

FIT3175 - Usability

UX Design Processes and Methods

Week 4 Lecture P1

Copyright Warning

Commonwealth of Australia Copyright Act 1968

Warning

This material has been reproduced and communicated to you by or on behalf of Monash University in accordance with section 113P of the Copyright Act 1968 (the Act).

The material in this communication may be subject to copyright under the Act. Any further reproduction or communication of this material by you may be the subject of copyright protection under the Act.

Learning objectives

Design Process Frameworks

- What is design process?
- Why implement process frameworks?
- Agile UX, Lean UX and Design Sprint

Design Process Methods

- Common design methods
- Understand, design, sketch, decide

Prototyping

- Prototyping methods
- Low-fidelity vs high-fidelity
- Static and interactive prototyping

Design Process Frameworks

What is design?

There are different perspective on how **design** works.

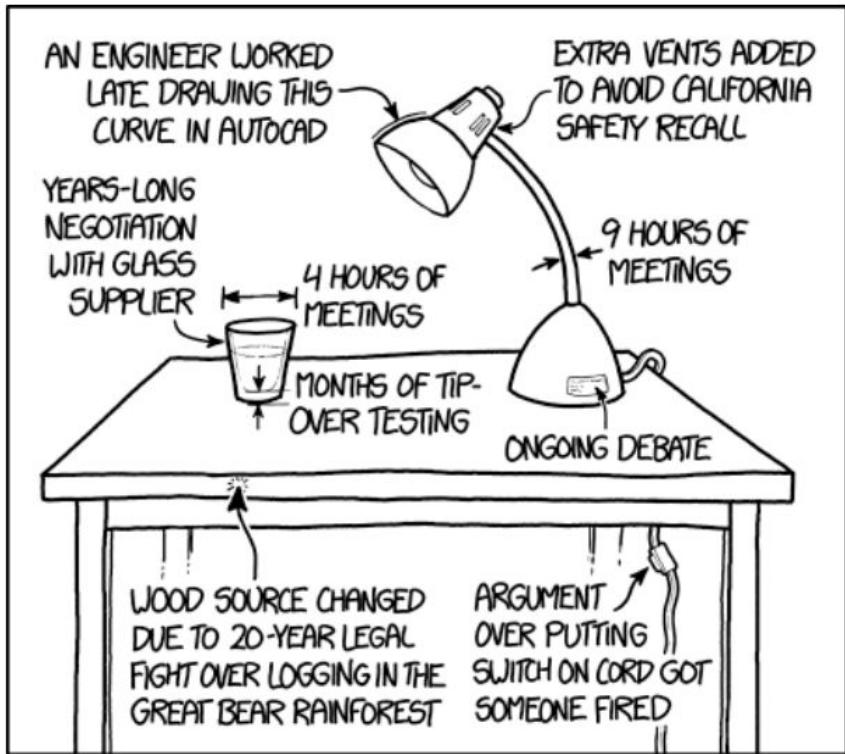
"Making plans and specifications for digital objects, which include devices, interfaces, services, and information."

- Ben Shneiderman, Designing the User Interface

"People think it's this veneer -- that the designers are handed this box and told, 'Make it look good!' That's not what we think design is. It's not just what it looks like and feels like. Design is how it works."

- Steve Jobs, New York Times article (2003)

Design is...



... to conceive, plan and create in order to solve a problem.

If design is "**problem solving**", then we are all designers.

- When you rearrange your kitchen cupboards, for better access to your pans, you're doing interior design.
- When you create a reminder on your phone for an assignment, you're doing information design.

SOMETIMES I GET OVERWHELMED THINKING ABOUT THE AMOUNT OF WORK THAT WENT INTO THE ORDINARY OBJECTS AROUND ME.

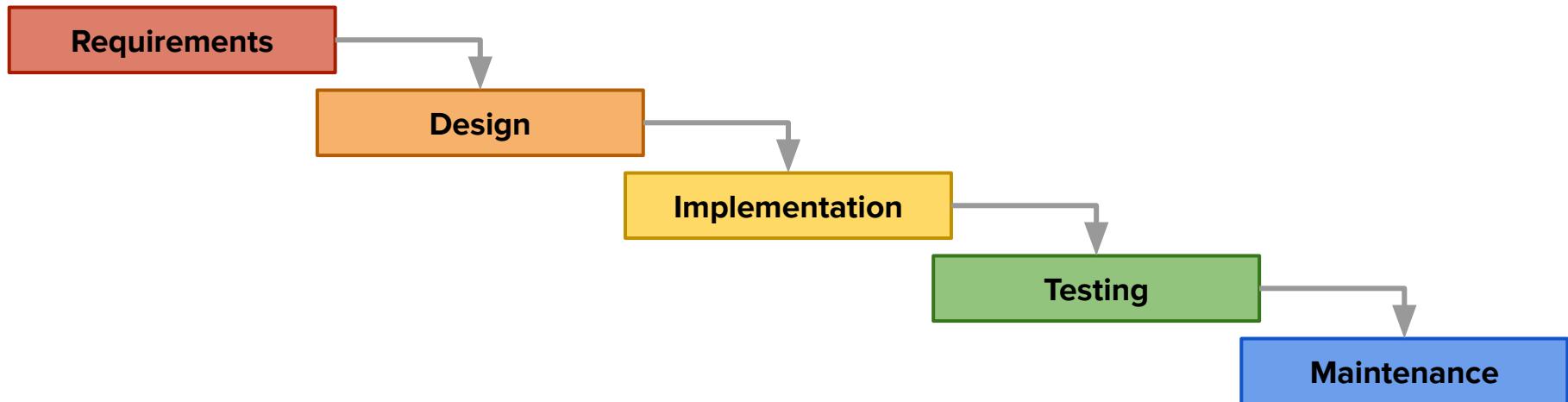
Design skills

Therefore, being skilled at design requires problem-solving skills.

- Seeking multiple perspectives
- Divergent thinking
- Convergent thinking
- Exploiting failure
- Externalizing ideas
- Maintaining Emotional Distance
- Seeking Critique
- Justifying Decisions

Waterfall development process

This process breaks a project down into a linear sequence of independent phases.



This process model is thought to have originated in the manufacturing and construction industries. ***What is the problem with this process for UX design?***

System-centred vs User-Centred Design (UCD)

A design process that is **system-centred**

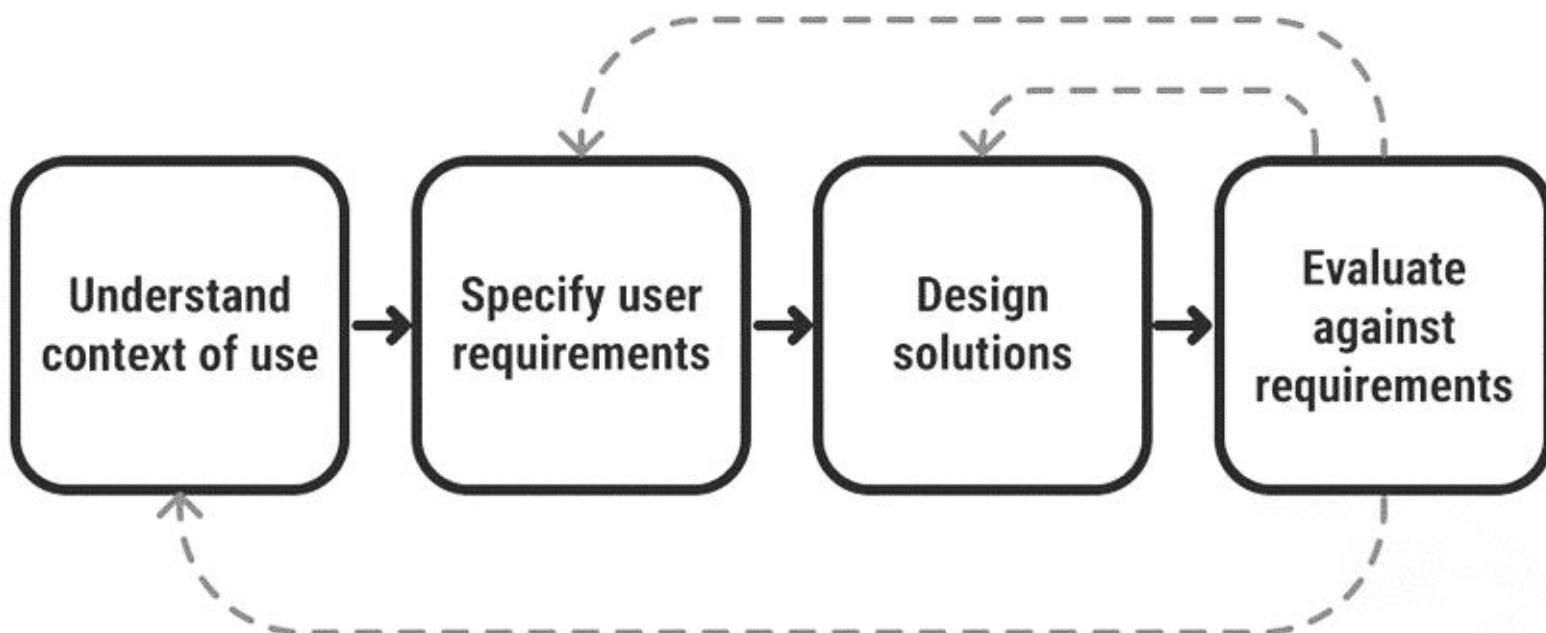
- Focus on implementation
- Focus on business goals
- Measure KPIs based on system performance
- Considers engineering of technical features

A design process that is **user-centred**

- Focus on user needs at every stage
- Involve users or represents users at every stage
- Use metrics based on usability measures
- Considers the entire user experience

Recap: Iterative UCD processes

Generally, a UCD process involves 4 distinct phases. Each phase has a different purpose and design methods to achieve outcomes.



UCD advantages and disadvantages

Advantages include:

- Can discover specific user needs.
Define realistic limitations.
- The design can better manage user expectations.
- Get greater acceptance of end product/service.
- Reduce product risk at the end.

Disadvantages include:

- Involving the user takes time, so may also increase costs
- Users are not great at precisely saying what they want
- Finding available participants can be difficult
- Users have little sense of project development constraints.

What are process frameworks?

A **process framework** is strategy tool that supports a specific design workflow:

- **A common philosophy for team members**
 - Allows individuals to strive toward a common goal
- **A common set of processes**
 - Ensures that team members are effectively utilised
- **A common understanding of workflow phases**
 - Consistent work through team harmony
 - Simpler scheduling and delegation of work
- **A common set of design methods and practices**
 - Problem-solving and decision-making tools

Background: Agile development

In response to over-regulated and heavy-managed processes (typically using waterfall processes), **Agile Software Development** focuses of iterative, adaptive planning, early development and continual improvement.

- **Individuals and interactions** over processes and tools
- **Working software** over comprehensive documentation
- **Customer collaboration** over contract negotiation
- **Responding to change** over following a plan

This may not suit all teams or projects. Some critics claim that Agile development emphasizes method over results.

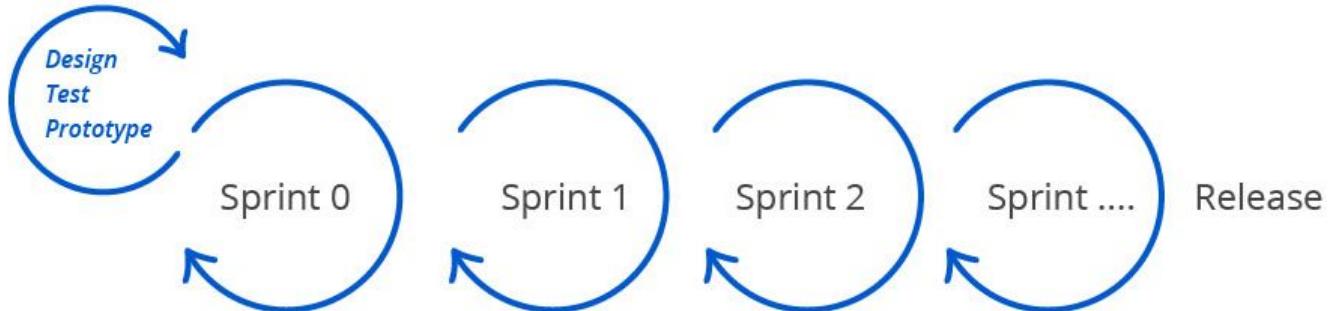


Agile UX

As opposed to completing the project as a whole, break it down into small chunks.

- Divide a project **epic** into smaller **stories**.
- Each story is treated as an entire mini-project.
- Conduct a project **sprint** to work on a story.
- Evaluate and iterate.

Right: UCD is frontloaded in this process, with successive development sprints iterations working towards a major release.



Agile UX workflow features

Agile UX follows a iterative process that is adopted holistically by team members. Work is conducted in **scrum teams** working in short sprints.

1. **Sprint planning**

Define the duration (2-4 weeks), goal and intended work items for the sprint.

2. **Daily scrum**

15-minute meetings conducted to inspect sprint progress.

3. **Sprint review**

A longer meeting to demonstrate completed work to stakeholders for feedback.

4. **Sprint retrospective**

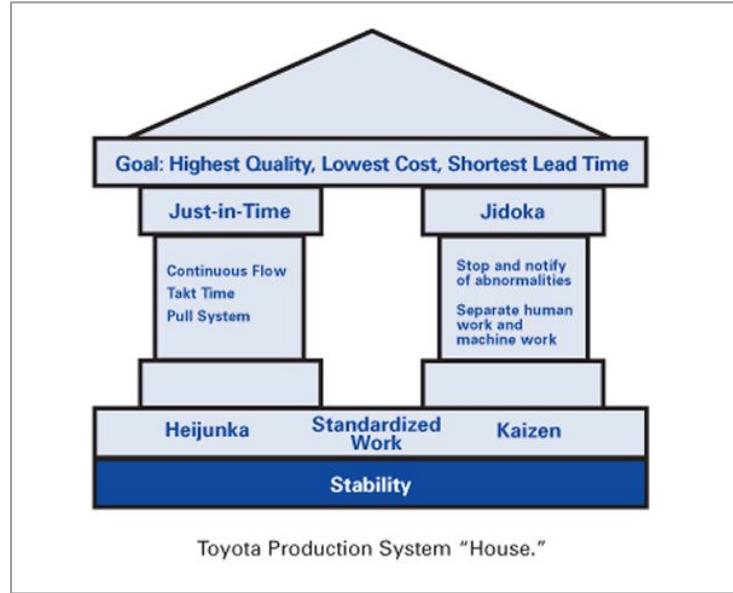
A longer meeting at conclusion to identify sprint process improvements.

Background: Lean manufacturing

Lean manufacturing is a methodology used in production of physical products.

Principles of this method aim to reducing times within the production system as well as response times from suppliers and to customers.

- 1. Value:** Customer-centric at all stages.
- 2. Value Stream:** Challenge wasted steps.
- 3. Flow:** Work is continuous throughout.
- 4. Pull:** Flow between steps based on demand.
- 5. Perfection:** Continuous value optimisation.



Above: The Toyota Production System is supported by "pillars" that aim to reduce wasted resources in production lines. Similar principles are applied in Lean.

Lean UX

An iterative development process popular with startups. Where Agile UX is very process-focused, Lean UX aims for quick delivery of a testable product.

Lean UX Principles

- **Early customer validation** vs releasing with unknown end-user value.
- **Collaborative cross-functional design** vs lonely hero design
- **Solving user problems** vs adding cool features
- **Measuring key performance indicators** vs undefined success metrics
- **Applying appropriate tools flexibly** vs following rigid methodology
- **Nimble design** vs heavy wireframes or restrictive specifications

Lean UX workflow features

- **Assumptions**

Define a problem and the assumptions to be tested.

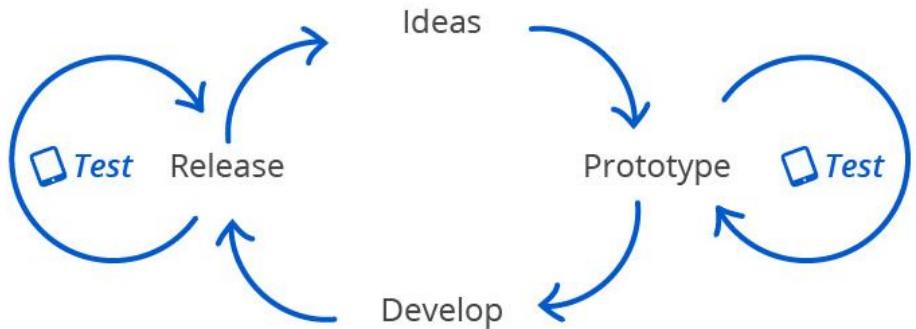
- **Hypotheses**

We believe [doing this] for [these users] will achieve [this outcome].

- **Minimum viable products (MVP)**

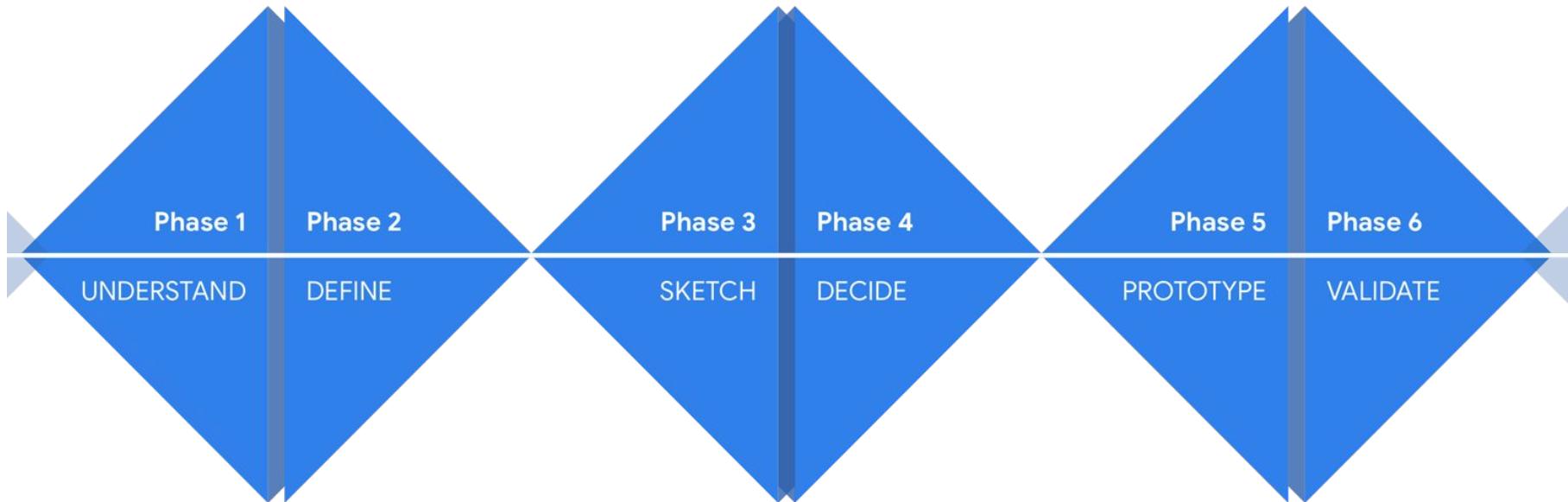
Prototype or develop the bare minimum required to test the hypothesis.

Right: UCD is present in each development cycle.
Continual iteration that involves testing in prototype/release forces user feedback to be incorporated to strive for continuous improvement.



Google Design Sprint

Google's framework provides design teams with defined sprint phases and methods.



A typical sprint lasts for about 5-days. Duration and selection of methods is flexible.

Design Process Methods

Design Sprint Stages

Each phase of an iterative design process has different goals and methods.

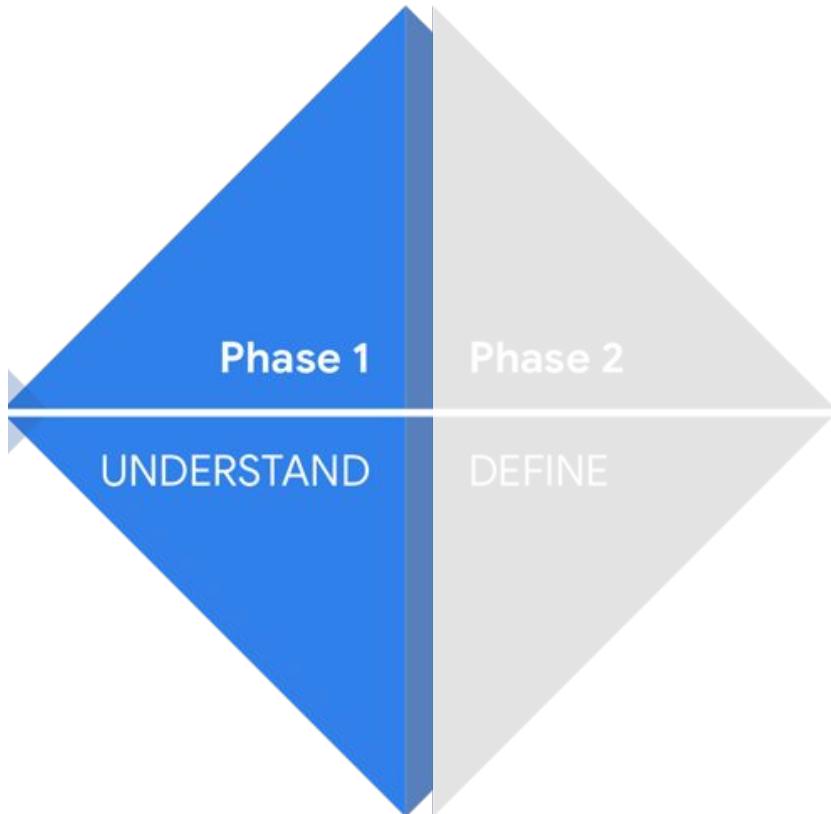
For example, consider the 6 stages of the **Google Design Sprint** framework.

1. Understand
2. Define
3. Sketch
4. Decide
5. Prototype
6. Validate

**The phases must be completed in order this order.
Flexibility comes from the choosing methods used.**

We will look at 3 examples each of common methods used in phases 1-4.

1. Understand



Create knowledge and articulate the problem space.

At the start of a project, there are many possible directions and many unknowns.

- Research the problem space
- Sharing of ideas
- Creation of shared knowledge
- User research
- User empathy

1. How might we... ?

A method for generating and collections ideas, insights and pain points.

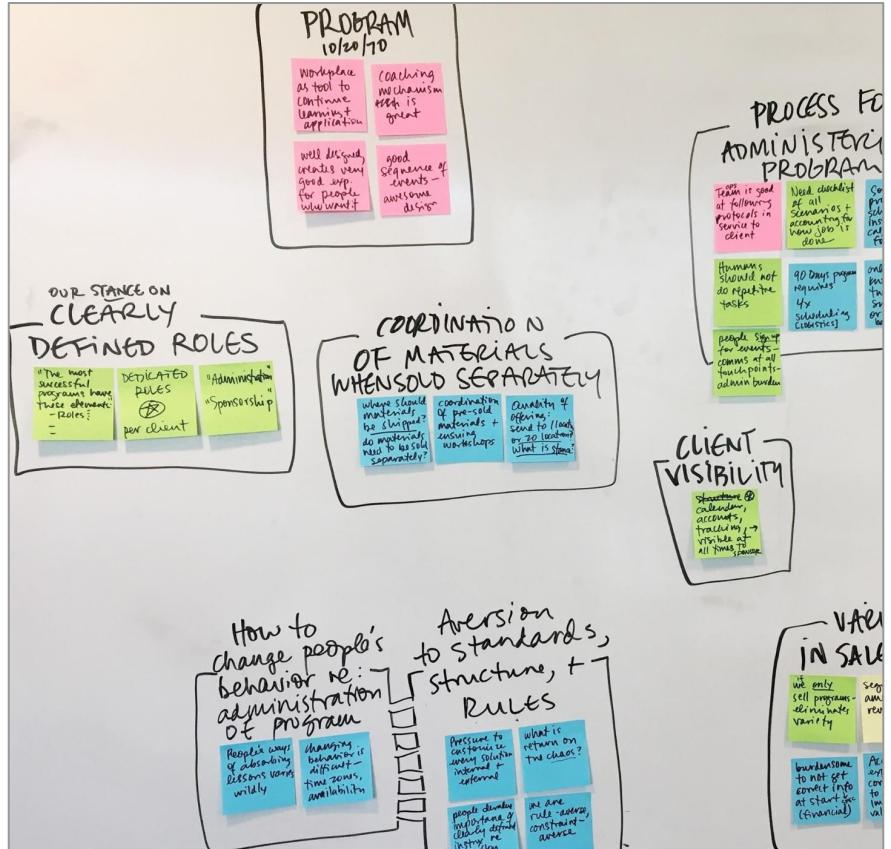
"**How Might We**" is a prompt that suggest a potential problem for a team to solve.

- Write each idea on a sticky note.
- Reframe pain-points as opportunities.
- HMWs notes are read aloud to encourage further ideas.
- HMWs can be collected into groups.



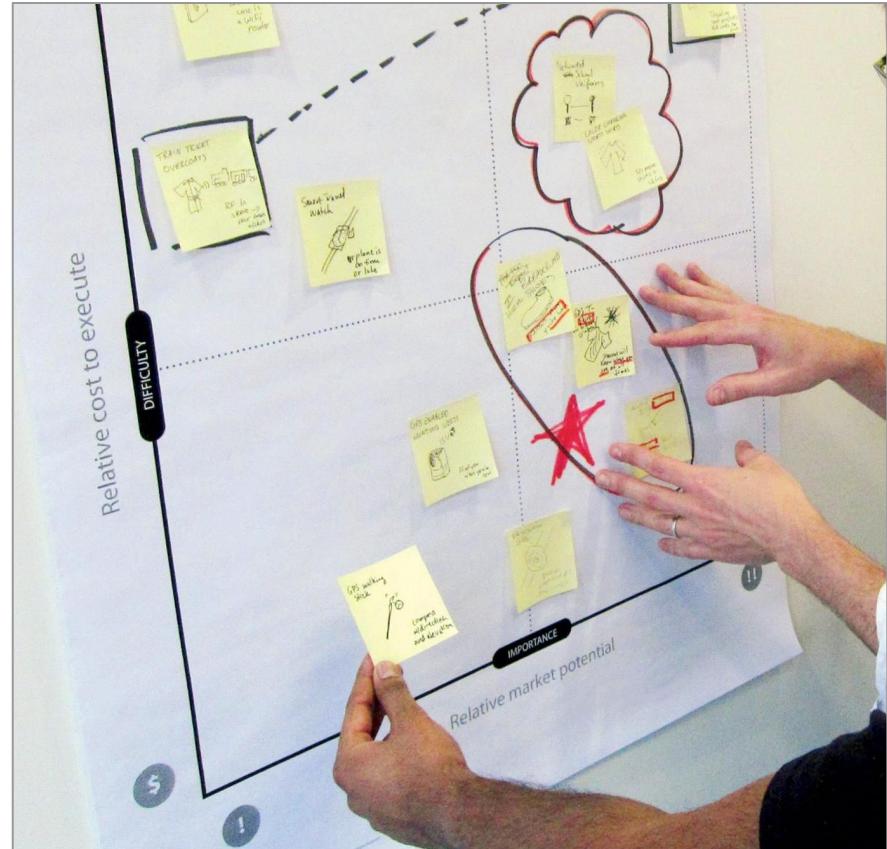
1. Understand: Affinity clustering

- Identify a discussion topic.
- Gather ideas. Record each on a separate card or sticky note.
- Start with one person describing and placing an item.
- Others to place similar items in proximity. Repeat the pattern until all items are included.
- Discuss and rearrange items into groups and sub-groups.
- Label clusters that finally take shape.

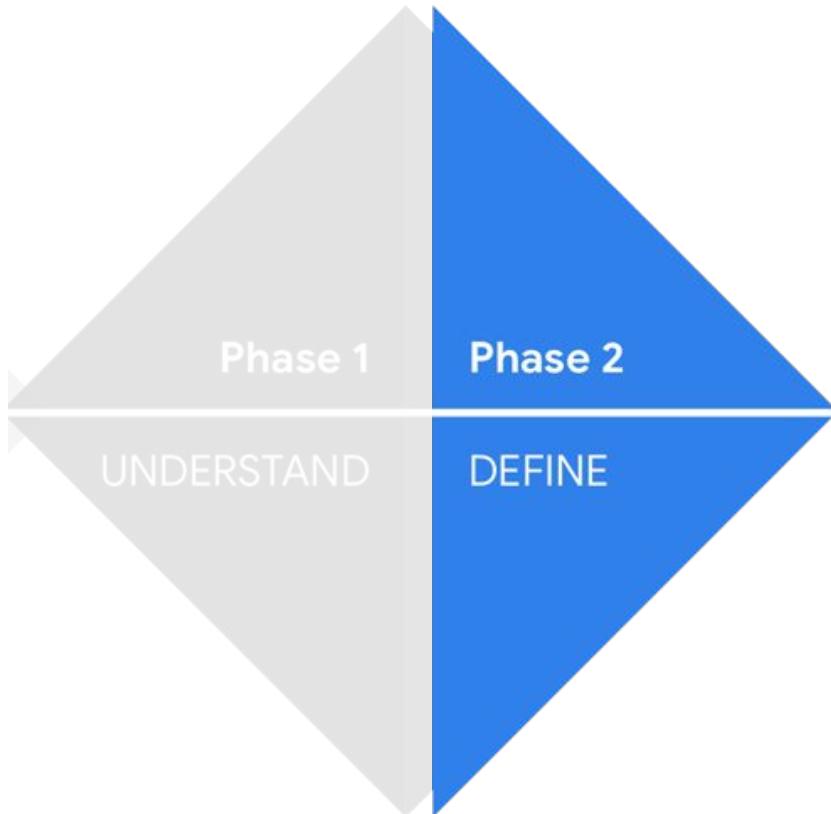


1. Understand: Importance/Difficulty Matrix

- Identify a project that requires prioritization.
- Make a large quad chart.
 - Horizontal axis = Importance.
 - Vertical axis = Difficulty.
- Gather ideas for discussion.
- Plot items on the chart based on
 - Relative importance.
 - Relative difficulty.
- Consider the quadrants where items are placed to set priorities.



2. Define



Focus the problem by defining goals, success metrics, and signals.

Phase 1 produces many ideas and data. We now focus those ideas into specific context and desired outcomes.

- Define important components
- Define specific goals
- Understand the desired successful outcomes.

2. Define: Success Metrics and Signals

In order to set goals, team members must understand how to determine if outcomes are successful:

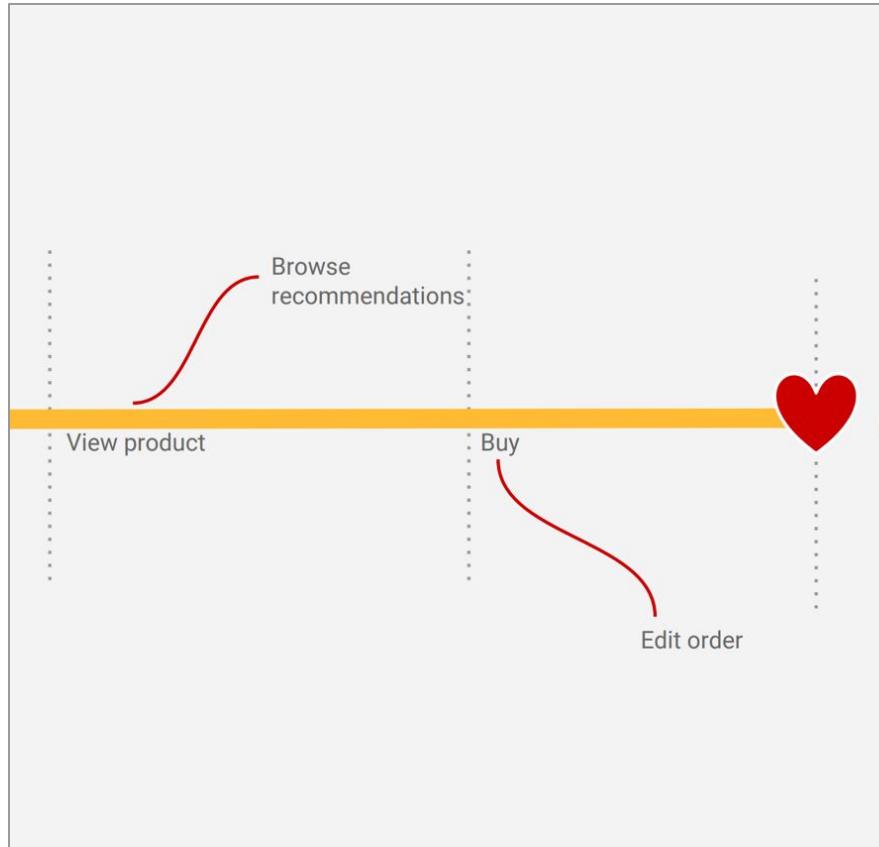
- **Goal:** What is the problem are you trying to solve?
- **Signal:** What noticeable change in user behavior would indicate the goal is successfully achieved?
- **Metric:** How can this change in user behavior me measured?



2. Define: Golden path

Identify the key user journey for a goal - focus efforts on an ideal path without error.

- **If the product already exists**
 - list possible user stories
- **If the product does not exist yet**
 - sketch the ideal path through the product as you understand it.
- Define the ideal path through the journey - this is **the golden path**.



2. Define: Brand personality sliders

Decide the products branding strategy.

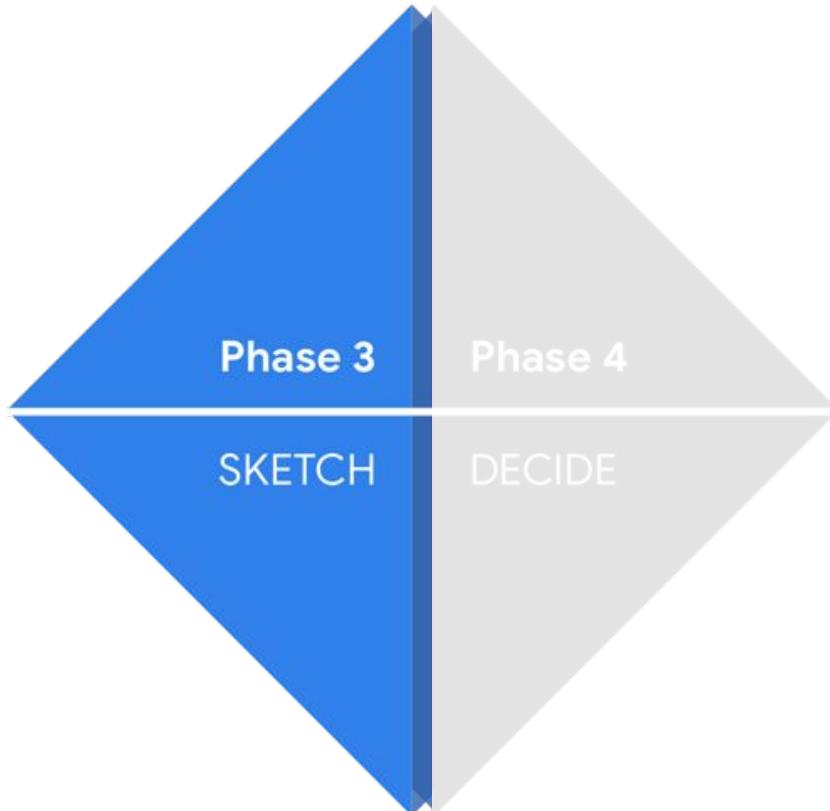
Draw 5 range scales on a whiteboard.

- **Elite to Mass Appeal**
- **Serious to Playful**
- **Conventional to Rebel**
- **Friend to Authority**
- **Mature & Classic to Young & Innovative**

Team members mark the scales to show where the brand may fall in each range.



3. Sketch



Generate ideas, then narrow down to a possible solutions.

With a strong shared understanding of the goals, we now propose solutions.

- Sketching to generate potential ideas
- Explaining, listening and suggesting
- Sketching to refine ideas
- Generate an number of viable concepts for a solution

3. Sketch: Comparable problem

Sometimes a solution already exists. This warm-up activity has the potential to significantly reduce future work.

- Each team member does research.
 - Direct competitors
 - Indirect-competitors
- Each person has 3 minutes to share discovered ideas within the group.

Often existing ideas and patterns can be reworked, recontextualised and improved.



3. Sketch: Crazy 8's

A fast sketching exercise that challenges people to sketch 8 ideas in 8 minutes.

- Each team member folds a piece of paper into eight sections
- Set the timer for eight minutes
- Individually, each team member sketches one idea in each rectangle, filling the paper in 8 minutes.
- Each idea is briefly explained.
- Each team member can vote for 3 ideas to help identify good ideas.



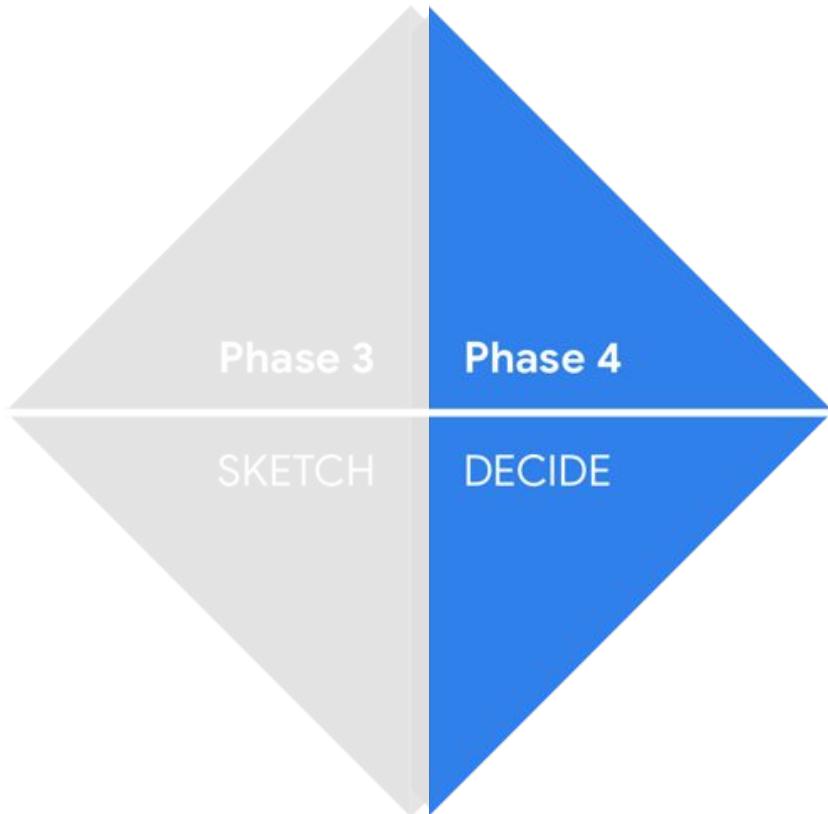
3. Sketch: Solution sketch

Present a solution idea as a new sketch with multiple frames to show how it works

- Each team member select an idea they think is the best.
 - Can be a combination of ideas.
 - Can be ideas of other members.
- Flesh out the idea in a new sketch.
 - More detail and clarity.
 - Multiple frames to show steps.
 - Explain using annotations, pictures, and words



4. Decide



Finalise the direction of the concept so that the team knows what to prototype.

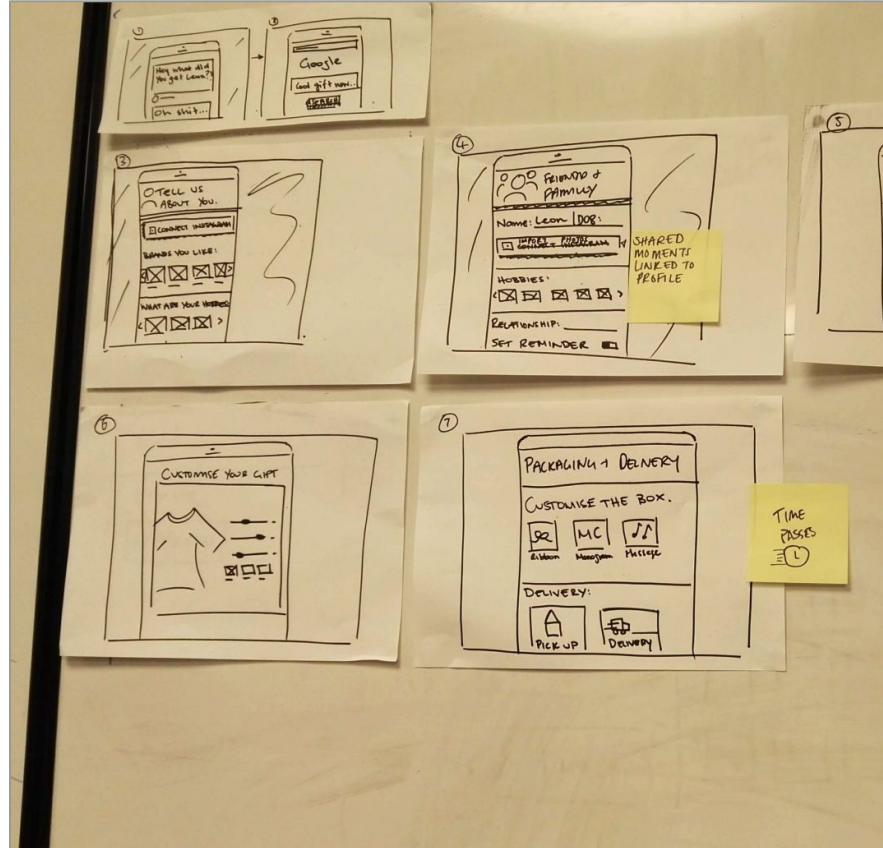
- Presentations of individual concepts for the solution
- Voting and deciding on a preferred solution
- Deciding a strategy for future stages.

At this stage a team must find consensus on the final direction of the sprint.

4. Decide: Solution presentations

A sketch can only say so much. Team members are given an opportunity to pitch their idea to the team.

- Put sketches up on a wall - clearly visible to all team members.
- Each team members presents a short pitch about their proposed solution.
- Other team members can ask questions or discuss details of the idea.



4. Heatmap and dot voting

A voting process that can be used for entire solutions, specific features or ideas.

- Put all solution sketches up on a wall.
- Give each team member a number of votes, represented by sticky dots.
 - 3 votes each for solution voting.
 - Unlimited smaller dots when voting for specific features.
- Team member place dots on sketches and features that they like.



4. Decide: Rumble or All-In-One?

After considering all options and voting,
the team needs to decide:

- **Go for an all-in-one?**

Combine best ideas into 1 prototype.

- **Go for a rumble?**

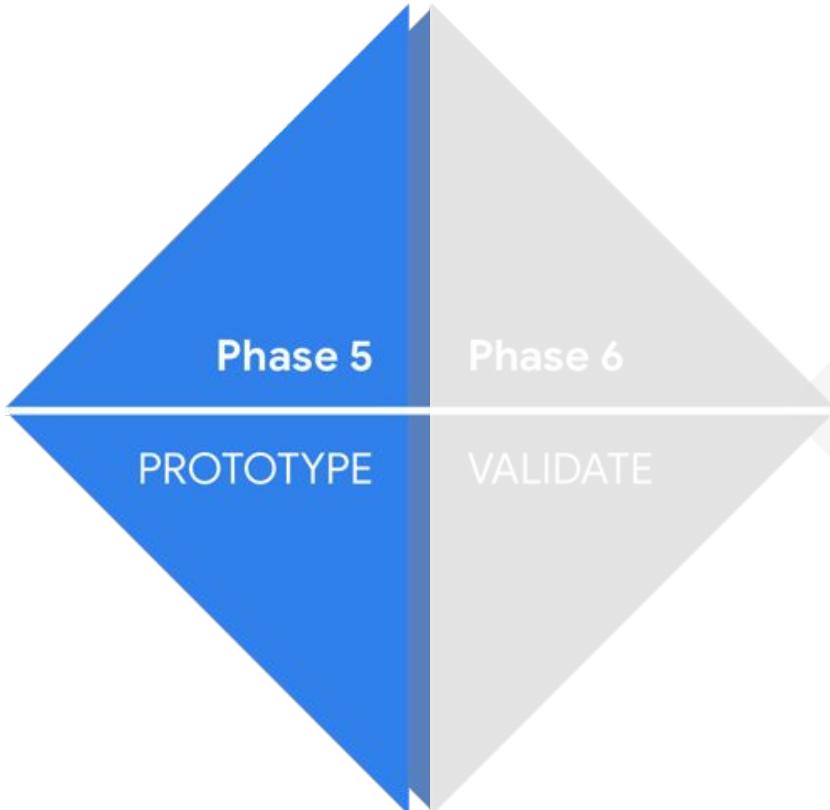
Develop 2 ideas and test them
against each other.

If the team chooses to rumble:

- How to present designs to users?
- Plan the testing methodology.



5. Prototype

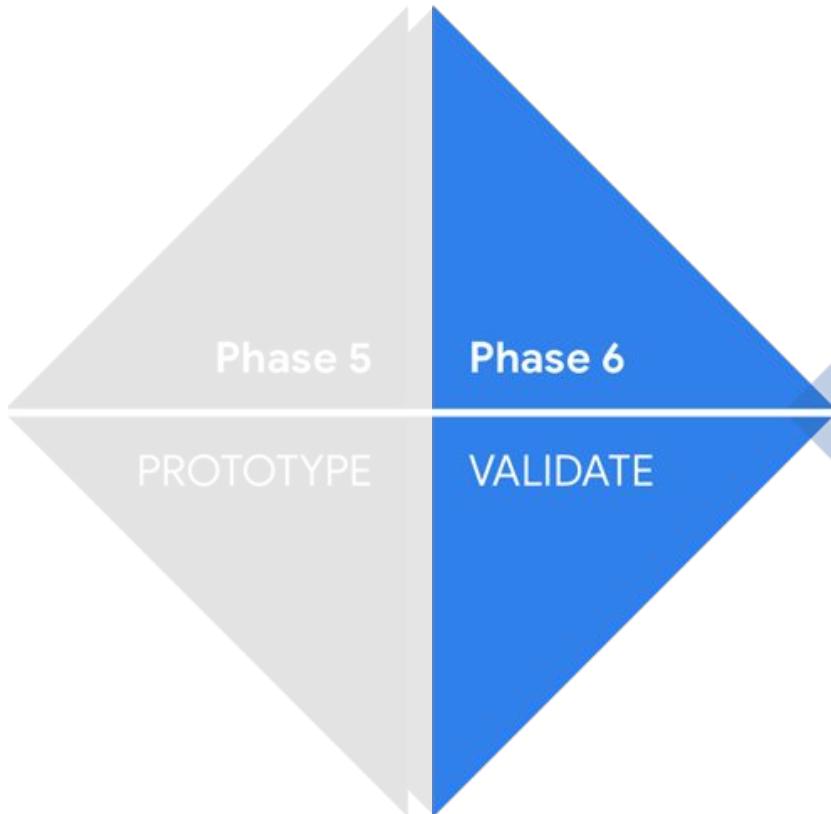


Build a version of the concept that can be tested.

- Assign tasks
- Choose prototyping style and tools
- Track work progress
- Review prototypes and identify required revisions.

Prototyping is covered separately in this lecture.

6. Validate



Get testing feedback from users, conduct feasibility reviews.

- Test prototypes with users to evaluate usability.
- Collect data to validate metrics
- Meet with stakeholders
- Review the sprint process

UX evaluation methods will be covered in the next lecture.

Prototyping

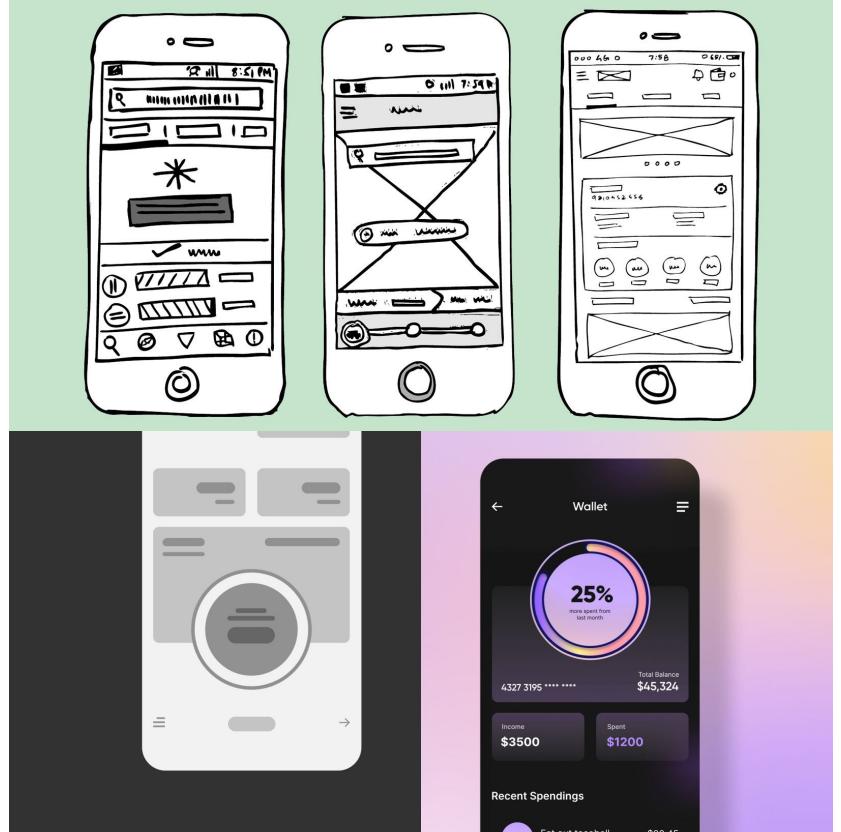
The purpose of a prototype

A prototype is a simple representation of a design that can be used for evaluation and validation.

Different **tools** can be used to create prototypes at different levels of **fidelity**.

Compared to developing a working product, prototypes are:

- **Low risk to develop**
- **Low cost to iterate**



Tracking team progress: Kanban boards

The Kanban Method

To Do

- Time out when accessing reports
- Webpack update

Development

- Analyze transactions performance
- Setup Staging 2 test environment

Code Review

Testing

Done

Butler

Show Menu

Prototype development can be one of the most resource intensive stages of a project.

Make sure team members understand the scope of the work ahead by listing out individual job tasks.

In a Kanban board (shown here in Trello), tasks are moved from left to right as pass complete each stage of development.

Types of prototypes

Prototypes run a full gamut of fidelity and complexity. "**Fidelity**" refers to the level of detail shown in the designs.



Low-fidelity prototypes

- Simple wireframe representations.

Higher levels of fidelity and complexity typically result in a increased level of risk.



High-fidelity prototypes

- Fully-realised visual representations.

Time commitment, emotional attachment and effort increases need to be managed with careful planning.



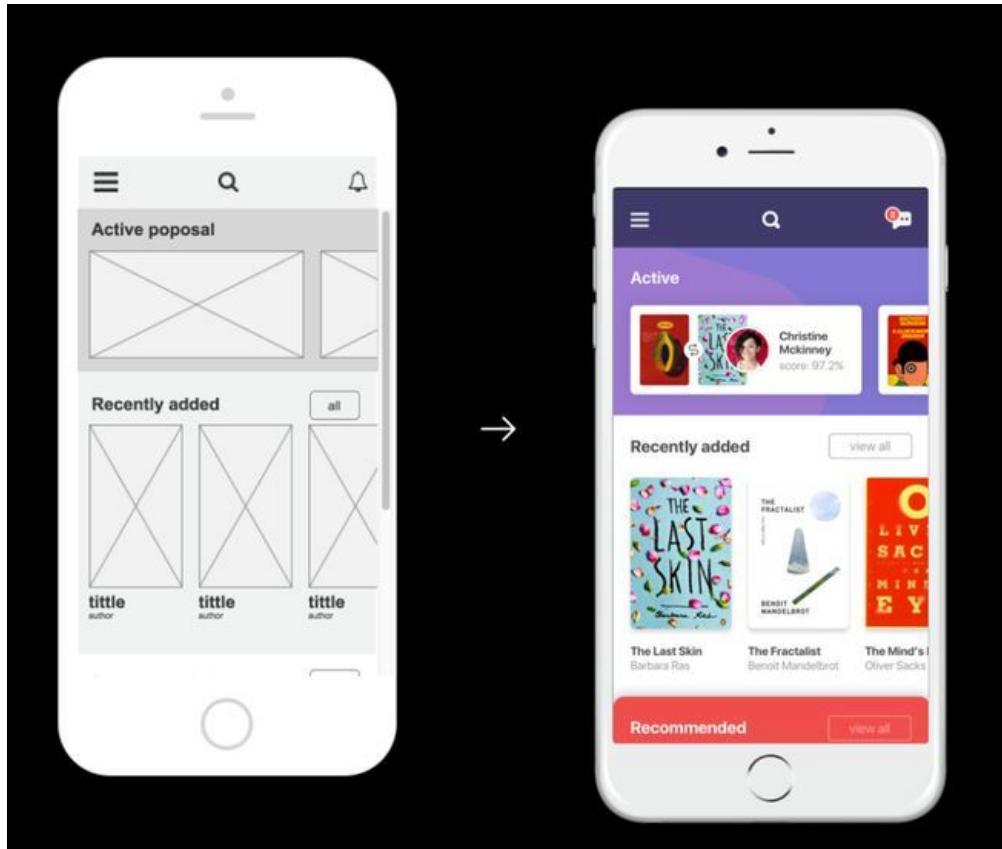
Native prototypes

- Working software representations.

Prototype fidelity considerations

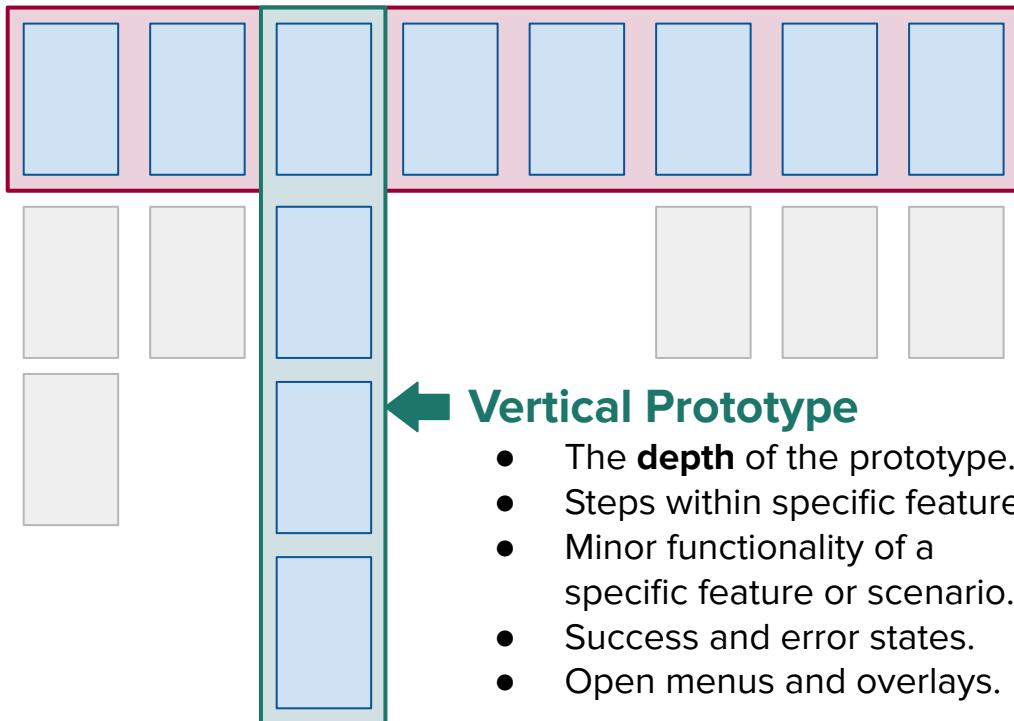
Increasing a design's fidelity may involve different changes:

- Accurate representation of screen **content**.
- Incorporating visual **design styles** and branding details.
- Adding **breadth and depth** to depict more states.
- Implementing **navigation** or realistic user interactions.



Prototyping broad or deep?

Imagine laying out all of your prototype screen designs in a 2-dimensional grid.



Horizontal Prototype

- The **breadth** of prototype.
- Depicts the range of features.
- Major sections of an app.
- Screens that depict different major functionality.

Vertical Prototype

- The **depth** of the prototype.
- Steps within specific features.
- Minor functionality of a specific feature or scenario.
- Success and error states.
- Open menus and overlays.

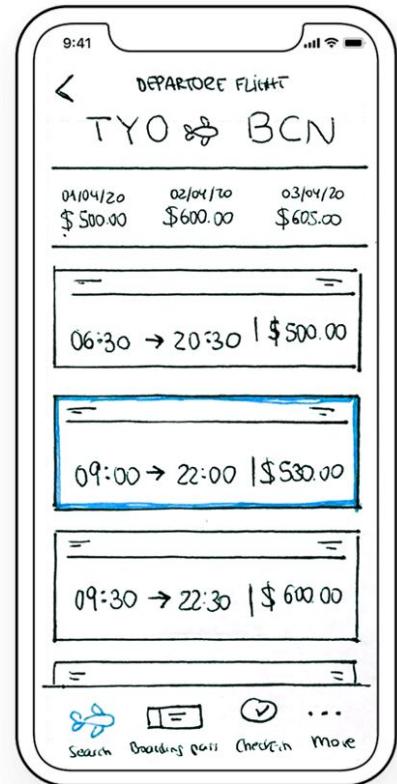
Comprehensively depicting depth and breadth is a lot of work! Aim to balance both by selectively designing the most important screens.

Low-fidelity wireframe sketch prototypes

A low-fidelity is used for early visualisation of an ideas due to its low cost and quick turnaround time.

Prototypes should be clear pencil and paper sketches.

- **High usability with minimal learning curve**
 - No time spent exploring software features of tools
 - No worrying about technical troubleshooting
- **Less emotional attachment to early ideas**
 - If an idea doesn't work, scrap it and move on!
 - Less time spent refining details.
 - More time spent generating unique ideas.



PICTIVE prototypes

Plastic Interface for Collaborative Technology

Initiatives through Video Exploration - a prototyping method that encourages users to participate in the design process:

- Construct screen elements using pen and paper, highlighters, sticky notes, etc.
- Work in a shared design space.
- Test users propose scenarios and interact.
- Designers can easily create, modify and update elements and screen layouts.
- Sessions are recorded for analysis.

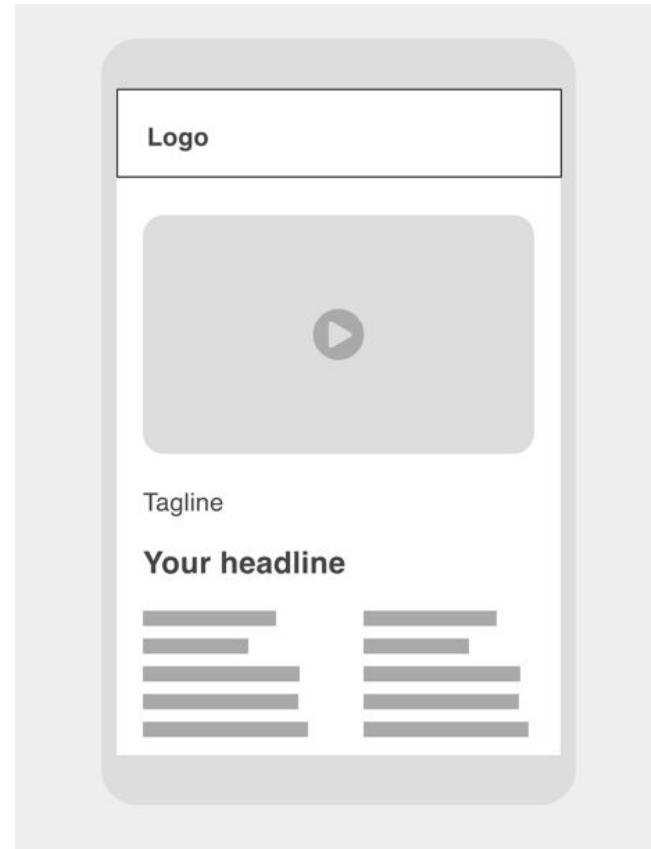


Low-fidelity digital wireframes

Low-fidelity prototype can be created using digital tools. **However**, designers must be careful to focus on low-fidelity problem solving.

- Continuously editable designs.
- Easier to organise and align elements.
- Clearer representation of some small detail.

A designer creating digital wireframes must exercise discipline to maintain a low level of fidelity. Avoid temptation to start refining designs before testing.



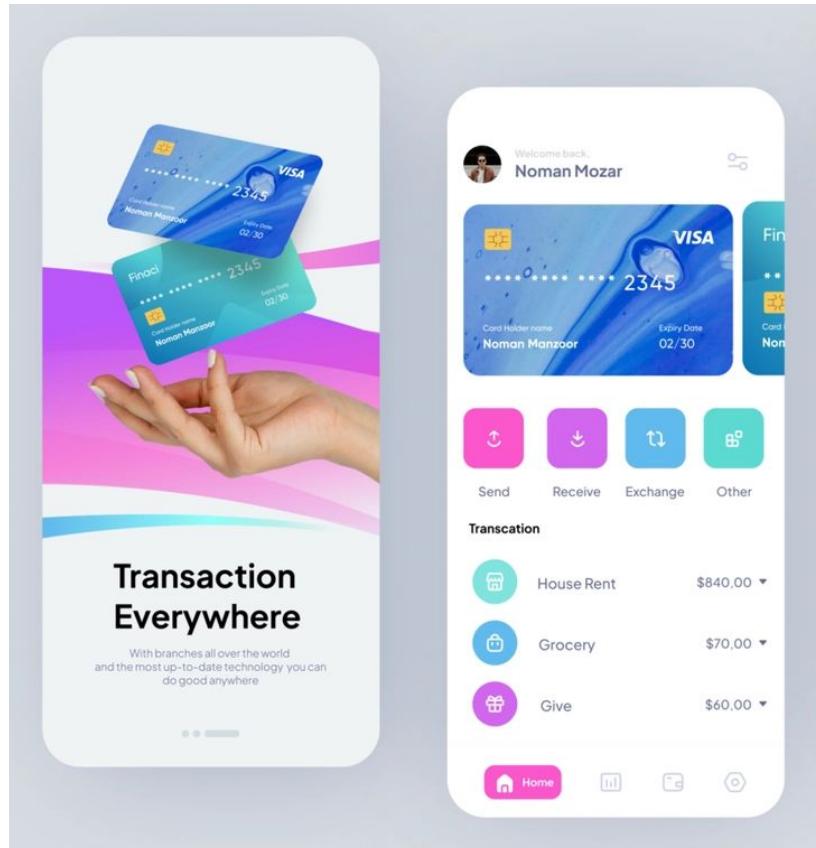
High-fidelity digital mockup prototypes

The higher the fidelity, the more the visual design detail **resembles a real product.**

- Application of design styles
- Inclusion of branding design
- Realistic text and image content

High-fidelity prototype visuals are typically produced using:

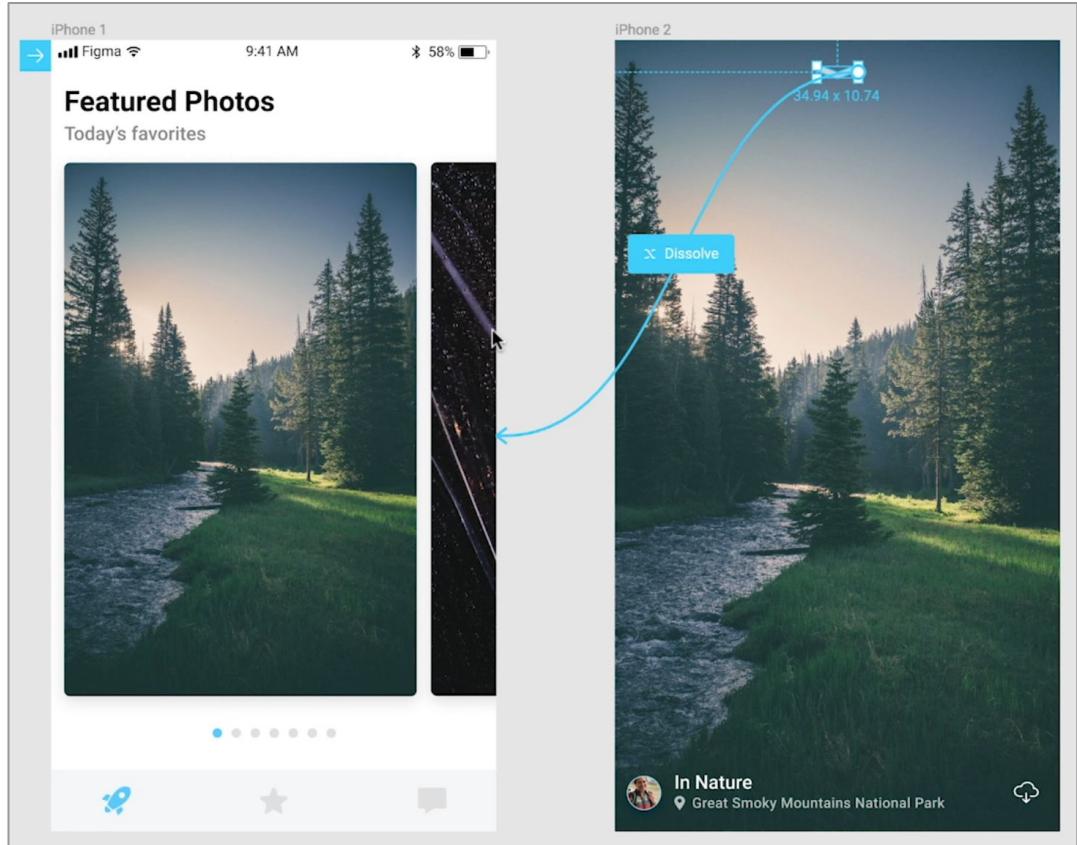
- Graphic design tools
- UX design tools



Prototyping navigation flows

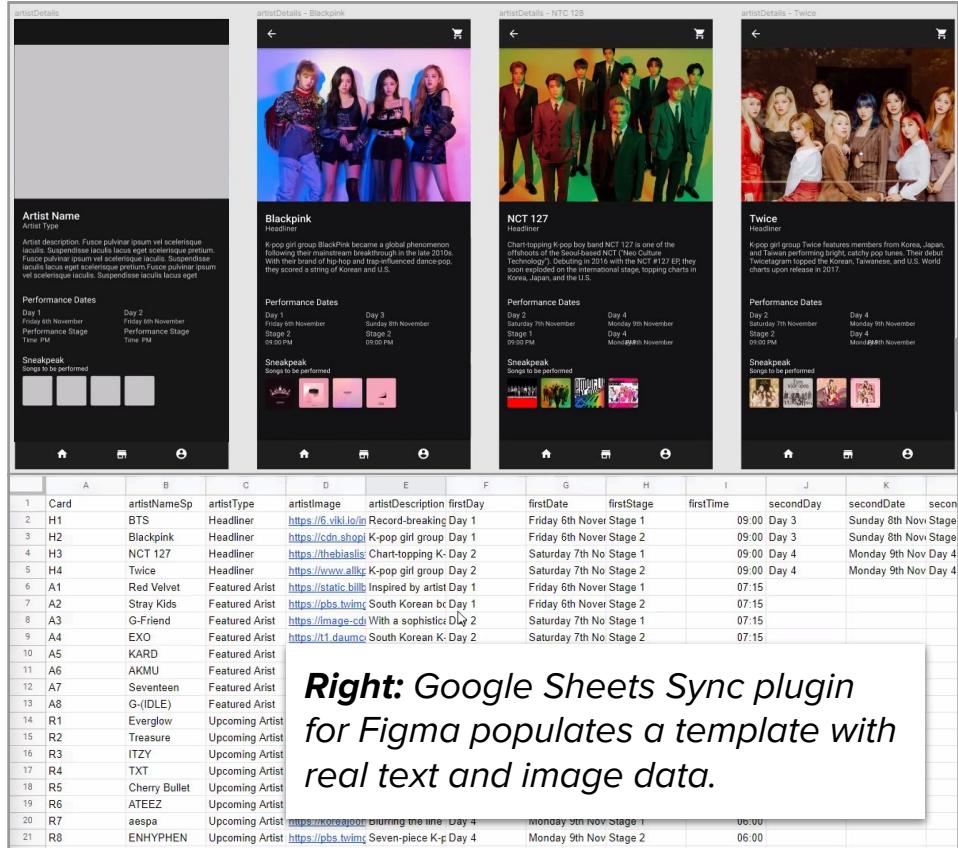
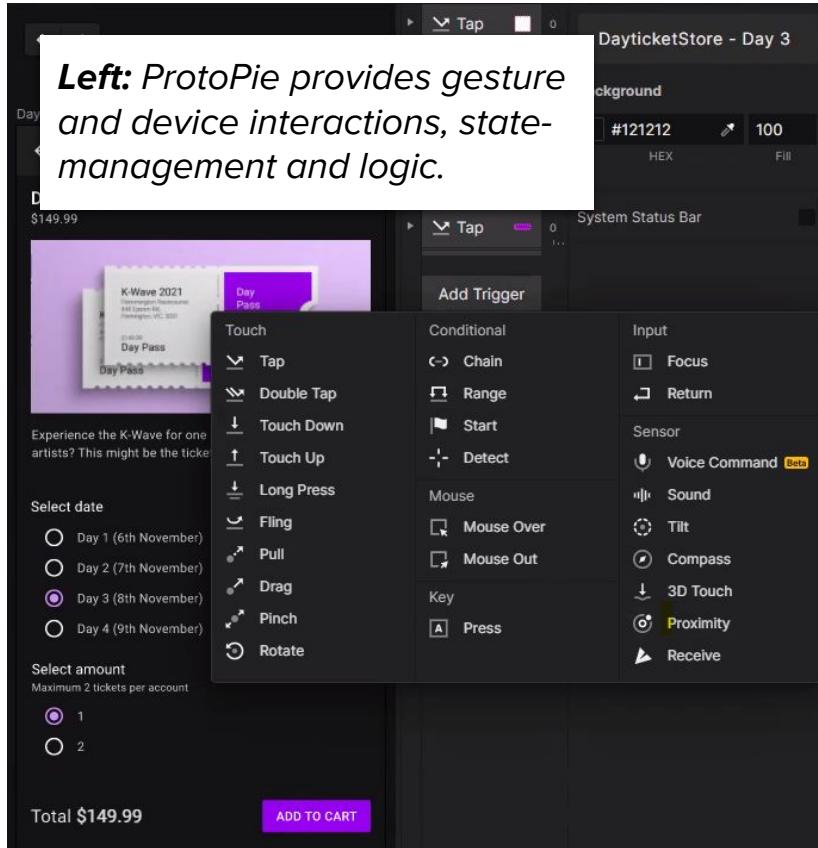
Interactive prototypes allow digital prototypes to simulate basic product behaviour.

- **Navigation elements**
- **Navigation flows**
- Screen overlays
- UI element states
- Scrollable screen overflow
- Basic animation



Advanced interactions and data-driven content

Left: ProtoPie provides gesture and device interactions, state-management and logic.



Native prototypes

Native prototypes are software implementations of a system.

- Prototypes created using rapid application prototyping techniques.
- Real alpha/beta software that may be missing some features.

Native prototypes allow users to test realistic uses.

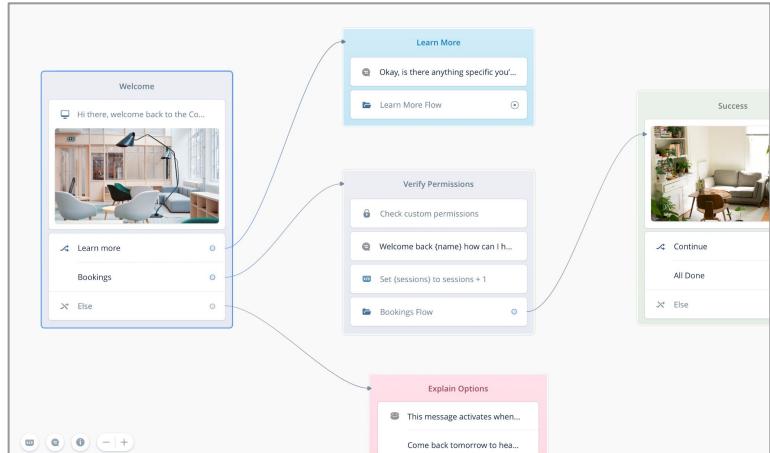
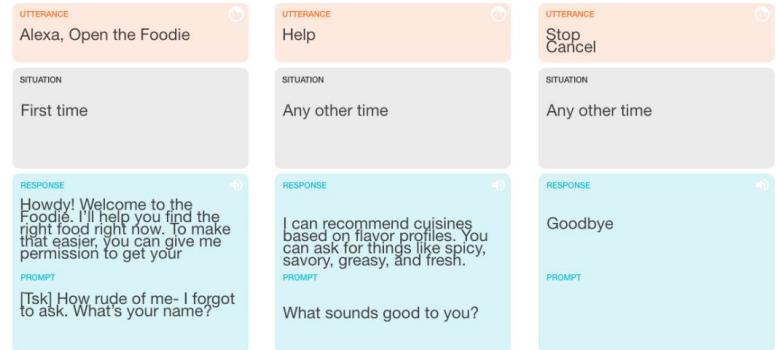
Prototyping voice interactions

Smart assistance and voice input present an interesting challenge to designer for:

- Prototyping without a UI
- Validation through testing

Top-right: Amazon's **Alexa** developer documentation instructs designers construct dialogue scripts for voice interaction when designing skills.

Bottom-right: Digital tools such as **VoiceFlow** allow designers to visualise and test voice proposed interaction flows.



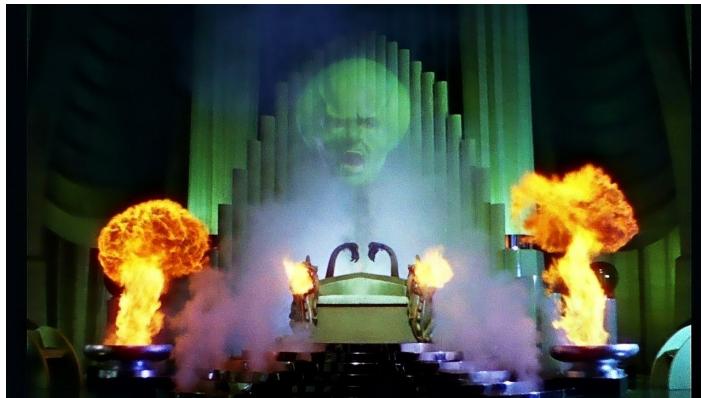
Wizard of Oz (WOZ) prototyping

How can you design and test features of a system that can't be implemented in a prototype?

Top right: Dorothy follows the Yellow Brick Road to Emerald City, where she can ask the "Wizard of Oz" to help her return home.

The Wizard stalls in fulfilling his promise until Dorothy's dog, Toto, pulls back a curtain, exposing that the "Wizard" is just a con man operating machinery.

Bottom right: Observe a prototype test of a gesture-based TV control system. **How is this prototype being demonstrated if this gesture system has not been integrated with the TV?**



Next session

- UX evaluation methods
- Nielsen Heuristic Evaluation

Reminders

- **Stage C is due on Tuesday this week**
 - 1 scenario, 3 acceptance criteria and a journey map
- **P3 replacement classes this week (Australia Day university holiday)**
 - Wed 26 Jan classes replaced with Thu/Fri classes
- **Stage D is due on Friday this week**
 - Low-fidelity prototype exploration
 - Low-fidelity application prototype
 - Theory-based justifications