Design Thinking

Rules of Behavior: Do’s and Don’ts

Ben Sigman - September 2022

Who am I?

20+ Years in Tech

Why should you listen to me?

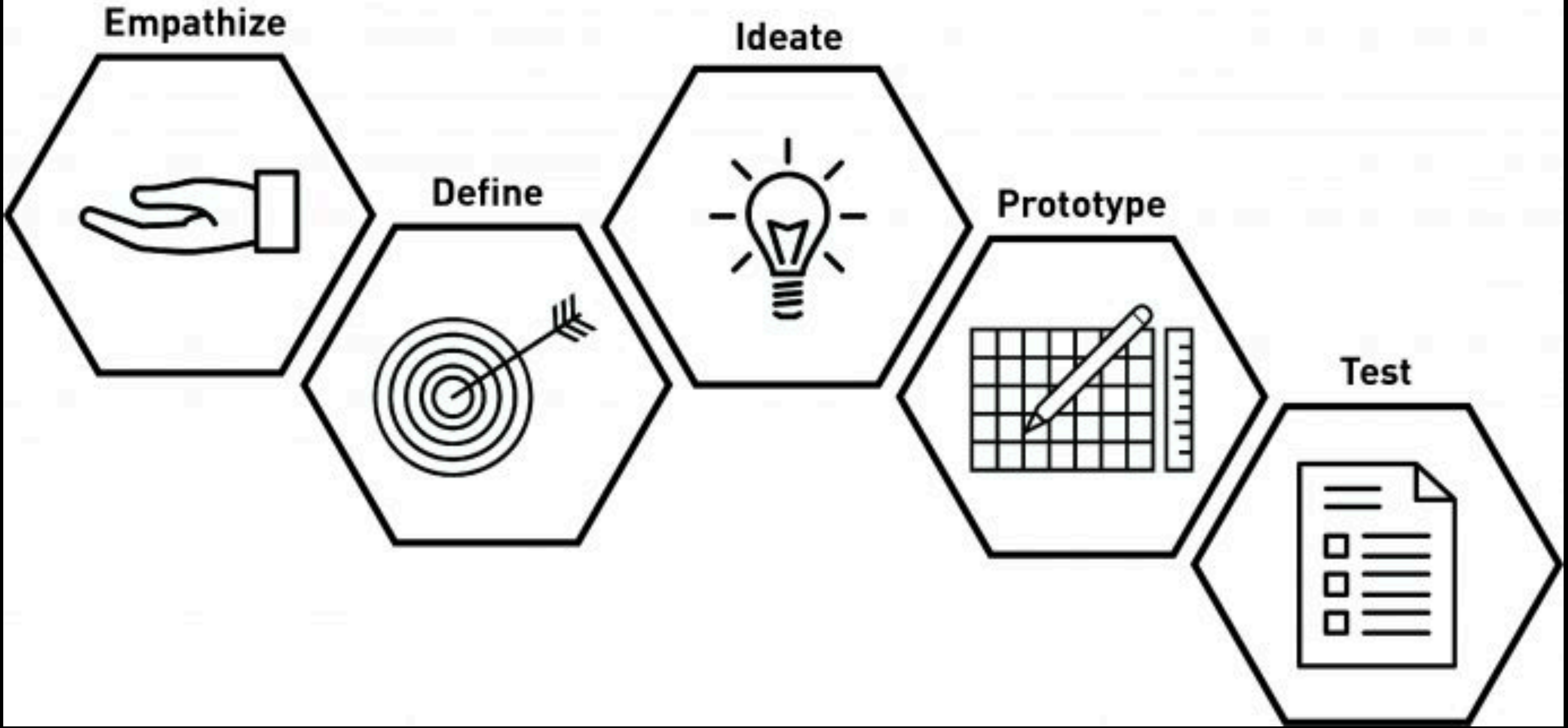
Fixing

Destroying

Building

What is design thinking?

Design Thinking



Empathize

Define

Ideate

Prototype

Test

What is design thinking?

A system for putting your customer / user at the center of your problem solving process.

Why should you care about design thinking?

Solving problems

Making Choices

Maintaining Creative Confidence

SPALDING
32

“You've got to be very careful if you don't know where you are going, because you might not get there.”

Do’s and Don’ts

Do : Define the User / Customer

- Who is this for?
- Why do they need it?
- Will they pay for it?
- Will they stick around if it doesn’t work 100% of the time?

Don’t : Build for Anyone

- Anyone = No One
- If you build too broadly, you will fail

Do : “Yes and” Brainstorming

- Collaborative brainstorm sessions
- Give everyone an opportunity to share ideas
- Create a safe space for sharing… just say yes and… see where that takes you

Don’t : Play “It Would Be Cool If…” forever

- It would be cool if is the ultimate creator and destroyer of products… if you keep playing that game, you will lose

Do : Make It Personal

- Best is if you are building a product that you will personally want to use. You will do much better if it’s personal

Don’t : Venture Into Unknown

- Don’t go and venture into building something you do not even understand yourself…

- Example: if you don’t know anything about food delivery, don’t build a food delivery app. If you don’t know anything about games, don’t build a game

Do : Start With a Clear Need

- The more narrow the better
- Squirrel - better portfolio tracking for people with multiple portfolios
- Bitcoin Libre - non-governmental bitcoin wallet for beginners in El Salvador

Don’t : Start With a Feature or Tech Stack

- Example: Add Crypto to WhatsApp
- Example: Add Blockchain to Shipping

Do : Define Success

- Really know what success looks like how many users? How much money? How many downloads?

Do : Start Simple

- Start with a simple clearly defined need, build the designs flush out EVERY detail before you begin building anything. This might take a week, but more likely it will take you 2-3 weeks full time… maybe even months

Don’t : Boil the Ocean

- If you want to build crypto chat, but you need to build the chat first - then you’ve already lost the game.
- Don’t try to build some thing that will take 6 months to develop… a good initial timeline is 2 months for ideation / design + 2 months for building
- Do not try to boil the ocean… you will run out of money, time, energy, and the patience of your friends and family

Do : Start with Wireframes

- Start with simple designs in Miro - ideally not on a napkin

Do : Flush Out Every Detail

- build the designs flush out EVERY detail before you begin building anything.
- This might take a week, but more likely it will take you 2-3 weeks full time… maybe even months

Don’t : Iterate on the Fly

- Developers are not designers Give them the complete designs - if there is an issue, then bring it back to design and have a SINGLE SOURCE OF TRUTH

Do : Build A Prototype

- Building a clickable prototype is worth it, you need to be able to click through every screen and know what every button does and where it goes next

Do : Start with Web

- If you are thinking about building an app, start in web - build in react and then you can build react native later…
- Even when you build react native, you really want to start with one platform: iOS or Android

Do : Start with iOS

- Even when you build react native, you really want to start with one platform: iOS or Android.
- I would recommend starting with iOS since it’s much simpler to build on and the users are worth more.

Do : Ask “What is the next most important thing we need to learn?”

“Simplicity is the ultimate sophistication”

Summary

- Simply put, your customer has to be at the center of your problem solving process.
- You must be deeply interested in the person for whom you are designing a product or service.
- Iteration at the core, no solution is optimal - everything requires testing and refinement