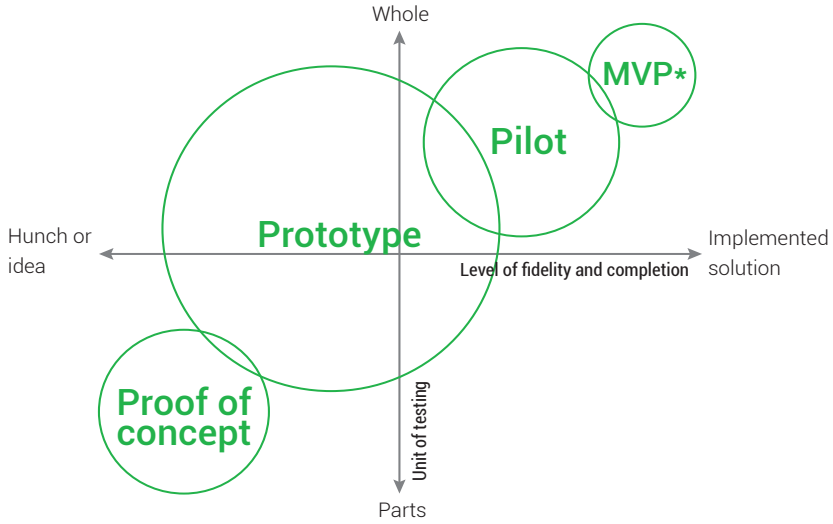
PROTOTYPE CONCEPT DEVELOPMENT

TOOL

What is rapid prototyping?

Rapid prototyping helps us to test aspects of a potential new solution in the real world in order to learn how the users of this new solution might respond. In the SLab, the prototype should also be a fractal of a larger, systemic intervention in the whole system even though it is intervening in a small part. Prototyping can test whether or not the intervention is likely to shift the larger system that we're working on changing.

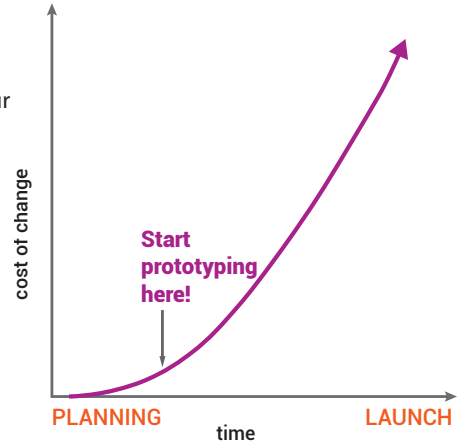
Rapid prototyping is a way to actively learn through small experiments. It allows us to make small bets, to take intelligent risks, and to use micro-failures as learning in order to make much smarter and informed decisions when it comes time to scale a solution. Prototyping also makes it much easier and clearer about when to make a major pivot in an approach when experimentation shows that the solutions that we've conceived of are not going to solve the problem that we're working on. Prototyping is different than a pilot or a program. It is small, quick, low risk, and cheap and has a very low investment of time and resources. Prototypes are about building to think; they aren't precious, and help us to move quickly and not fall in love with our ideas before we know if they are actually good ones.



** MVP=minimum viable product. Image adapted from Nesta.*

When to use this tool:

- To frame and test questions about the functionality, efficacy, viability, desirability, feasibility, and impact of your potential solution
- To reveal questions that need to be answered
- To deepen your understanding of the challenge space and users.
- To quickly and cheaply test and refine ideas or solutions.
- To inspire (yourself, and others) by showing your vision
- To test assumptions
- To engage users/stakeholders



How it works:

Once you've decided on a promising idea for a solution, you'll need to answer the question of "how is this going to work?" Visual and experiential tools like storyboarding, role playing, making a model, and journey mapping can help you to visualise your solutions in action. Once you've done this, there will be several touchpoints - or places where the solution interacts with real people in the real world - that emerge. These touchpoints are the foundation for your prototyping work.

These questions can help you prioritise what to prototype first, remembering that you'll likely go through several/many cycles of prototyping as you develop your solution:

- What are the assumptions embedded in your solution concept that need to be tested?
- Where is the place where you have the biggest question? The linchpin in your concept? The place that needs to be understood first, before the other parts can be tested?
- What do you want to learn from your prototype? What other questions do you have?
- What parts of your solution need further thinking, insight, and exploration in order to be developed?
- What parts of your solution need to be communicated and interacted with in order to progress further through receiving feedback?

How to prototype

Once you've decided what part of your solution you are going to prototype, you then need to figure out how to do it. It's better to start quickly and get your concept out into the world than to spend too much time thinking about it and planning! The sooner you start with your user testing, the more quickly you'll be able to iterate your solution as you're definitely not going to get it right the first time.

What is the main purpose of your prototype?

- Exploring and making ideas tangible?
- Testing and evaluating how well your solution works?
- Making the idea accessible to others through how it is being communicated?
- Something else?

What is the main focus of your prototype?

- To figure out its potential value?
- To figure out what role it plays in the problem that you are working on solving?
- To understand how the solution looks or feels through an experience?
- To test functionality and how different aspects might work?
- To test viability?
- To build understanding about implementation?
- Something else?

How are you going to make it real?

- There are many different ways to do this, but make sure that you are creating an experience (not having a theoretical or hypothetical meeting). Create a visual, role play an experience, imagine a future possibility, and use your creativity to build your prototype.

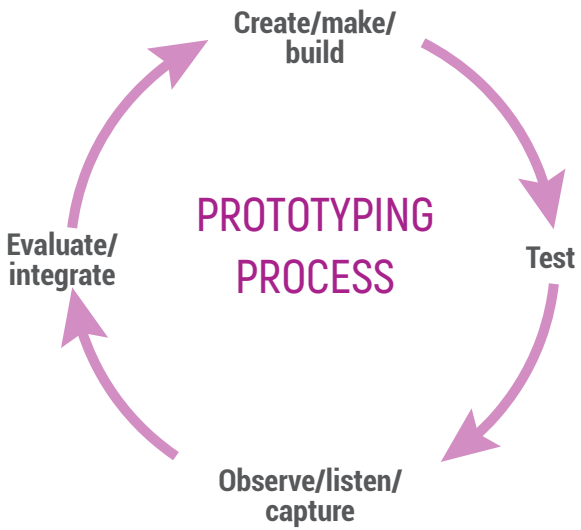
Who to prototype with

- Who do you most need to hear from in order to understand if your solution concept is going to solve the problem in a way that matters to them?
- Who holds the greatest experiential insight? Will you be able to access that insight?
- What are the ethical considerations in your user testing? How will you earn and maintain consent? How will you honour people's time? How will be accountable to what they are sharing with you?

Evaluating your prototype

When you are at the earlier stage of choosing what to prototype, you should also be thinking about evaluation. Prototyping is most successful when your goals are clear, specific, and focused. Here are some considerations:

- List a small number of key questions you want to answer, or key information you want to gather.
- Figure out who will be taking an evaluative perspective during the prototype testing activities.
- Develop some easy ways to document the feedback you are receiving - going back to the "think - feel - say - do" grid from the research phase could be useful. Perhaps there are some visual documentation methods you'd like to use (i.e. video, photos, participants making visual representations of their feedback).
- Don't get attached to your idea as this may blind you to the feedback that you are witnessing.
- Take a developmental approach asking what, so what, and now what questions in order to iterate and improve your solution.



More info:

- Sarah Hay - Slow and Steady Design
- Moura Quayle - UBC School of Public Policy + Global Affairs
- States of Change
- IDEO - Field Guide to Human-Centred Design
- This is Service Design Thinking and Doing