



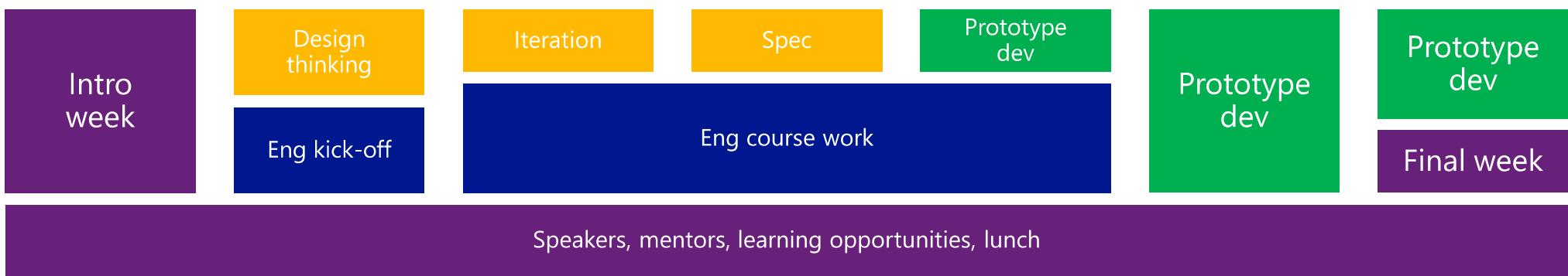
# Introduction to PM and Design Thinking



# Agenda

- Course overview
- What is a PM
- Design sprint overview
- MVP
- Agile engineering

# Program sequence



This is a high-level overview of the next six weeks. We'll go through a product design sprint and come up with an app idea. Meanwhile you'll learn the engineering skills to prototype the app. In the second half you'll begin building the prototype as a team. The final week you'll pitch your app, demo the prototype, and present on your journey as individuals and teams during the program.

# What is a PM?

# What is a PM?

- Responsible for defining the why, when, and what
- At the intersection of user experience, business, and technology

Program Managers (at Microsoft) determine what the most valuable thing we can do for our customers is and why we should do it.

# Names and responsibilities

- Program Manager – what it's called at Microsoft
- Product Manager – common name
- Project Manager – (bit different) in charge of planning and execution of a particular project

What a PM does varies by job and point in the product lifecycle. It can include:

## Framing

Vision  
Product Definition  
Planning  
Prioritization  
Customer identification  
Value proposition

## Defining

Communication and Storytelling  
Requirements specification  
Risk management  
Team building  
Customer and Partner engagement  
Passion and Clarity

## Delivering

Accountability  
Technical acumen  
Creative problem solving  
Agility  
Consistency  
Responsive design

# Design thinking

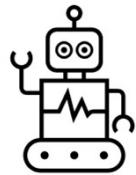
# What is design thinking?

The process for solving problems and creating solutions. It's rooted in understanding the needs of the customer.

It's not about the technology, it's about people. It's user-centric or human centric.

It answers a business question through design, prototyping, and testing ideas with users.

# Tech-centric vs Human-centric: example



## Tech

We need an app that's a digital front door to our health services; allowing users to book appointments, view their charts, navigate buildings with maps and pay their bills.



## Human

Patients need to be able to manage their healthcare from their phone and easily access critical services in order to improve health outcomes.

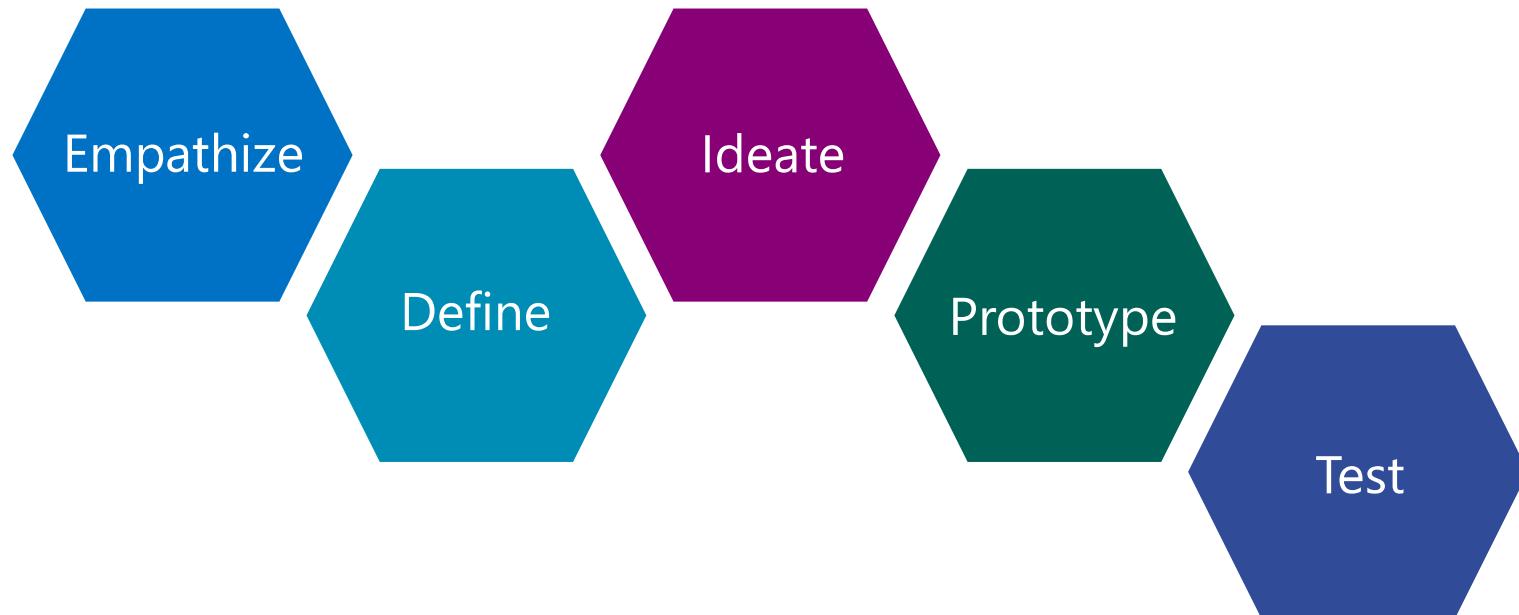
# Three principles

These principles are the operating model, they hold true through all stages of design thinking:

1. Empathy – understanding the needs of the user, representing the user not yourself
2. Ideation – pushing to explore the many possibilities, reaching for creativity and innovation
3. Experimentation – iteratively prototype, build, and ship, continuously learning and applying feedback

# Phases of design thinking

These are the five-phases of design thinking. We'll go through all of these phases to figure out what to build.

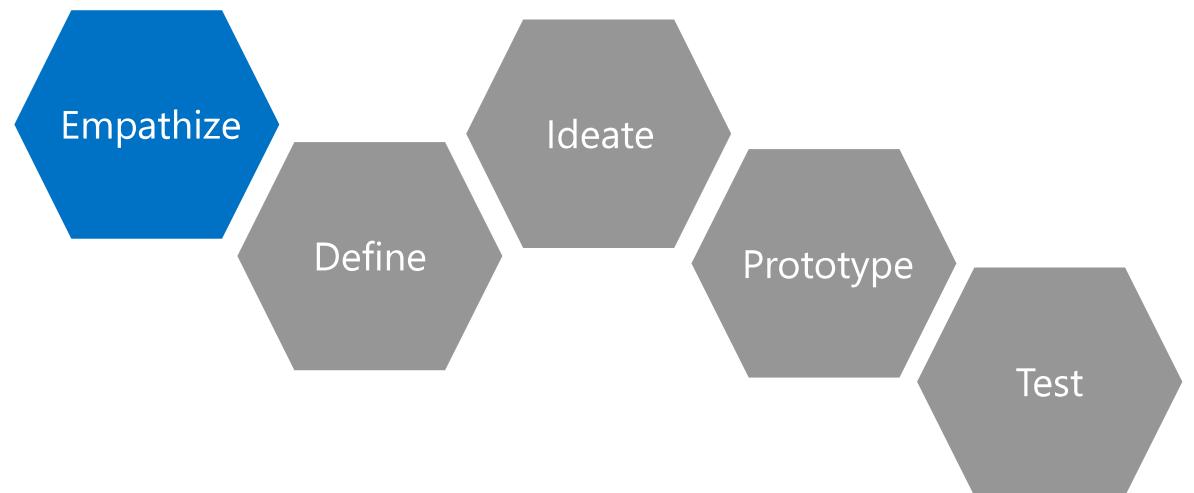


[Stanford's Design Thinking Model](#)

# Empathize

Develop a deep, empathetic understanding of user needs.

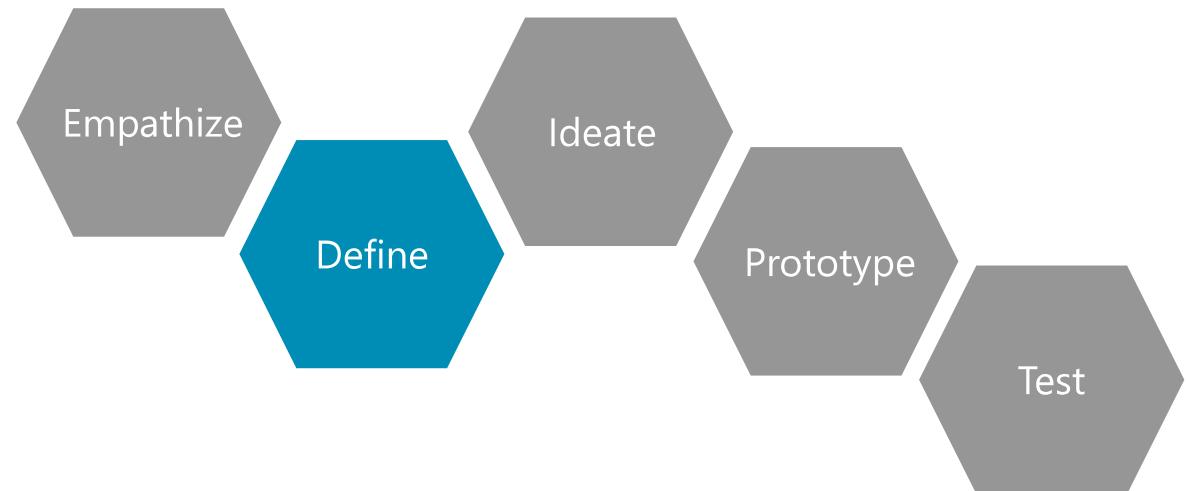
Observe what users do today. Engage in interviews and conversation to understand needs and objectives. Immerse in the experience of the user.



# Define

Use information gathered in the empathize phase to identify pain points and patterns, clearly defining the pattern.

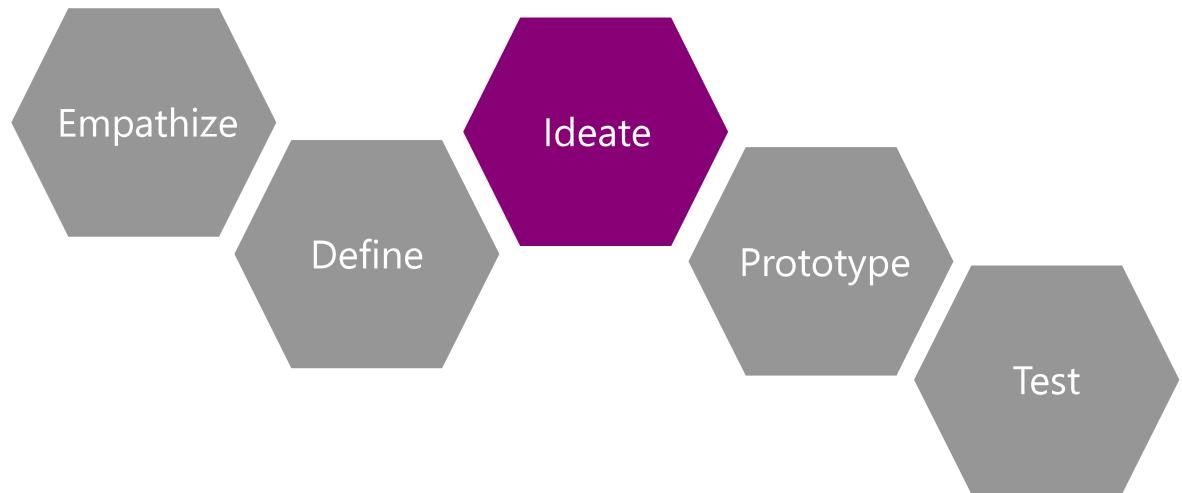
Create human centric statements agnostic of solution or technology to define the right challenge.



# Ideate

Generate ideas for the solution.

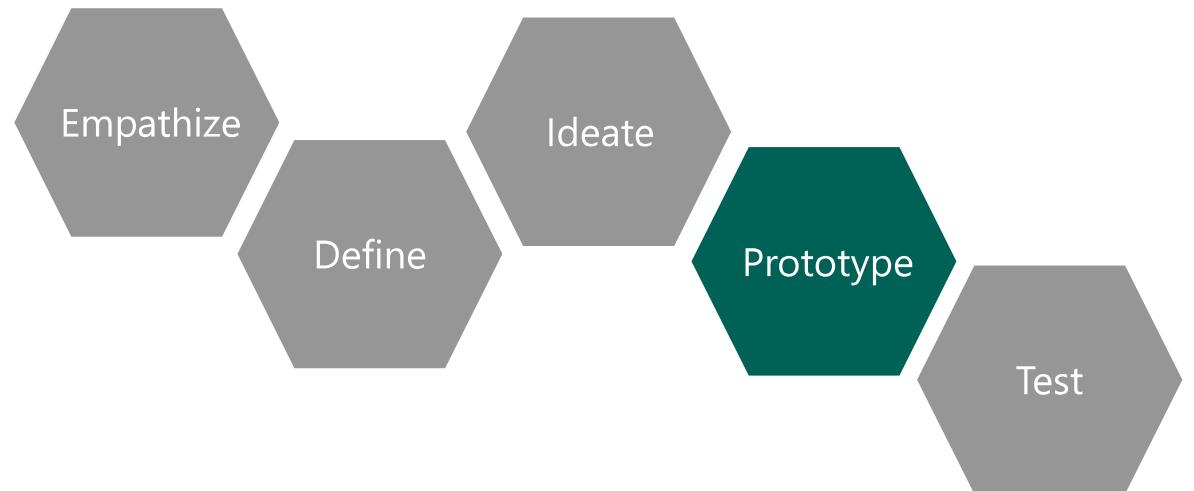
Go beyond obvious solutions, conduct exercises to expand thinking, and bring in diverse perspectives to go wide.



# Prototype

Experiment to narrow down the ideas by creating a sample or model to be tested.

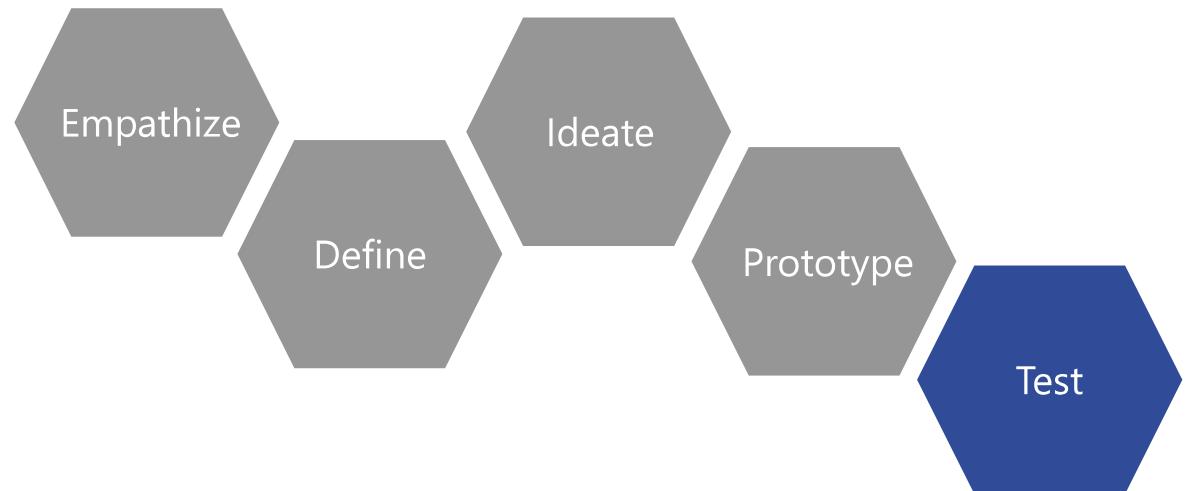
Build prototypes by drawing, using digital tools, or coding to help ideate, problem solve, start a conversation, fail quickly and test inexpensively.



# Test

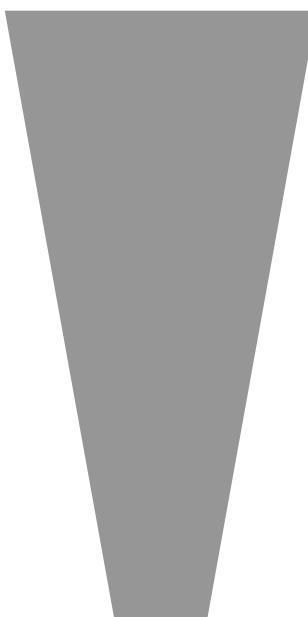
Get feedback on the experience from users to evaluate how well it meets the needs.

Conduct usability studies and have conversations with users to test your hypothesis, refine prototypes and learn more about users.

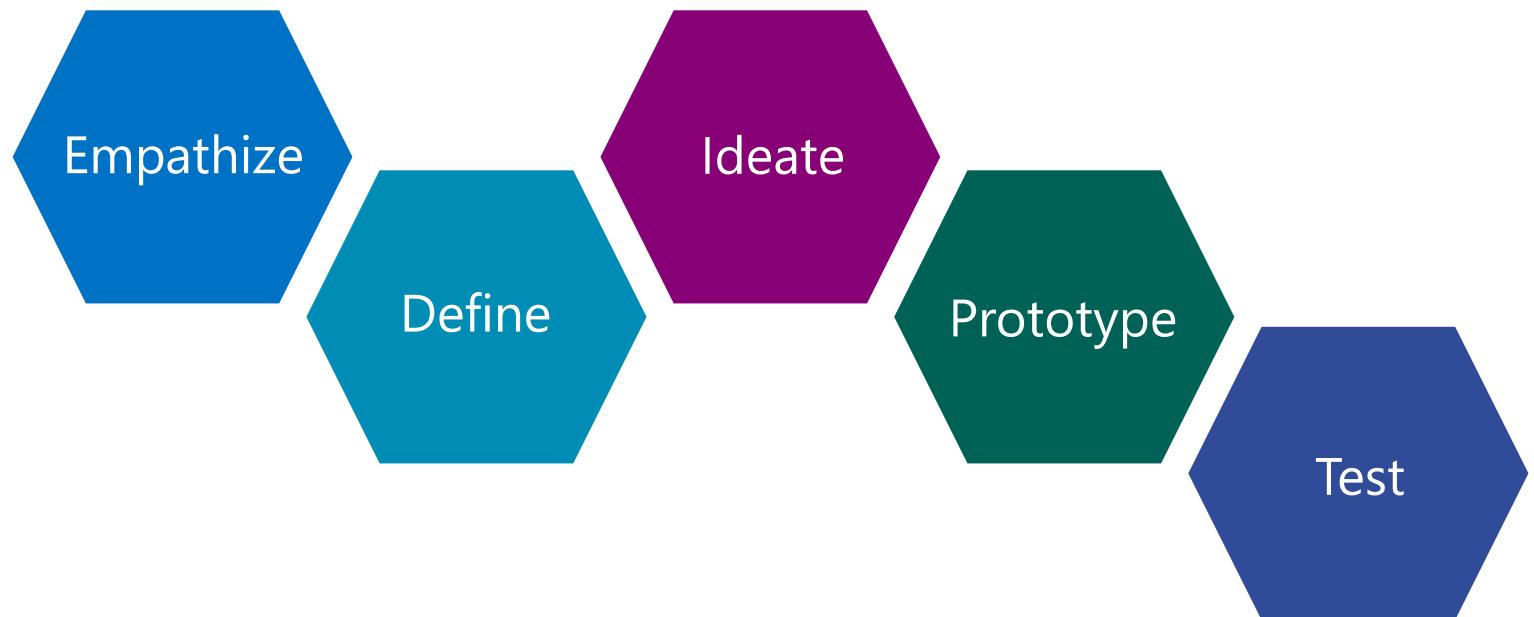


# Expand and refine

**Broad thinking**  
expand possibilities, creativity,  
innovation, stretch

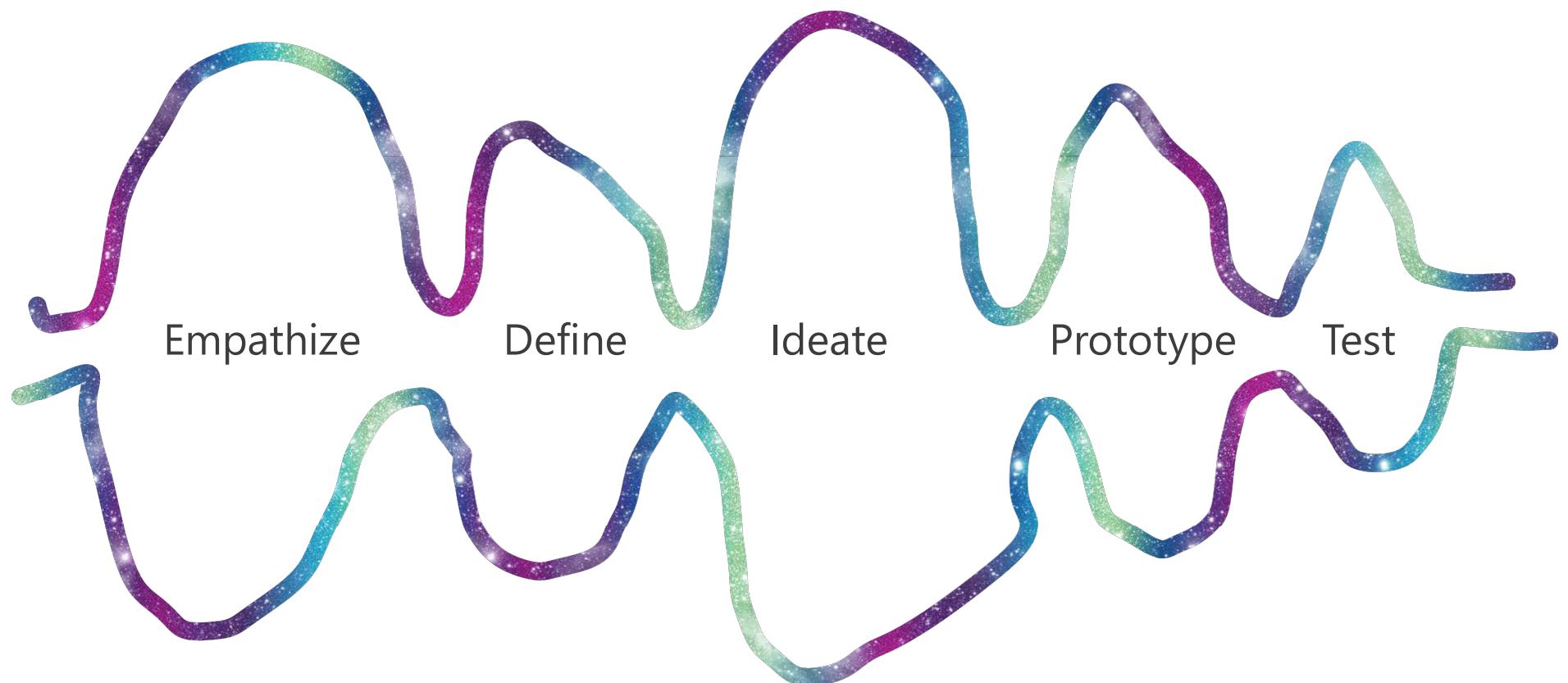


The design phases go through periods of expanding thinking and refining thinking.

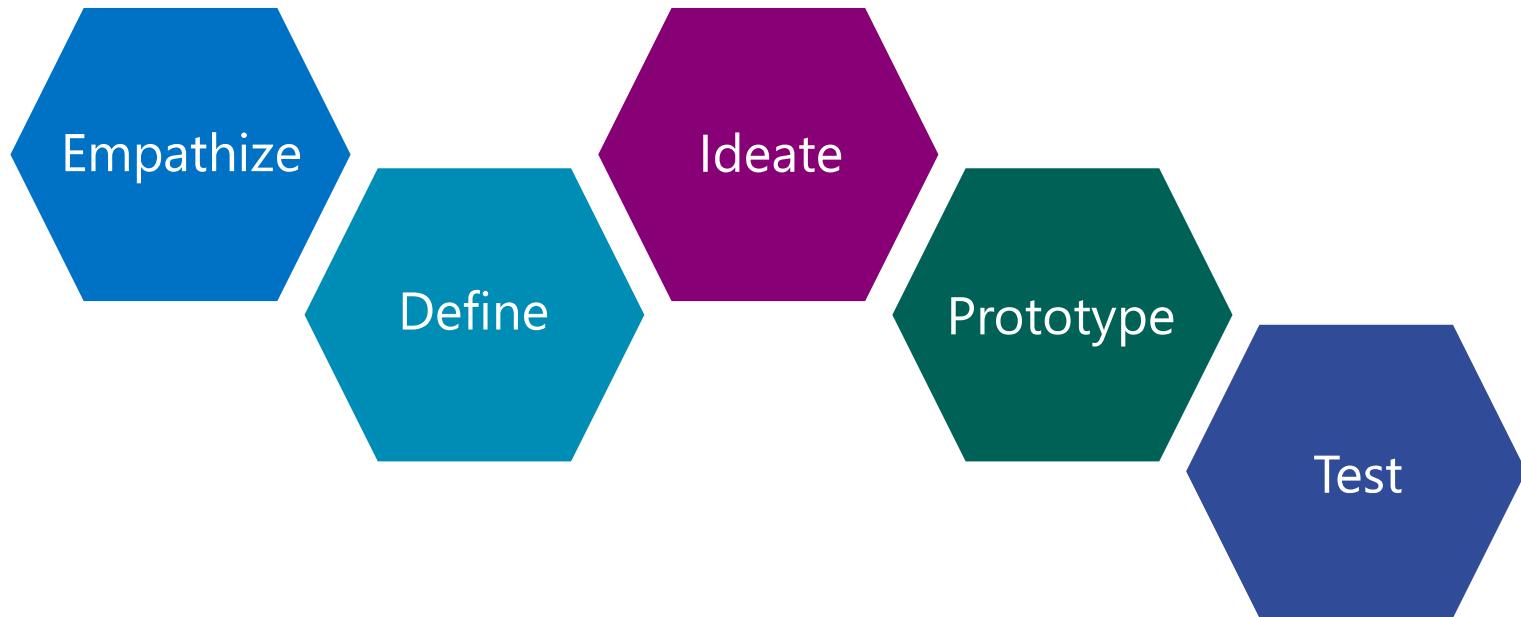


**Refine thinking**  
narrow possibilities, evaluate  
and reduce options

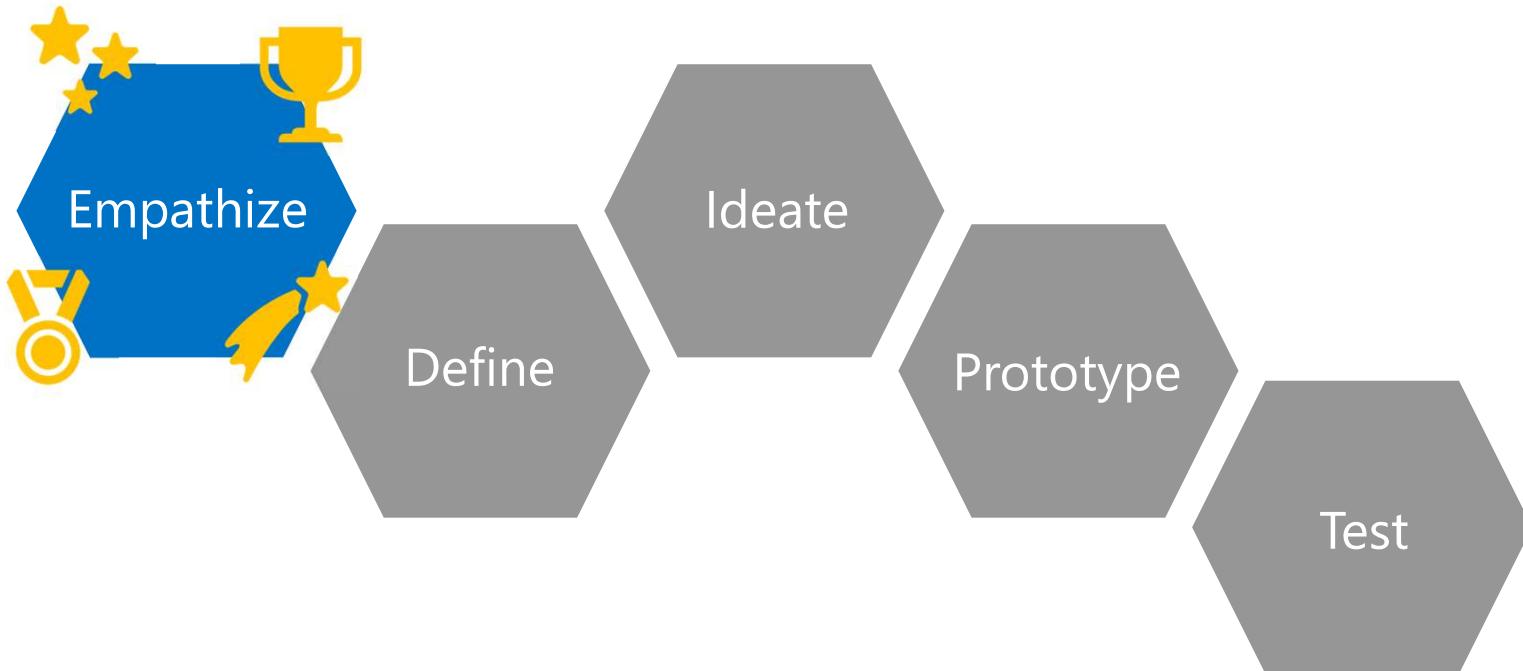
# Expand and refine



# Which stage is most important?



# Which stage is most important?



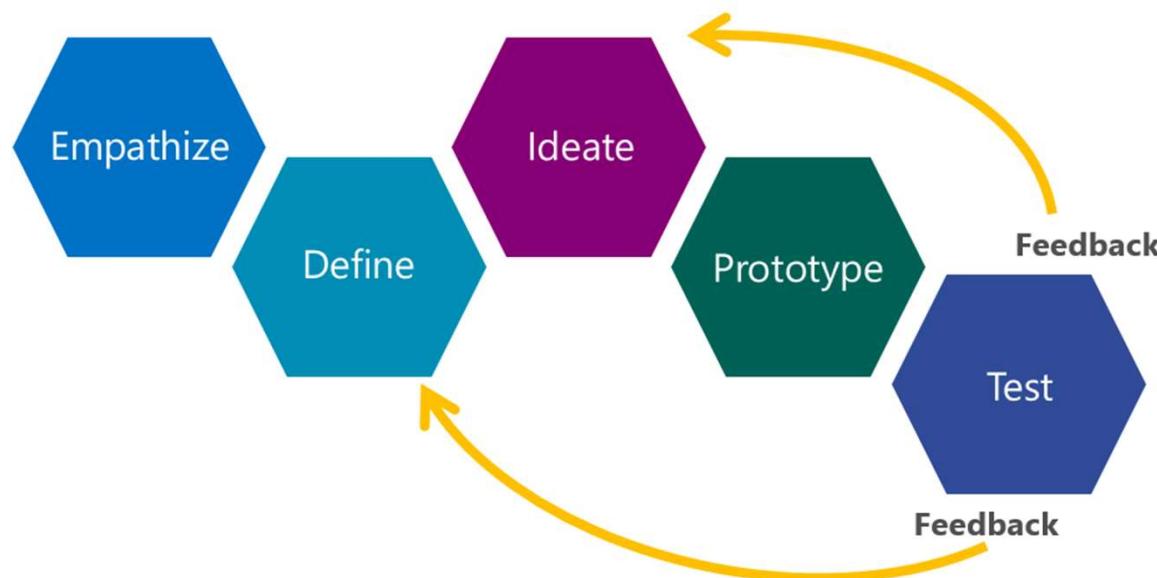
1<sup>st</sup> place: Empathize – find the right path to start on

2<sup>nd</sup> place: All of them together as a process

Without understanding the customer needs it's just a guess of what you should build.

# It's an iterative process

A cycle, feedback from testing with users can help refine the point of view, spur a creative process and improve what's being built. It's cheaper to change before development (though iterating happens there too). As you learn more it helps you get crisper on the problem you're trying to solve and how best to do it.



# MVP

# Minimal Viable Product

MVP is the first stage of releasing a product. It gets the tool in the hands of the users quickly. An MVP isn't perfect. It is good. Waiting for perfect someone else will get there first, attract and retain early adopters. It must have:

1. Value to attract early adopters
2. Value to retain users
3. Feedback loop to guide future development

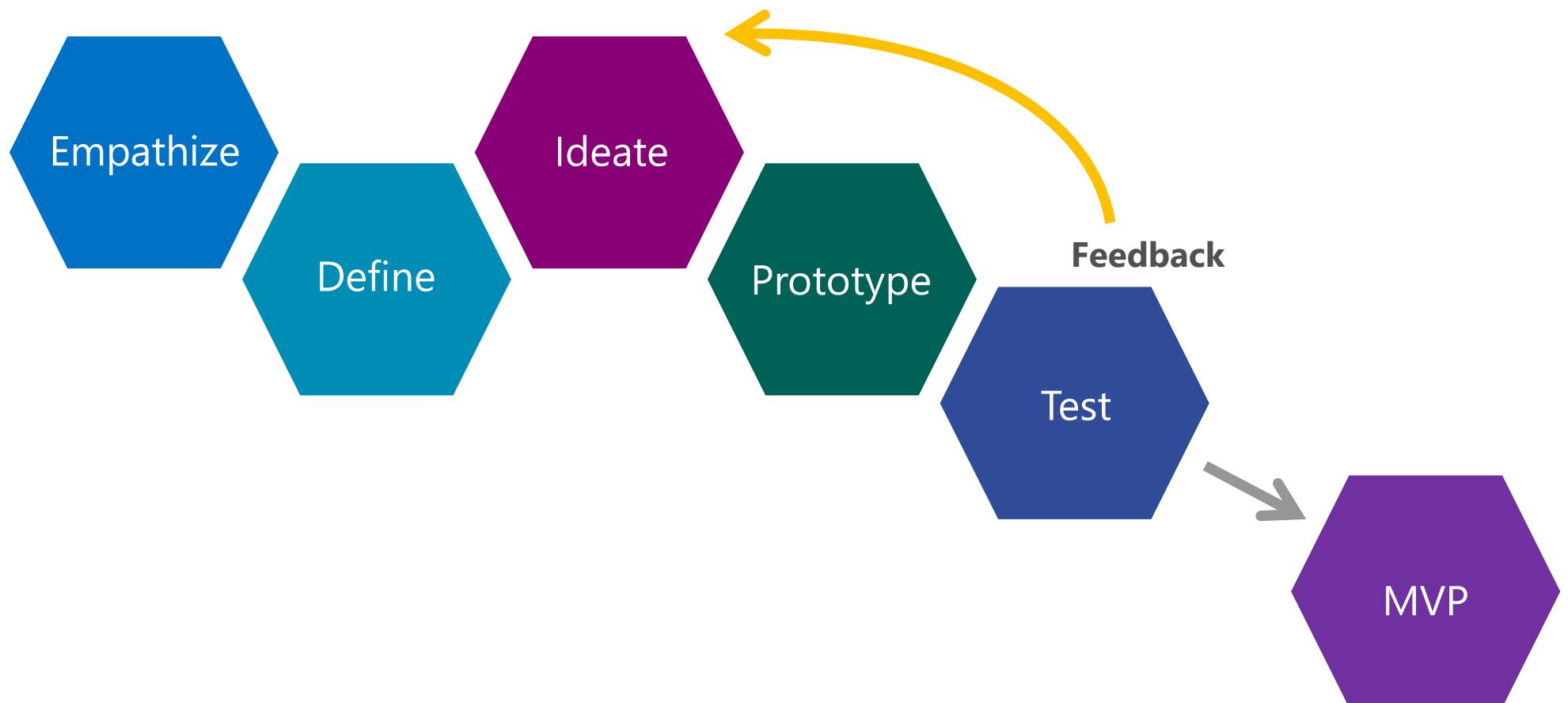
An MVP prototype is what your team is building together.

# How to determine the MVP?

The design thinking phases help find the MVP. Three key considerations:

1. Desirability: What does the user need?
2. Viability: What is the business case and market opportunity?
3. Feasibility: What is technically and financially feasible?

# MVP



# Design phases fit in with product lifecycle

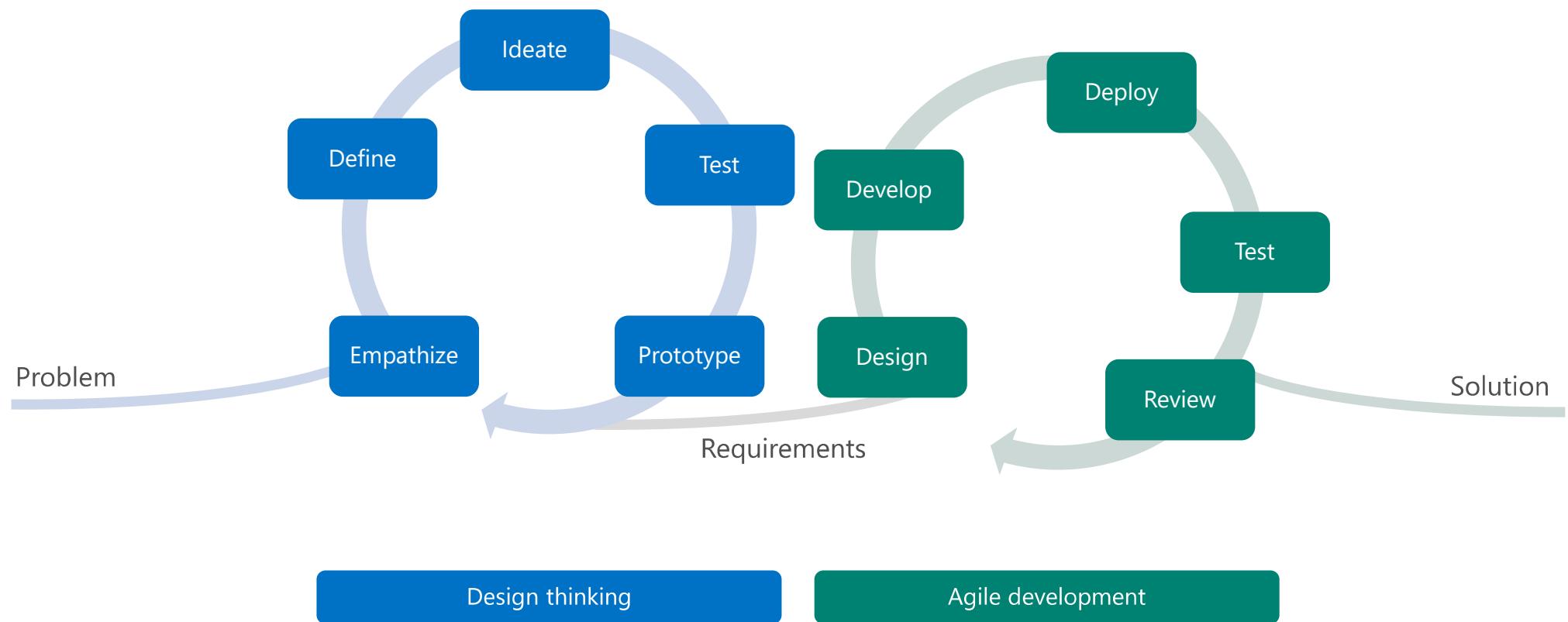
The product lifecycle is how things are built and delivered to customers. The design thinking process combines with the product lifecycle.

The product lifecycle is also iterative. Products aren't released one and done, they're updated, expanded, refreshed and sometimes deprecated based on what's learned from users.

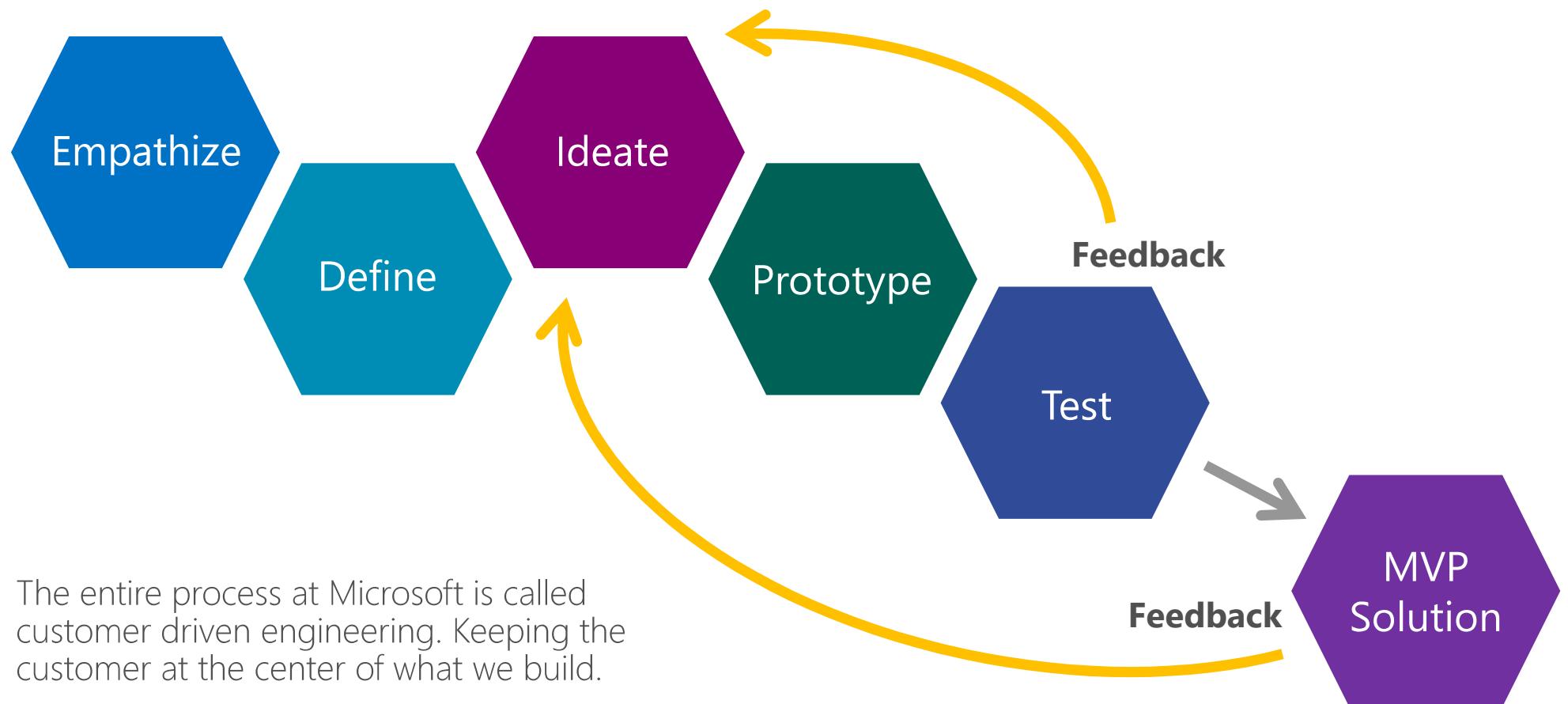
How we design and how we engineer are tied together.

# Iterative product lifecycle

Agile development – the process of building, shipping, and learning



# Customer driven engineering



# Agile engineering

# What is agile engineering?

Agile engineering is a process of iteratively building and releasing. Shipping units of customer stories, or features, based on user and business needs.

KM0

It allows a product to quickly react to opportunities, improving the user experience and keeping the product healthy.

Another method is waterfall. Waterfall is structured and rigged. It's design to mitigate risks and best suited for products that can't be delivered quickly or iteratively.

## Slide 30

---

**KM0**     [@Noel] is Waterfall opposite of agile? We should say that if this has to be a deck they read on their own  
Kristin McMeen, 2020-05-06T01:53:07.176

**NH0 0**   They're not quite opposites, they share some things. I'll call out that it's another different way of engineering  
Noel Hanzel, 2020-05-06T19:13:40.208

# Agile



Agile releases frequently and operates in short sprints, segments of work.

# Waterfall



Waterfall released at the end of the process. Each stage is clearly defined. There isn't much room for iteration after the project is released.

# How agile development works

- A feature crew determines what to build. A feature crew can include PM, designers, engineers, and UX researchers through the design phases.
- PM sets the requirements, design defines the user interface (UI), engineering figures out how to build it producing a spec (specification) and engineering plan.
- The feature crew scopes and prioritizes the work.
- The work is split into sprints, usually 2-week segments.
- Engineering builds, consulting the feature crew as needed.
- Everyone tests the product. Researchers define how to validate with customers.
- The feedback is applied and the product gets better.
- The product ships when ready!
- Learn from the customer and repeat.

## Slide 32

---

**KM0**     [@Noel] define UI

Kristin McMeen, 2020-05-06T01:53:30.781

**NH0 0**



Noel Hanzel, 2020-05-06T19:17:51.139