



Outcome-driven lean experimentation in Discovery and Alpha

~~A tutorial~~ An experience



Ryan Haney | Happygeneralist LTD
ryan@happygeneralist.com

Generalist but I
very much respect
specialists and other
role

Thankful to play
in this playground

Fine with
project phases

Agile prime
directive

I ♥ civil servants

Fine with short
term gov projects



DISCLAIMER

If you're not working to
outcomes, your project is
out of control



Why take this approach?

Do effective productive work, simplify product discovery, remove process as a barrier

Outcomes

Outcomes are the result of impact you actually achieve (positive change) by creating something (output). It is a placeholder & goal for the actual added value that a product or service creates

- This allows me to know if my work is effective or not
- Unlocks an ability to be more creative in how we solve problems
- Aligns teams making the work easier to do and more energising

Lean experimentation

Lean experimentation is a systematic, scientific process for testing and validating hypotheses about products, services, and features using the least amount of time and resources possible*

- Creates a stronger evidence base for making recommendations and decisions
- Is more collaborative, allowing ideas to come in from all areas
- Focuses on important learning and challenging key assumptions

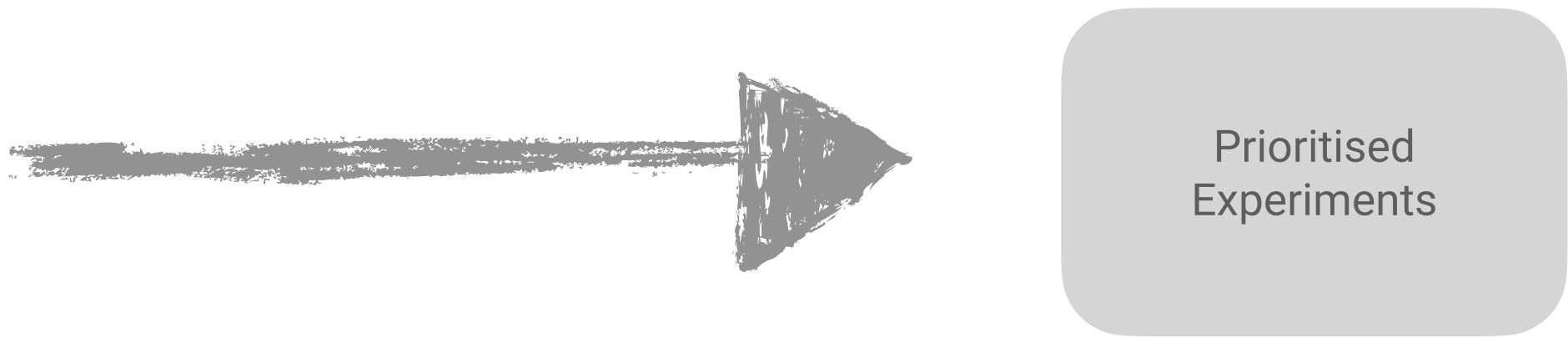
*<https://userpilot.com/blog/lean-experimentation/>

Behaviour design

Behaviour design is the practice of using insights from psychology to design products, services, and environments that nudge people towards desired actions

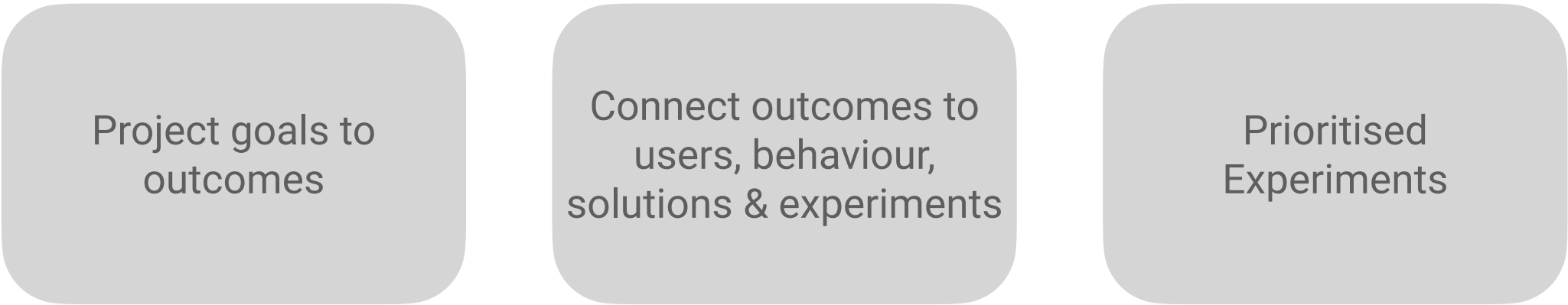
- The only way to create real outcomes is to use behavioural design
- Most all policy is trying to change behaviour
- Behaviour is already the backbone of good user-centred design (e.g. personas)

What did I do?



Started working towards an MVP for tools, knowledge and processes to make this work simple and painless

What did I do?

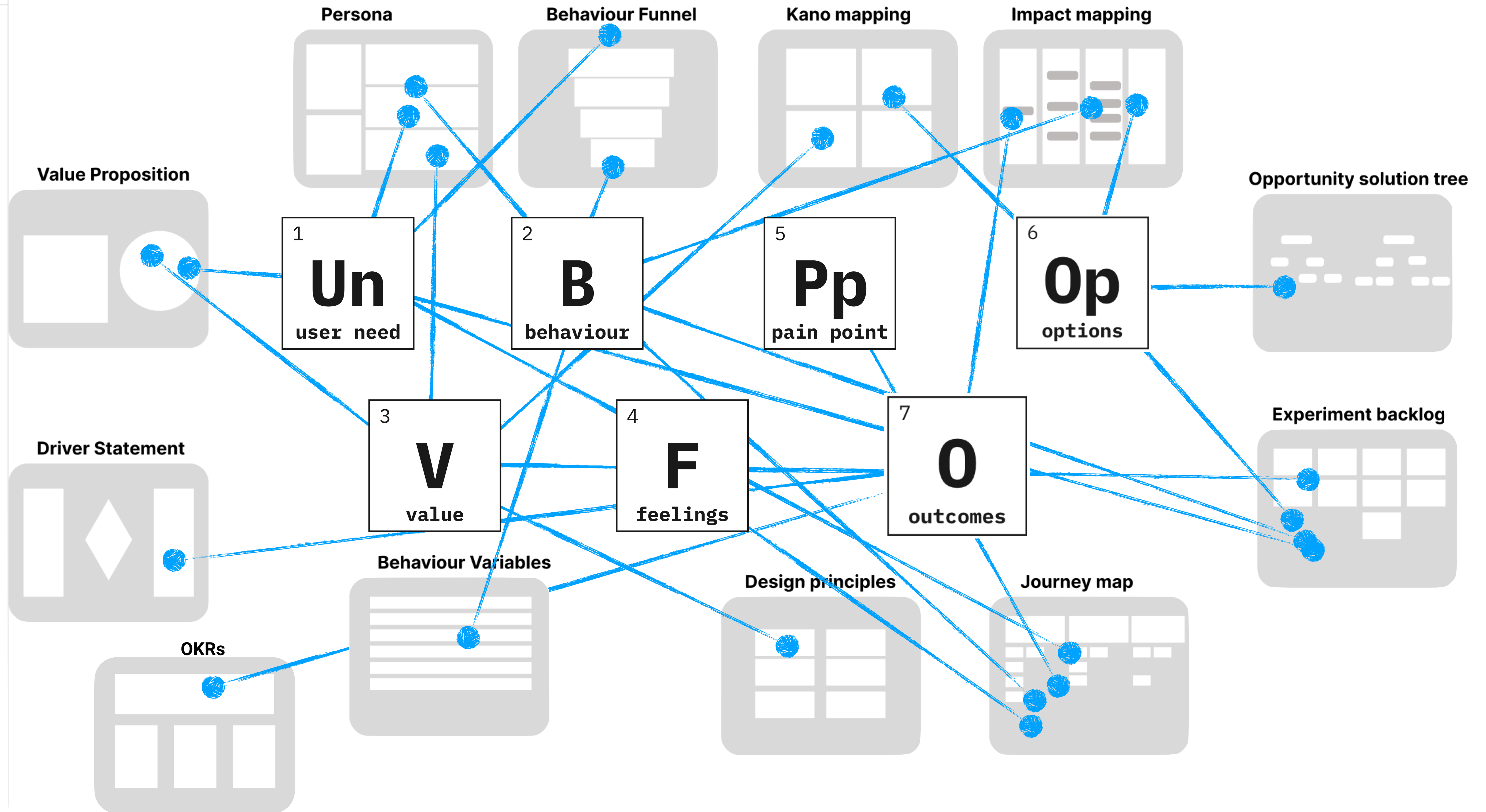


Started working towards an MVP for tools, knowledge and processes to make this work simple and painless

What did I do?



Started working towards an MVP for tools, knowledge and processes to make this work simple and painless



Project overview

- Introduction
- Requirements Engineering
- Software Development Process
- Testing
- Deployment
- Maintenance
- Conclusion

Team

Problem space



Problem solution fit

Actor / User information

Solutions, options and prioritisation

_Maps

Service landscape

Teacher services map | DfE

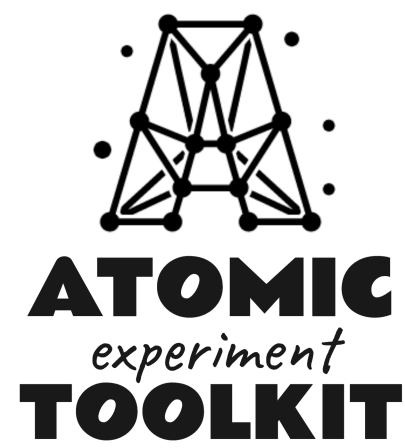
GIT & BAT Service map | DFE

Additional information

Glossary

Personas

Evidence



Atomic experiment toolkit is a
Rapid experiment toolkit for outcome based,
behavioural lean product & service design

- Align a team on an Experiment backlog in <1 week
- Elevate key shared information for core teams
- Reduce friction in updating important tools & design information
- Stop dragging tools all over Miro, picking tools up and forgetting them

Design principles

Grab & go

It should be atomic in its information. It should be easy to disconnect different pieces to focus them and then bring them back together

Easy update

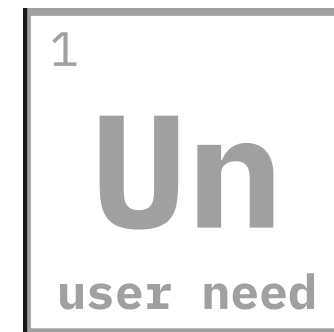
A usable kit is one that's easy to update and revisit. It requires the least amount of effort to find and learn from areas and reduces the need to update multiple things

Make connections visible

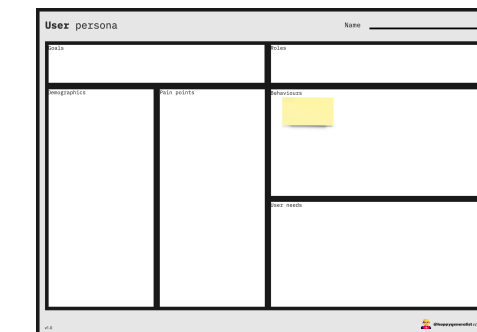
Everything is connected! That should be easy to see and highly visible

Just use it

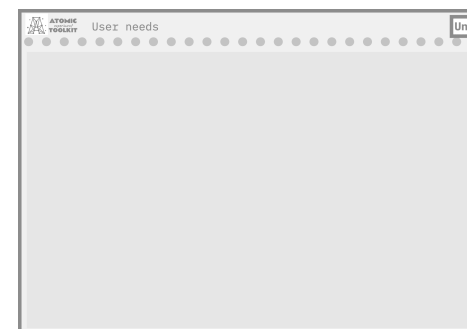
Everything should be obvious to use and quick to learn and facilitate. By having additional information and help in place there is no need to look elsewhere



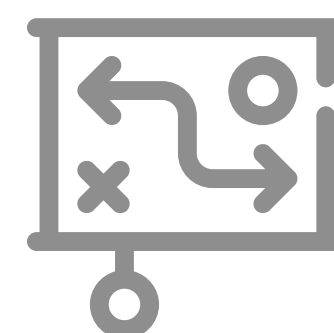
7 Key elements
and information



22 Key elements
and information



17 post-it posters

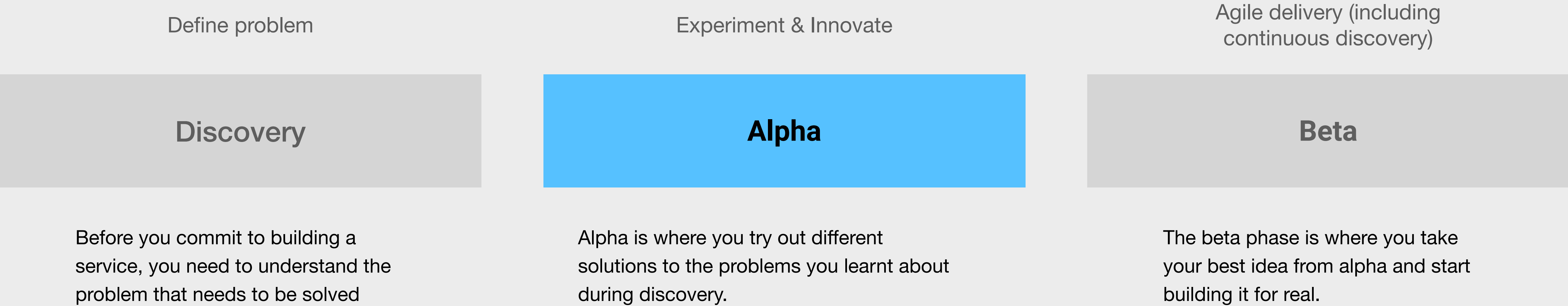


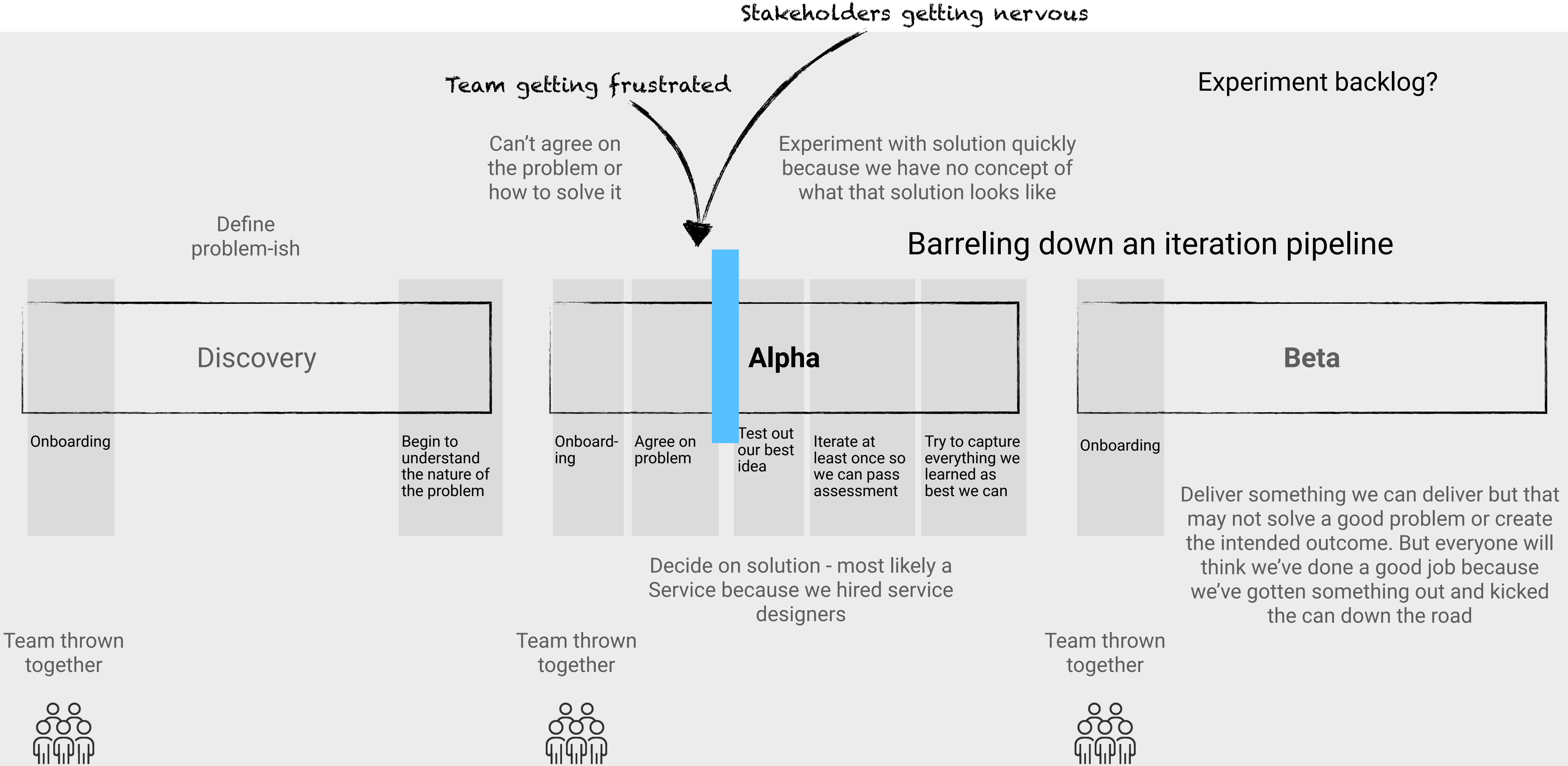
Training
workshops

Additional information

- Usage info
- Patterns of usage
- Tips
- Element onboarding information
- Roles and responsibilities





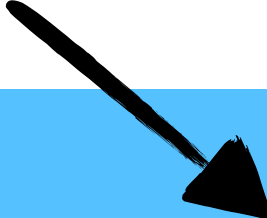


**How do we shift from a small
window to a continued process
of innovation?**

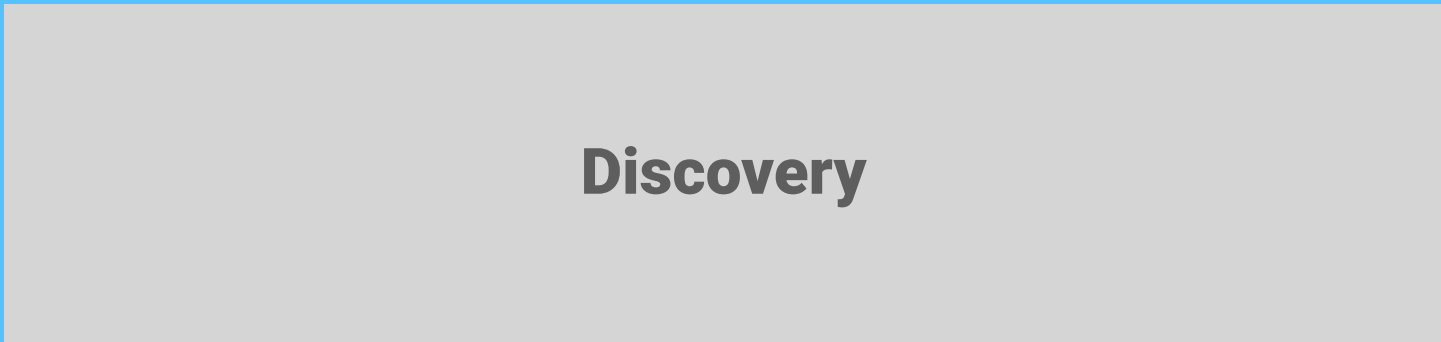
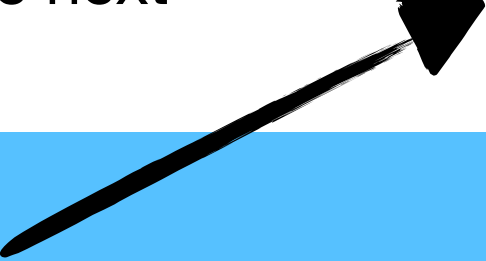
Can we shift departments to innovation machines achieving outcomes, reducing stress churn and slowly build?

My vision for outcome experimentation & innovation

Bringing in insights
from previous efforts

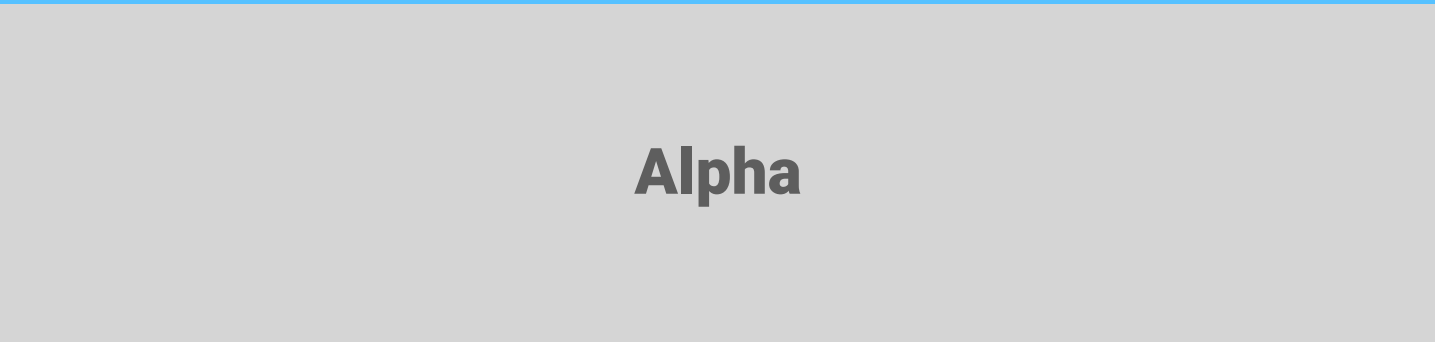


Handing off the work to
the next



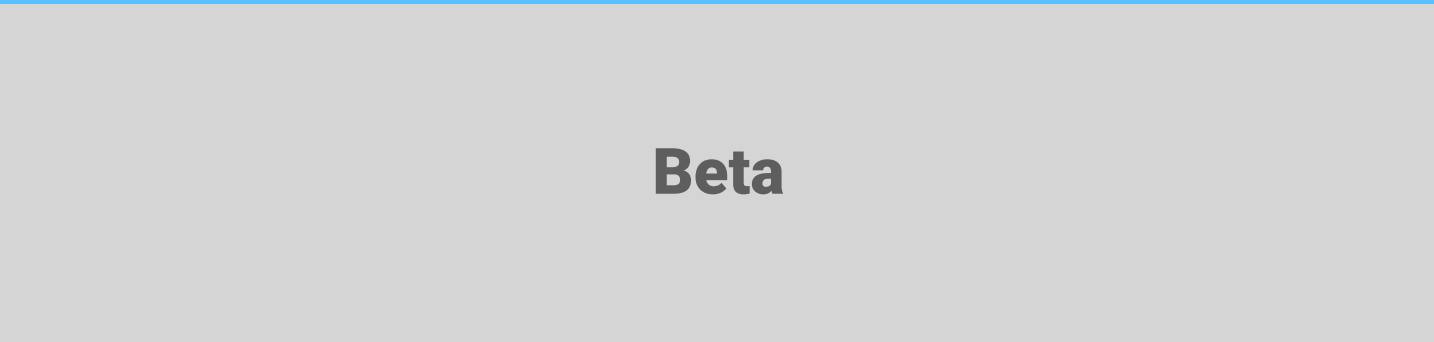
Discovery

Prototype, experiment, build things



Alpha

Prototype, experiment, build things



Beta

Prototype, experiment, build things

Why doesn't this work?

Problem statements from departments are often set and yet not suitable to design for outcomes/impact	GDS is a standardised structure, this reduces variability but it constrains innovation	Learning new tools is hard for teams as they vary in skill, experience and knowledge	Everyone is speaking a different language (e.g. Jobs to be done, user stories, user needs)	Rapid delivery constrains good problem framing and innovative design	There are (too) many different ways to approach innovation	Testing things that didn't really matter
Contractors bring in tools that help serve a purpose but often get dropped and forgotten about	The team is making the process in which they are working up as they go	Evidence created by discoveries is very weak with a large amount of confirmation bias	Learn and iterate culture is arrogant and toxic and doesn't necessarily solve problems (you can't iterate your way out of a prison/grave)	Teams are trying to learn too much and too many different things as they wayfind	Agreement of the processes being used is often shallow and can fall apart very easily	Not enough time to experiment Taking too long to agree on process, problem to solve etc
Stakeholders often are solution oriented and can be frustrated by an unknown process and no understanding of how their ideas fit	Expensive solutions get built on weak evidence	Miro boards become unusable with not enough time (or value) to organise them	User research is just a dump of information that is too much to create good learning	Key learnings get lost and half remembered during the write-up	Forming a team, rapidly learning and finding direction is exhausting (too much context switching)	Cargo cult experimentation (not all hypotheses are created equal)
Focus in Discovery and Alpha is constantly being battled	We stay away from solutions because we fear stakeholders getting too attached	Learning is ad hoc and not always tied to specific deliverables	Decision-making is often not considered or not explicit in the process	User researchers unable to code qualitative data	User researchers not knowing what user needs are	No set agreed upon things to learn
Tools seem disconnected and it's very difficult to follow a red thread through them all	People equate agile and lean with low rigour or no documentation even though this isn't true	Identifying problem type , quickly is difficult because of obfuscation by the org and limited view of information	Implementing policy is driving poor outcomes	No clear chain of evidence to make decisions	Problem statements aren't clear	Chaotic work - swimming in tools, frameworks and learning
Time to make decisions is limited	This work is extremely difficult in optimal circumstances	False starts create a lot of exhaustion	Learn something very important but it doesn't get documented	At a basic level a lot of UCD people don't understand core concepts	People see experiments as solutions, not paths to solutions	

And many more...



USER NEEDS

Foundational

User needs are the starting point and if they are wrong or absent then you're building on a house of cards

Poorly understood

There isn't a lot of information out there about how to create good user needs, yet it's a prolific tool in departments

Incorrectly used

Many, many people incorrectly write user needs. The majority of user needs I see in departments are poorly written or unusable

Stakeholder implementation

Sometimes projects are pushed through without a user need, it's not the end of the world but we have a responsibility to reverse engineer this and build evidence

Tech innovations, unanchored

If you're talking about solutions like Machine Learning and you aren't starting from user needs you have already failed

Solutions wag the tail

Solutions are and should be neutral but without a strong understanding of the user need solutions can be pushed without justification

**If your user needs
are wrong you don't
achieve your
outcome**

You **need** User Needs

UX Scotland 2023



Ryan Haney
happygeneralist



<https://drive.google.com/file/d/15X-BtrNIEad7BoDlyqc0xuoNC6E5EBPo/view?usp=sharing>



ABSENCE OF DESIGN & PRODUCT STRATEGY

Things that change: Policy, Intent, focus, organisational value

Things that don't change: User value

How do we stop this from happening?

Design

Business requests

Market forces

Changing strategy

Technical developments

Strategy evaluation scorecard

Scale 0-3
From (not true at all) to (very true)

We have mapped out our entire service landscape, including hidden or un-designed services

We have a plan to systematically improve user needs across all of our journeys

We have a clear and agreed upon way to prioritise service, product and user need improvements

Our research and evidence for user needs is integrated and updated

We have a baseline of health for all our services and products

All of our projects are working on outcomes connected to our design strategy

The information about our users, services, products and journeys is easily accessible by everyone in the org

It is easy for anyone to add new evidence about our users and their needs

We're honest about when policy and user needs don't line up

Total Score:

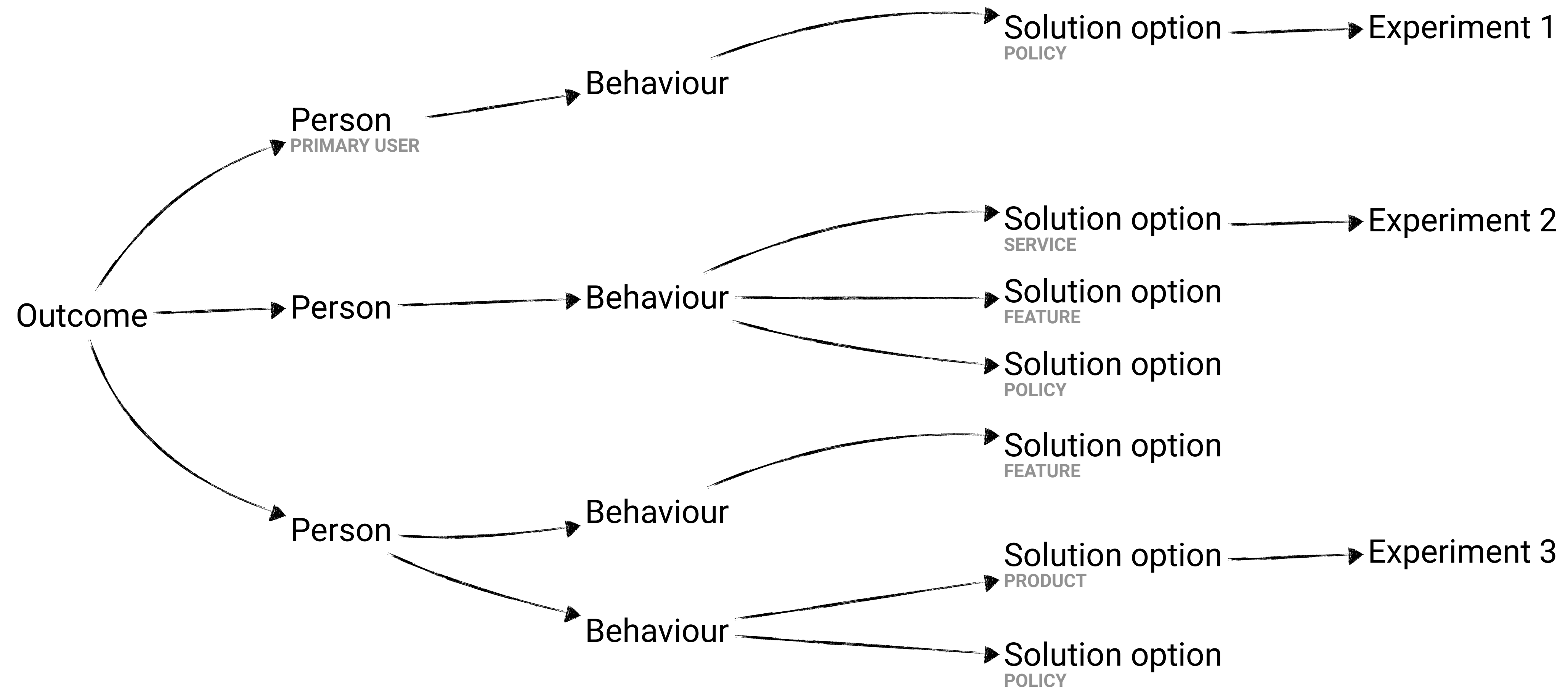
Max 27

An abstract, low-poly geometric landscape featuring several mountain-like structures in shades of teal, orange, and yellow. The mountains are set against a background of vibrant, overlapping geometric shapes in purple, pink, and orange. The overall style is modern and digital.

USER NEED $\neq$ SERVICE

SERVICE $\neq$ OUTCOME

Solutions need to be diverse & blended



The path to solutions is not singular

Incremental Innovation

Moving a single product or service up the hill to a higher point, mainly driven by human-centred design (Norman/Verganti)

Product/Service/Process Innovation

Moving from one hill to another, generally from a smaller hill to a larger hill

Value proposition innovation

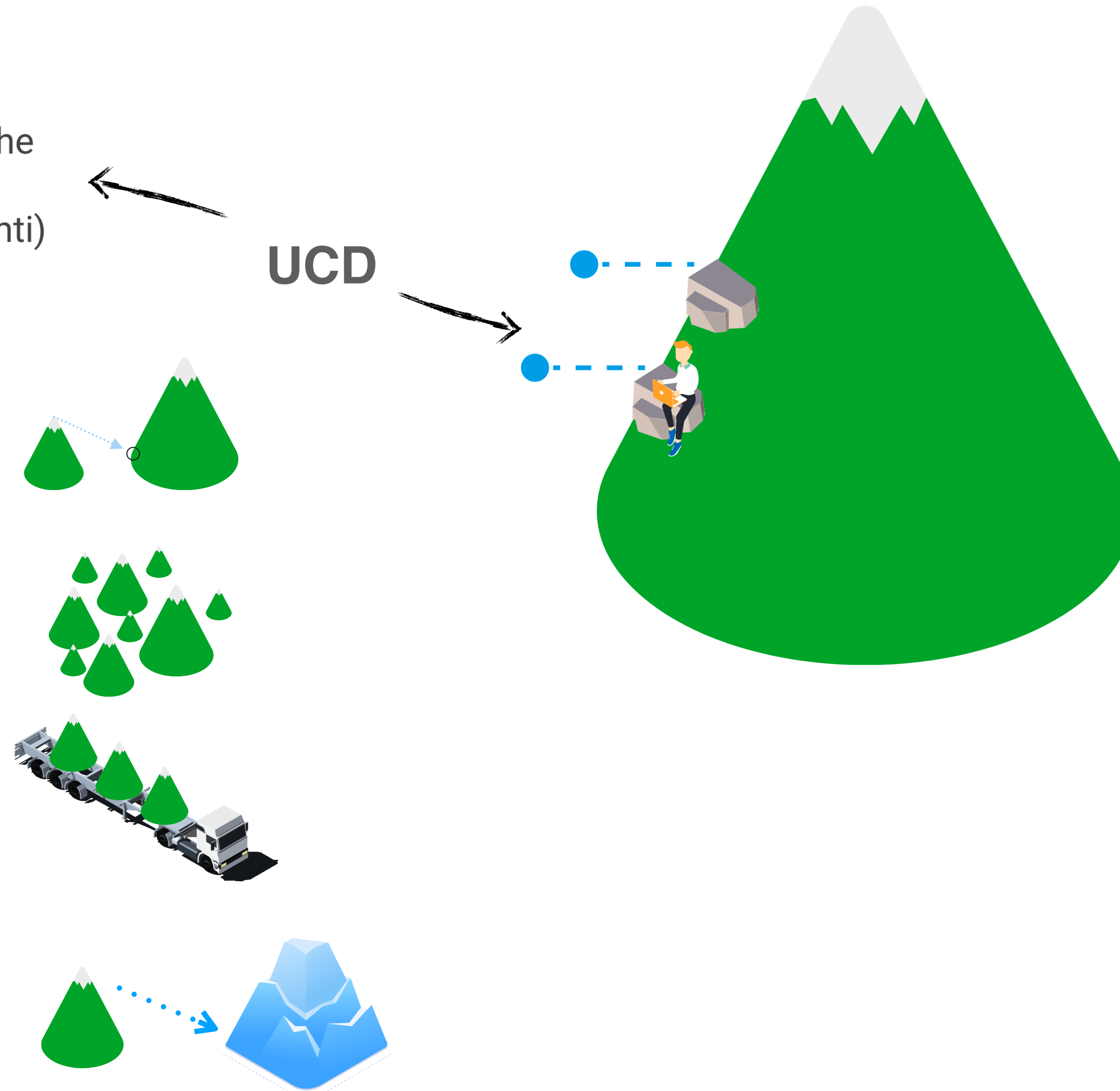
Matching the desire for particular features, products or services to user needs. This can be a set of hills, not just a singular hill

Business model Innovation

Finding a better way to deliver or package an innovation

Radical Innovation

Moving to a radically different type of hill representing a paradigm shift by combining a technological change (ability) with a design change (meaning)



What does your kit look like?