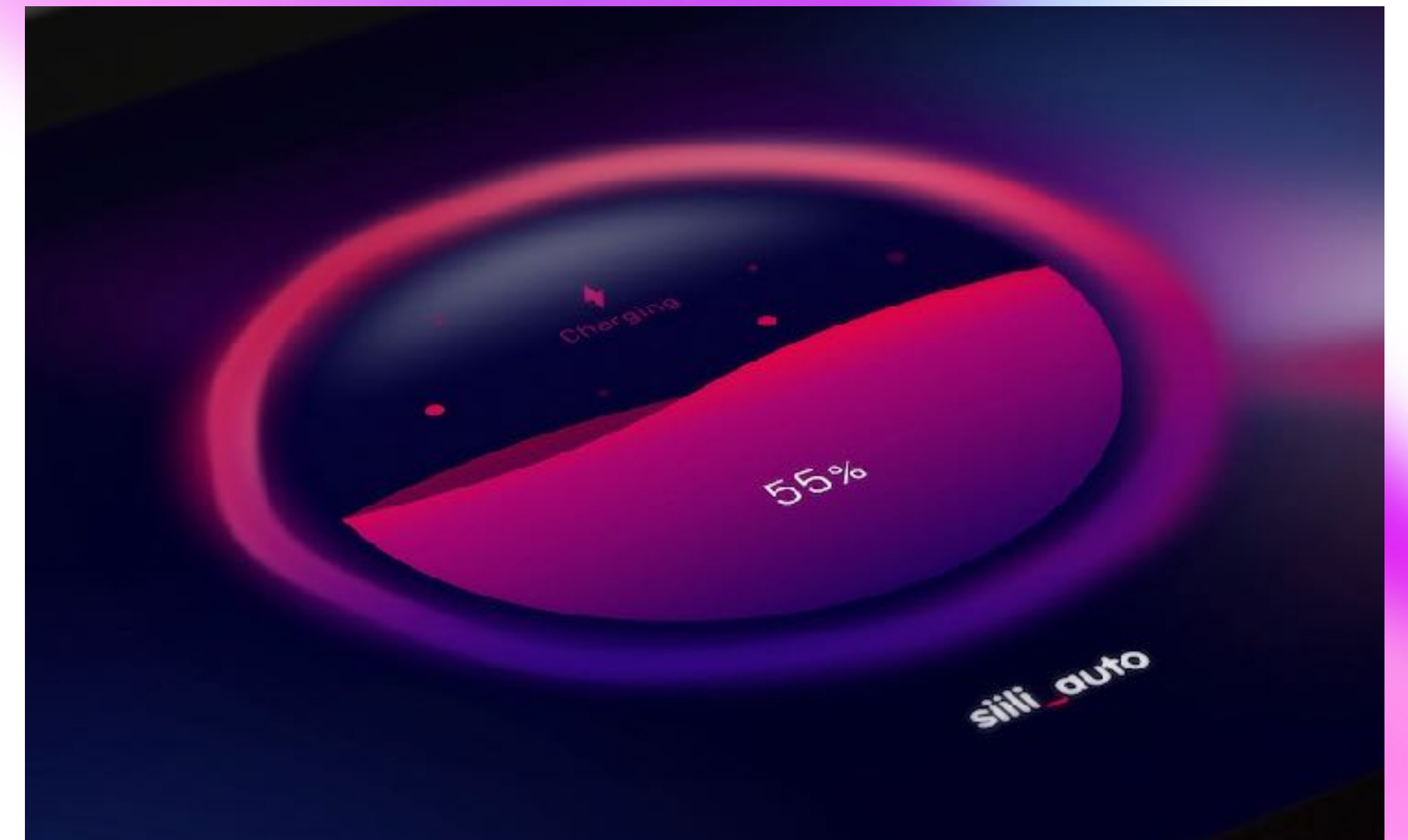
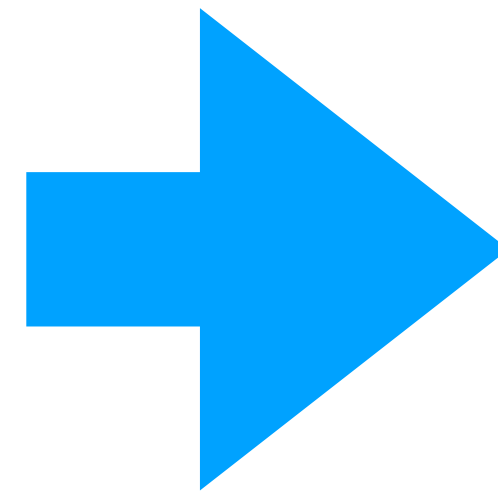
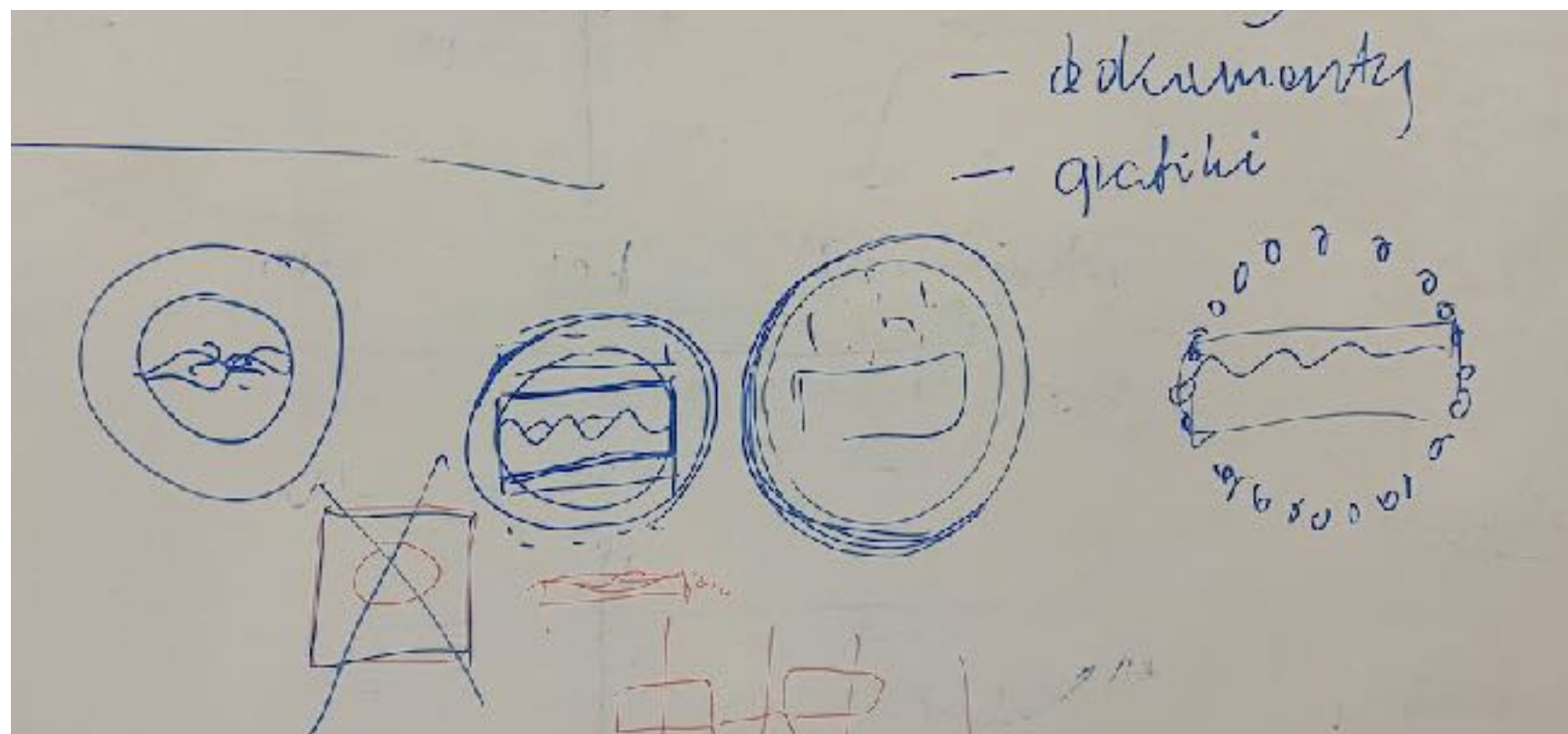
Graphic design, tools

Define the visual direction and start design work using favorite tool.



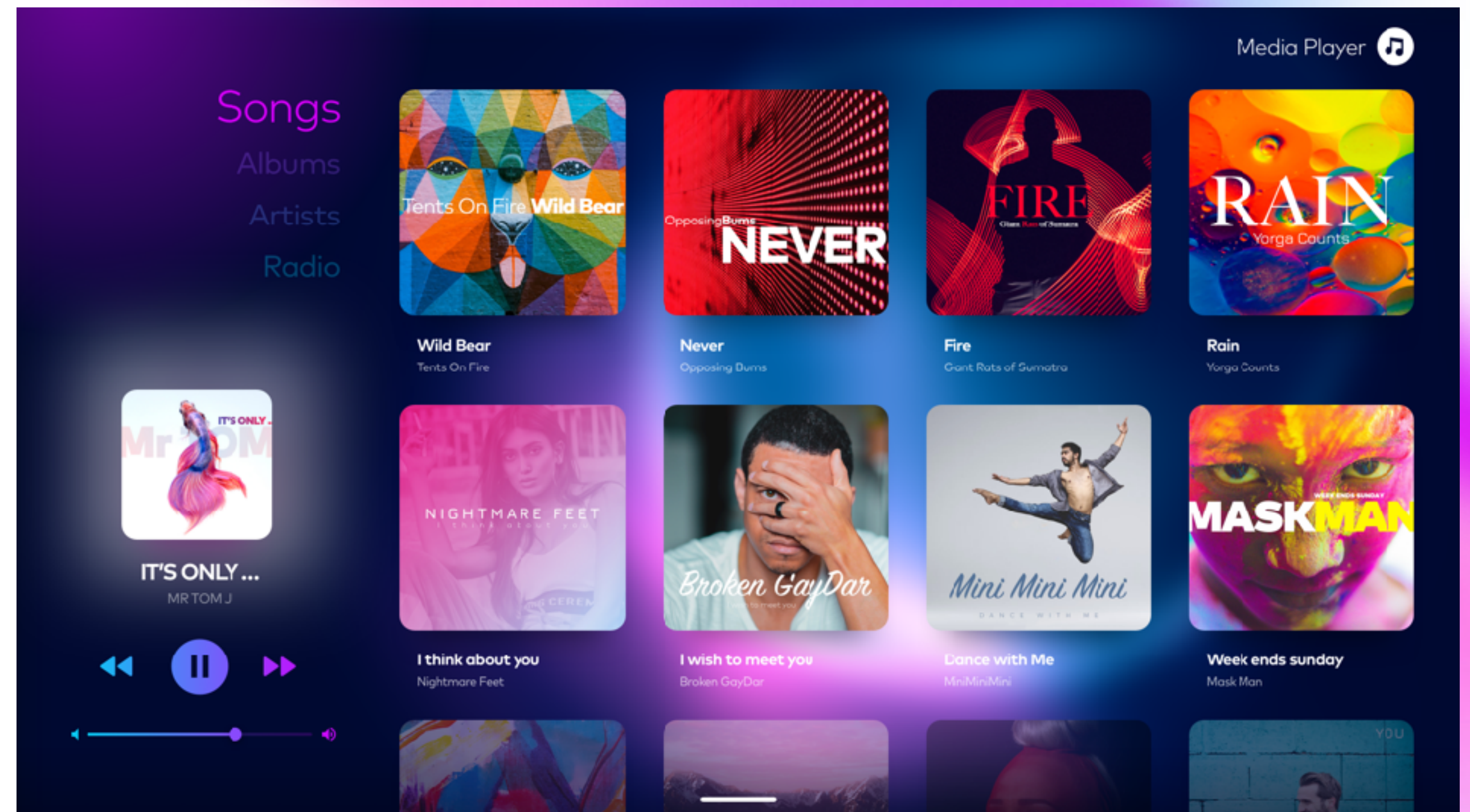
Iterative analysis of ideas

Rarely is the first version the best. Iterate.



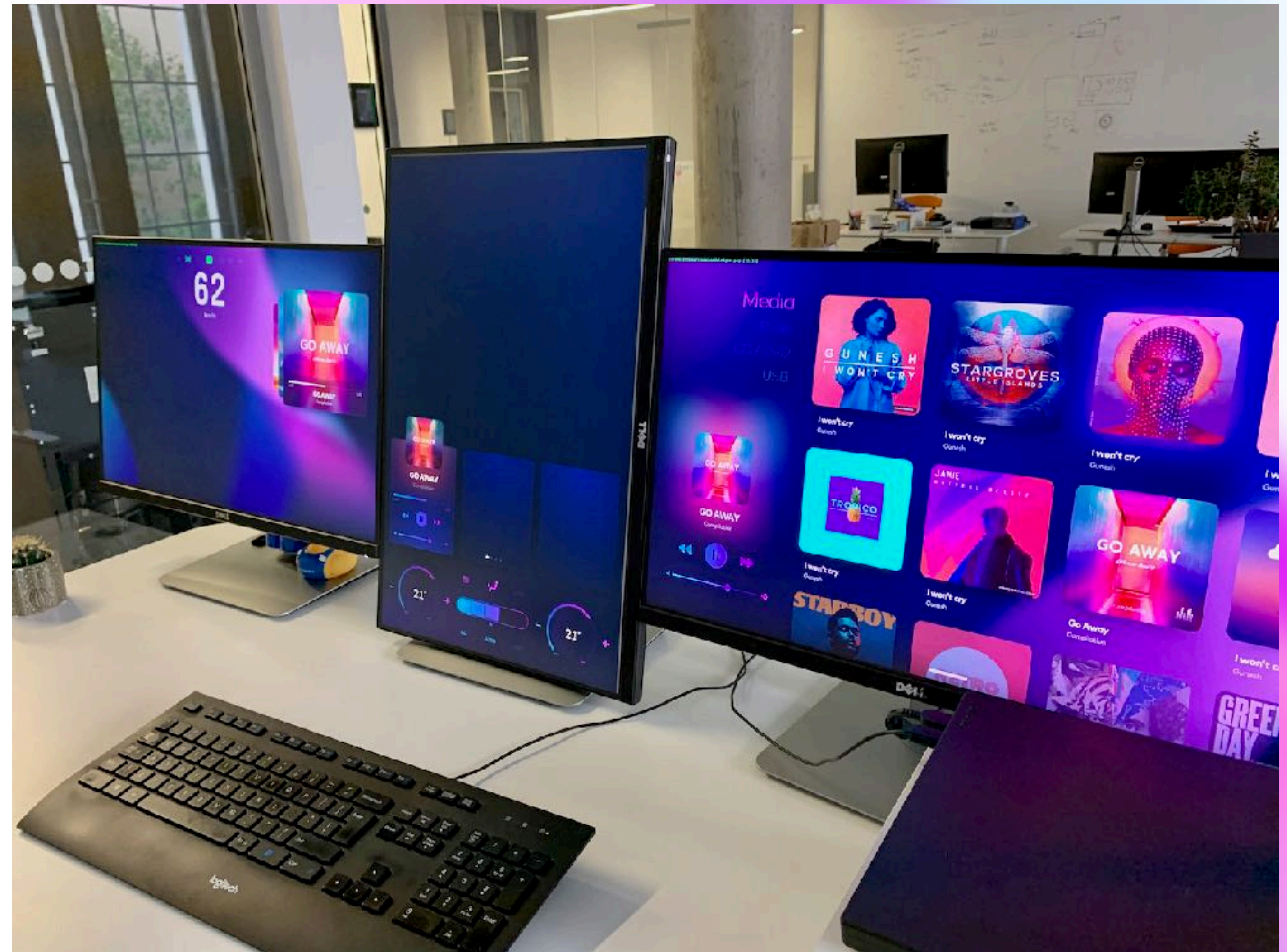
Less time, more problems

Look for simple solutions. If you make a mistake, you have time to make a new attempt.



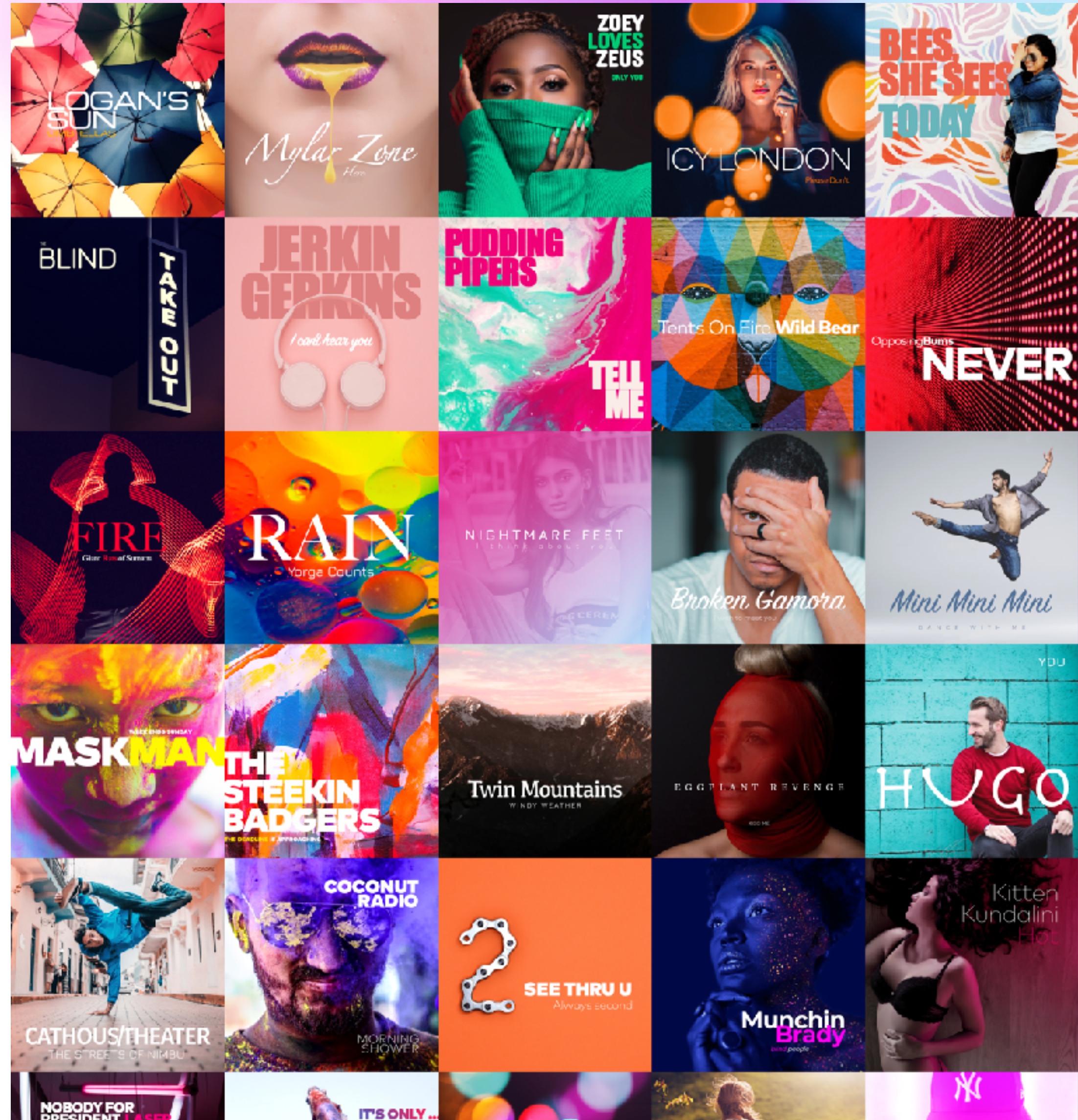
More and more problems

Don't be afraid to change your initial assumptions. They often improve the final result.



You say problem, I say challenge

Constraints and problems stimulated our creativity. It is worth leaving the comfort zone.

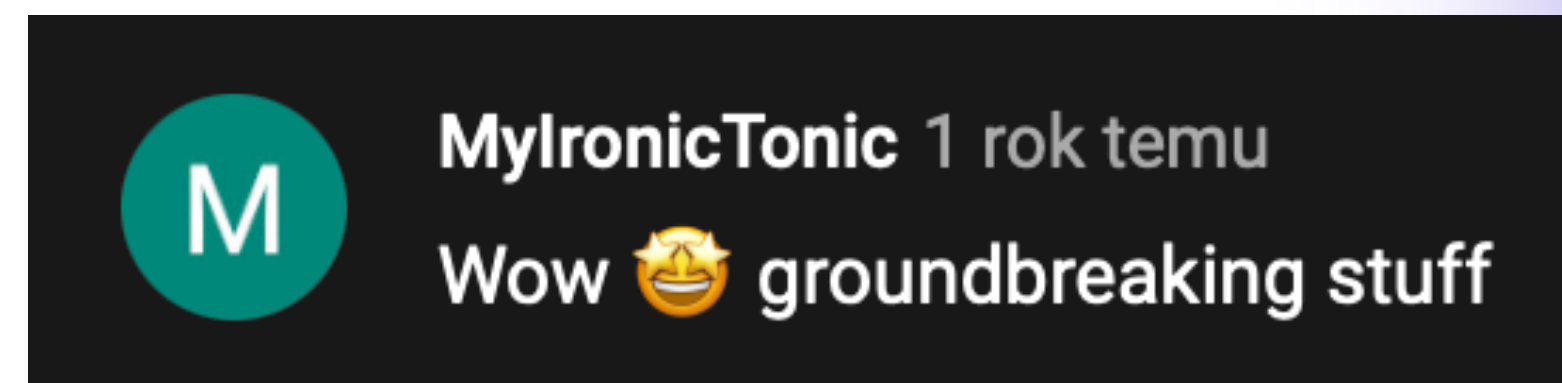


How do we know if it was successful?

We completed the project on time. Was it successful?
Of course.

How do we know that?
We asked.

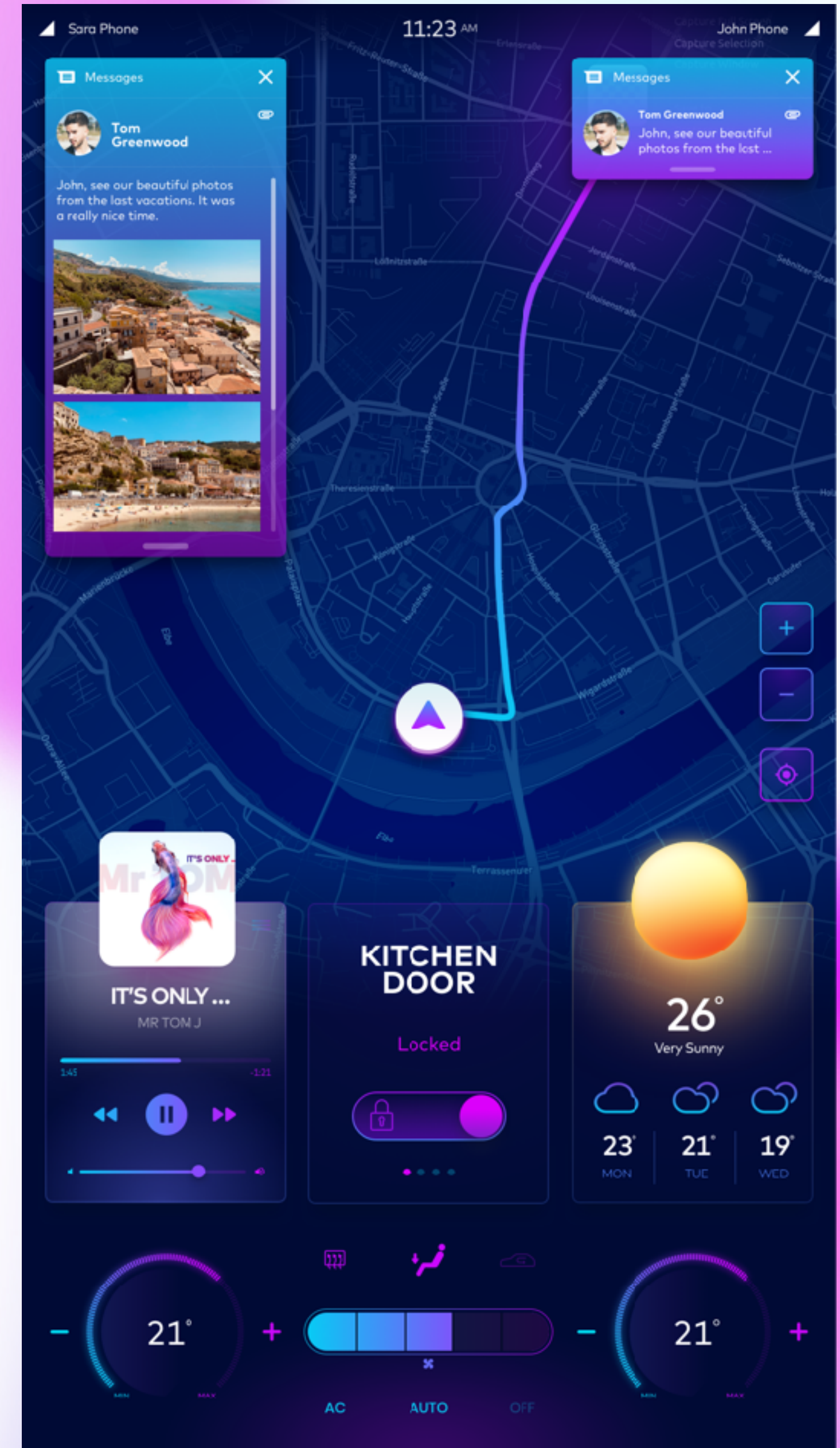
... but not only ;)



What's next?

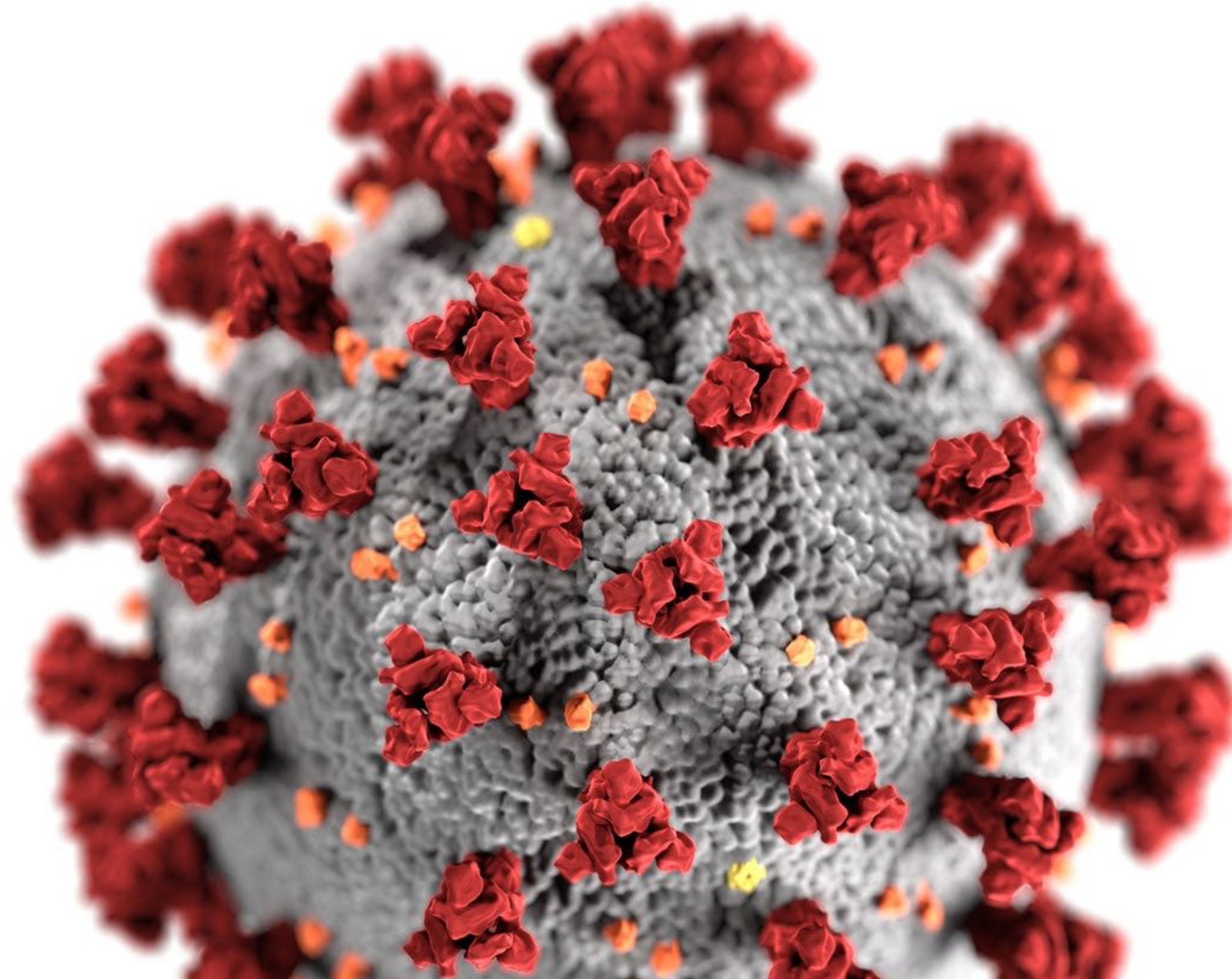
Excited by the project and the success, we immediately sat down to work on expanding the demo with more features. After all, we had a full backlog.

A new adventure was beginning, but ...



The year 2020, all in white

... COVID.



siili_auto

Lesson learned for the future

- **Agile collaboration**
- **Constant communication**
- **Information sharing**
- **Modify the process to the situation - adapt.**

Thank you

siili_auto



The Qt logo is displayed in a white speech bubble shape in the top left corner of the slide. The background of the slide is a dark, top-down photograph of a person's hands holding a smartphone, with a green folder, sunglasses, and a coffee cup on a wooden table. A large green geometric shape is overlaid on the right side of the image.

Qt

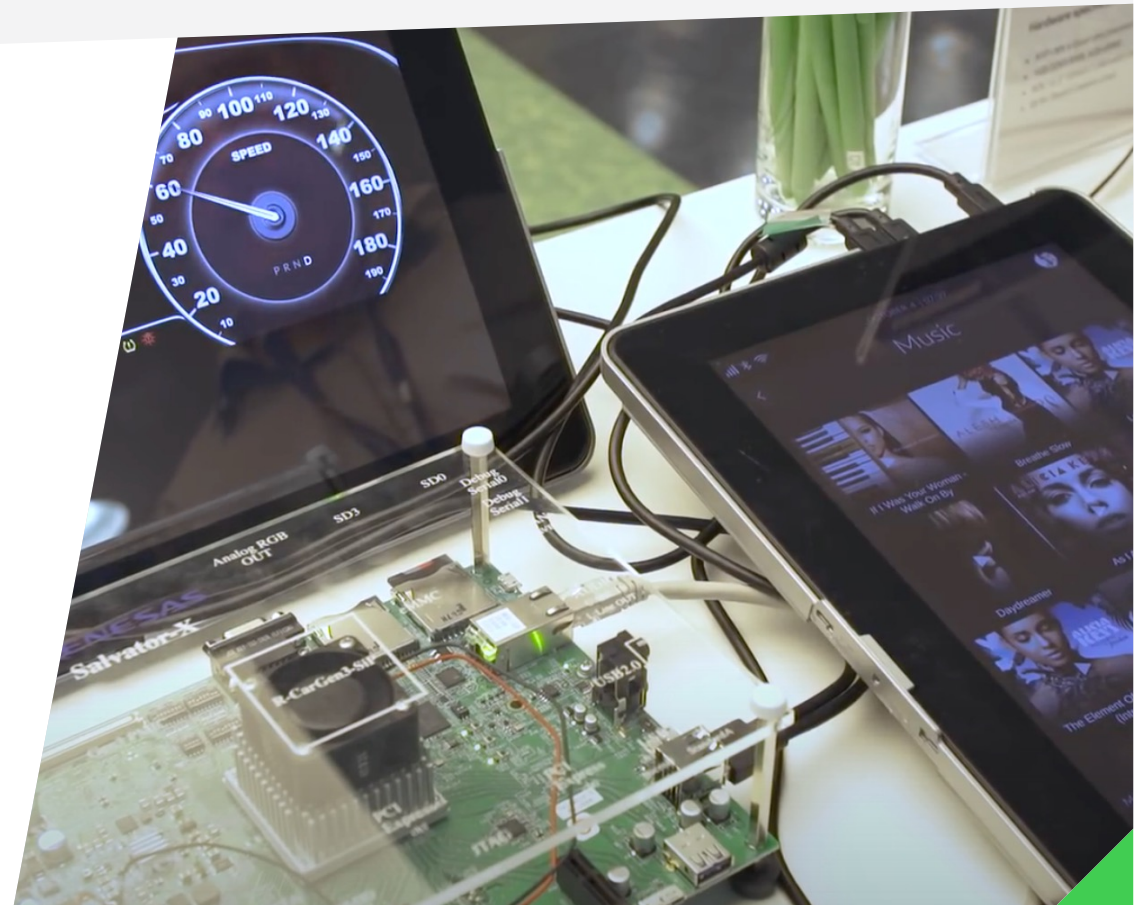
Companion App Design with Qt

Tino Pyssysalo
Product manager – Tooling / The Qt Company

Mar 30, 2021

Contents – Designer challenges

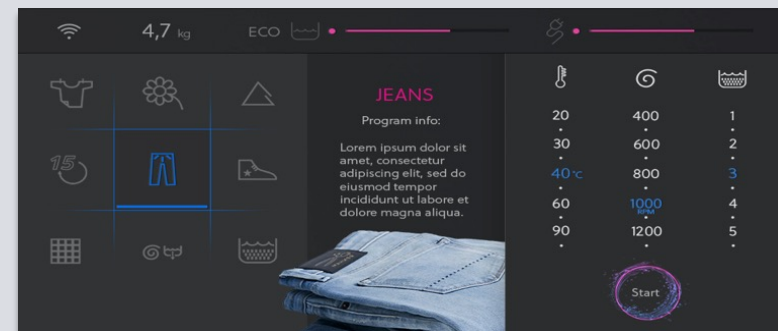
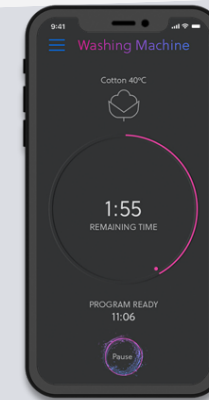
- 1 Design and implementation consistency
- 2 Switching between different tools
- 3 Optimizing repetitive tasks
- 4 Communicating advanced layouts and interactions
- 5 Re-using designs
- 6 Interaction design with minimal code
- 7 Concept creation



Companion Applications with Qt

- › Re-use existing design of the main product
- › Re-use the same assets
- › Apply the existing brand
- › Use Qt scalability features
 - › Do not drop UI features unnecessarily

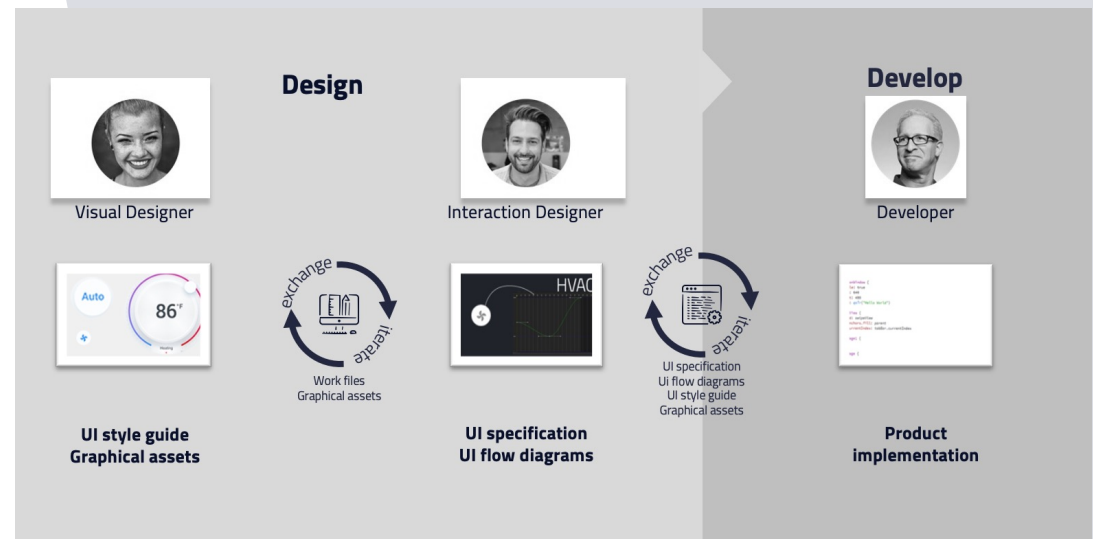
- › Utilise Qt cross-platform capability
 - › Use applications in desktop, embedded, mobile



Typical UI creation workflow

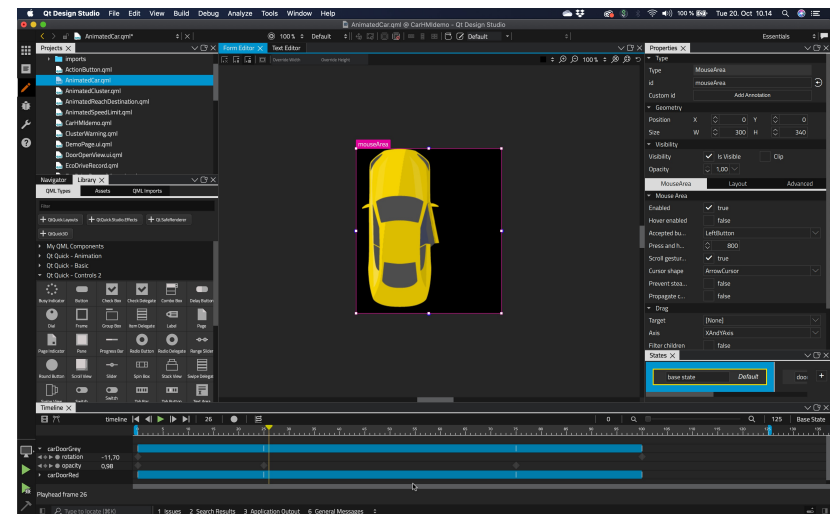
Design and implementation consistency

- › UI and interaction design results to a set of specifications
 - › AKA waste as the specifications are not directly part of the end product
- › Time-consuming to create, time-consuming to read and interpret
 - › Time-to-market risk
- › The design is not part of the end product
 - › 20% of features take 80% of the time
- › Qt way: **Design = UI implementation**



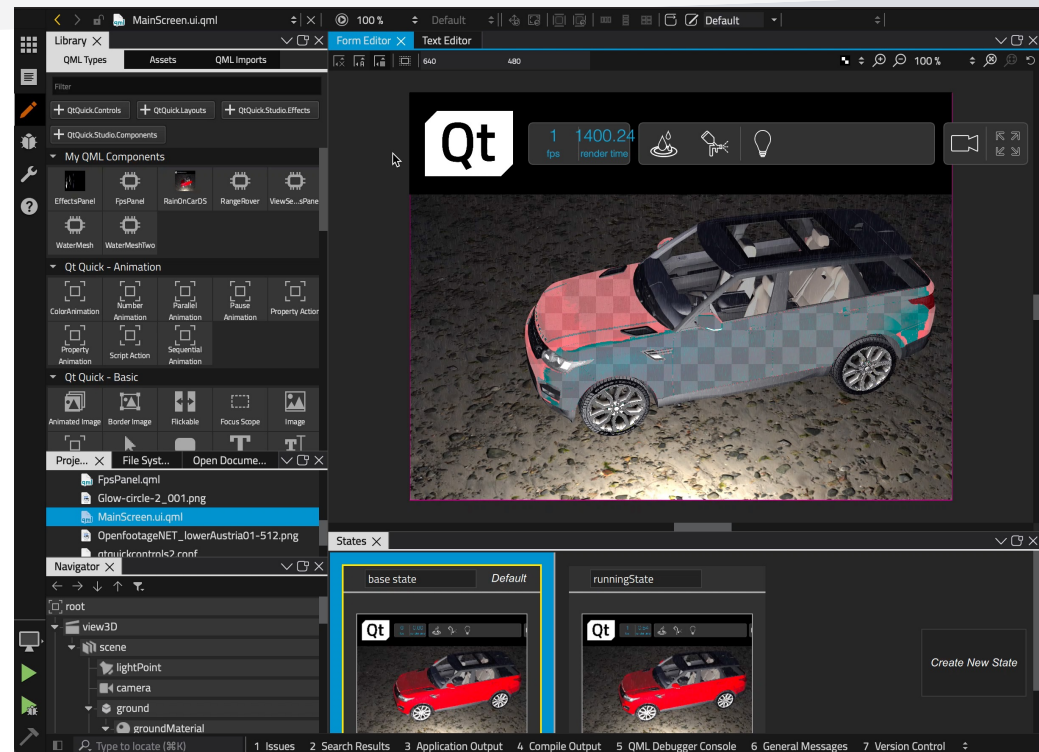
Qt Design Studio

- › Design beautiful experiences all the way from early wireframes to final pixel-perfect implementation
- › Rapid, iterative and incremental prototyping to validate the designs with target device
- › Bridging the gap between design and development with the unified toolchain



Qt: design = implementation

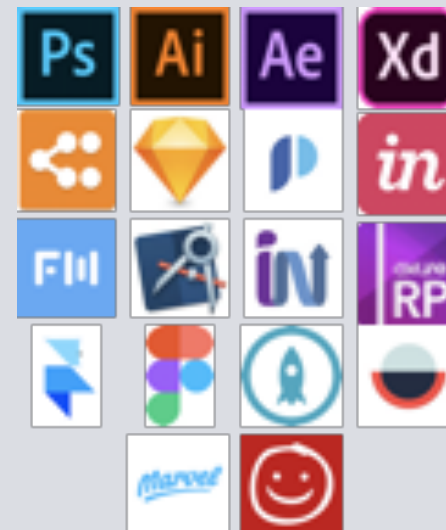
- › WYSISYG GUI editor
- › GUI converted to an implementation
 - › Directly usable by the backend developer
- › No changes in the implementation
- › Design is part of the end product => waste eliminated



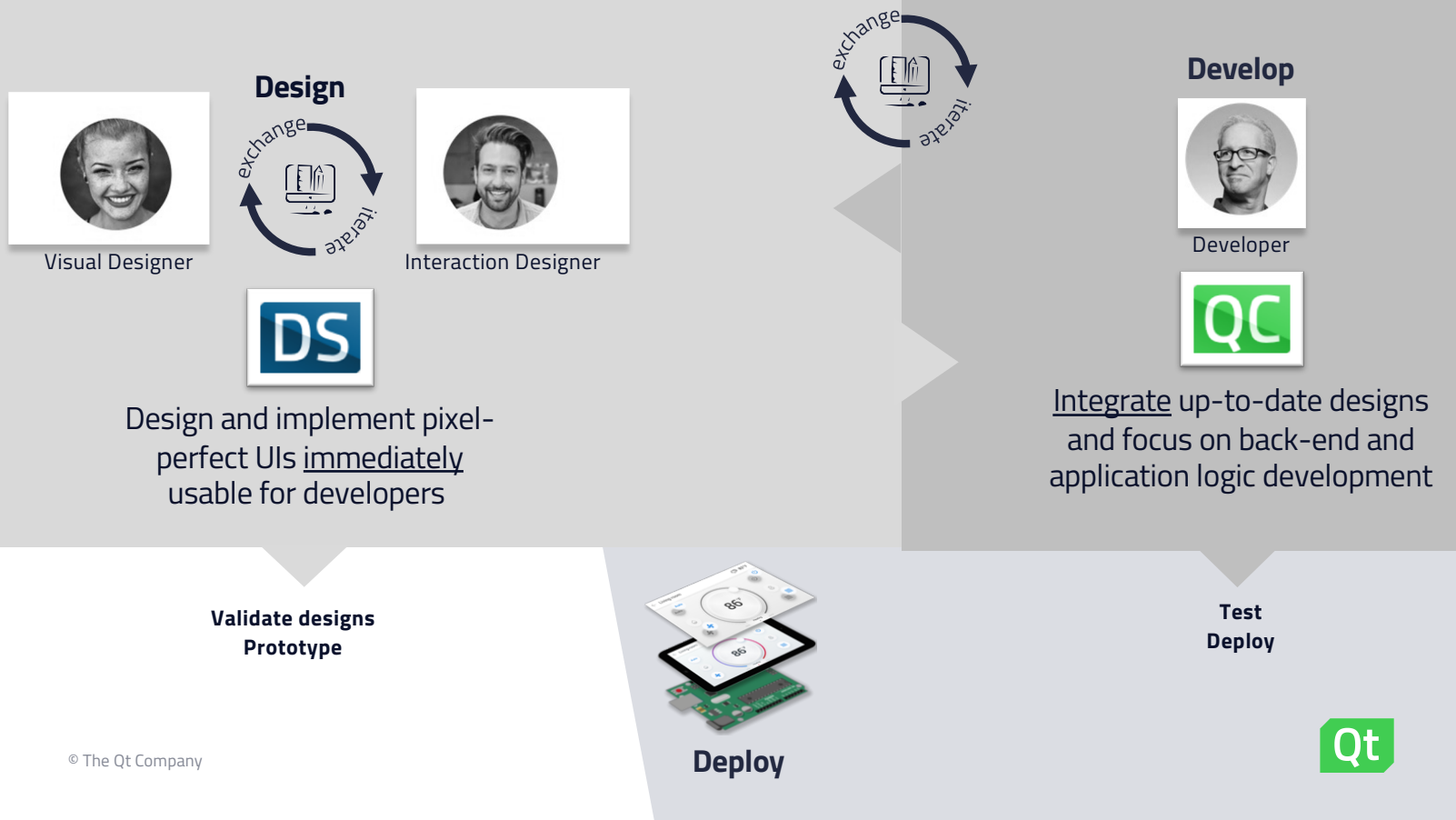
Switching between different tools

- › Tools for graphical asset creation
- › Tools for UI creation
- › Tools for creating animations
- › Tools for creating interactions

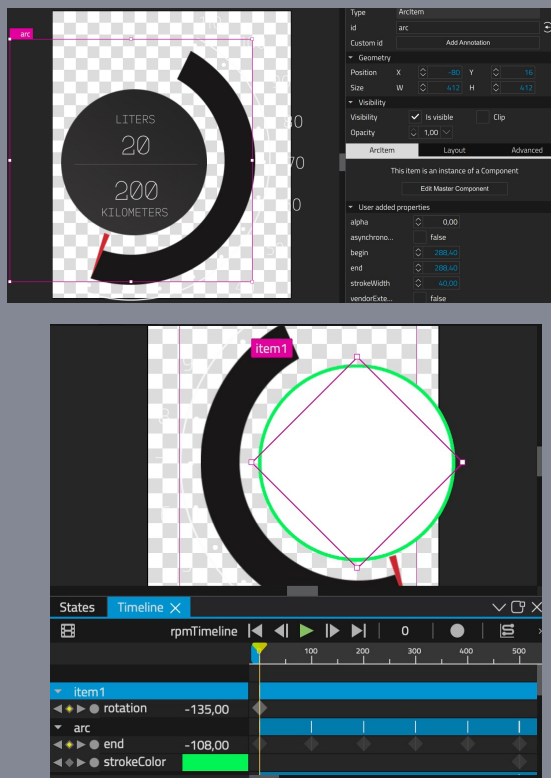
- › What is the best tool for each task?
- › Are the tools compatible?
- › Do the tools change the design?
- › No tool at the beginning of the project
 - › Total cost of ownership may increase



Qt approach – One design tool for everything



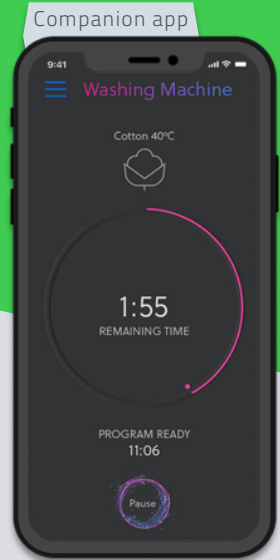
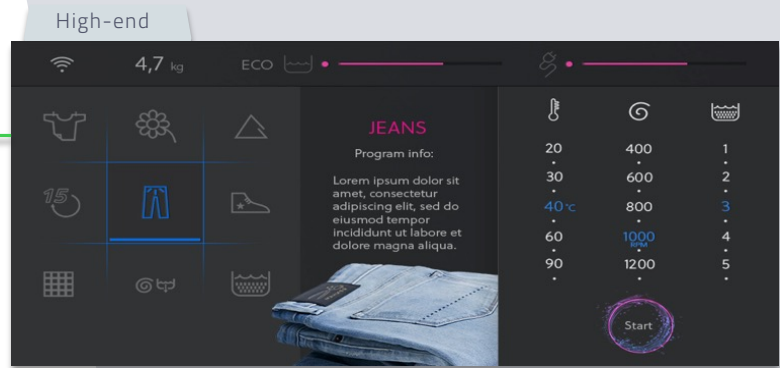
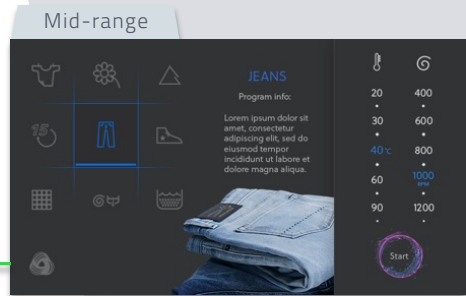
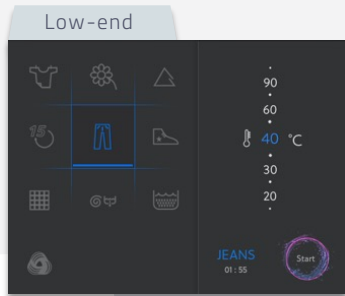
Communicating advanced layouts and interactions



- › Pixel-perfect layout can be set by just dragging and dropping UI items to the canvas
 - › Pixels do not change in the UI
- › Complicated, nested layout can be applied as well
 - › Improves scalability
- › Interactions can be added to any item
- › Animations can be added in the timeline editor
 - › Item, property, key frame values, duration
- › Would be very time consuming to explain in a UI specification

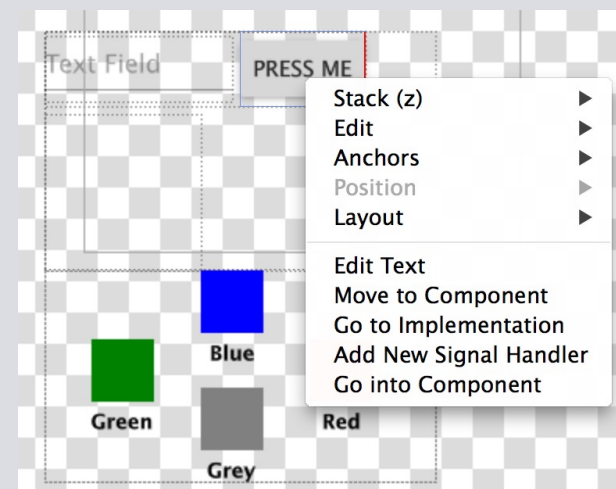
Re-using designs

Scalability to different displays

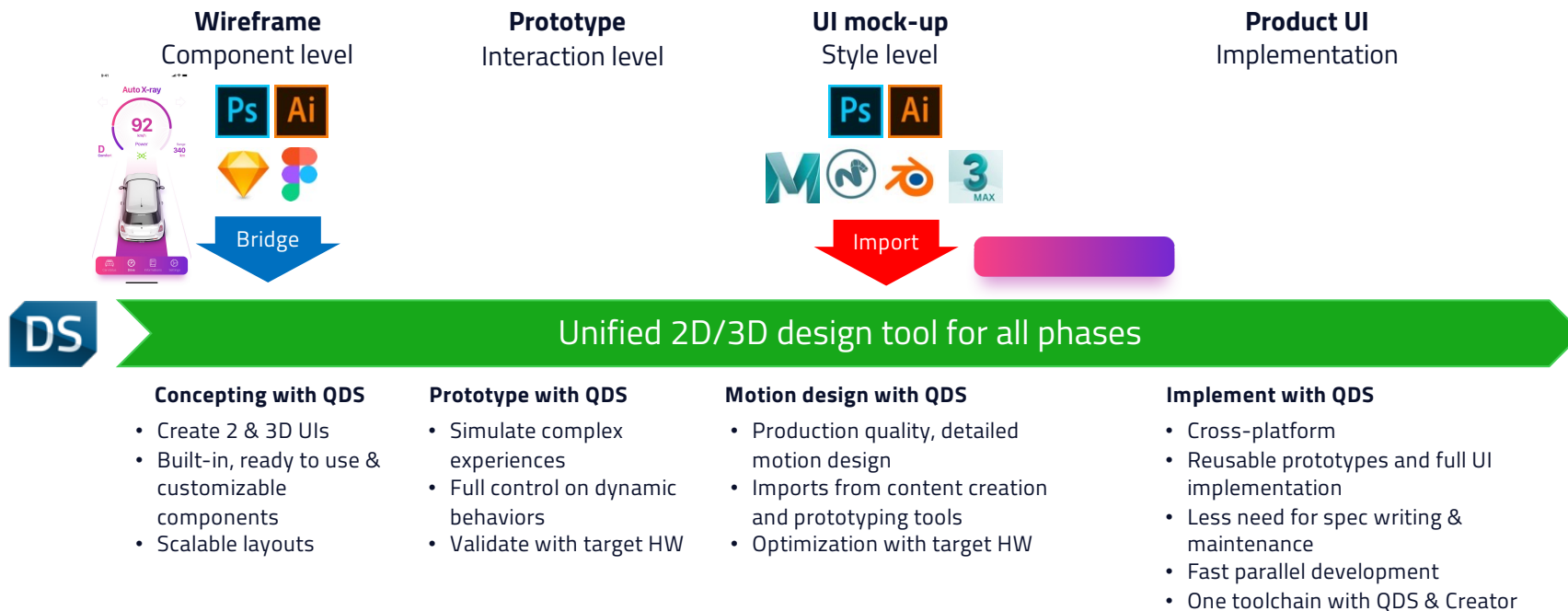


Interaction design with minimal code

- › Any UI item can be added an interaction
 - › Tap handler
 - › Swipe handler
- › Interactions change item properties
 - › Hides an item
 - › Rotates an item
 - › Changes the geometry
 - › Starts an animation
- › No code needed to implement these



Qt Design Studio from concept to final implementation



That's all folks!
Thank You!

Q&A

Tino Pyssysalo
tino.pyssysalo@qt.io

