# Design playbook

### Introduction

No matter how cool software looks or how well it runs, the true measure of its success is pretty simple: does it meet the needs of its users? Instead of asking this in the middle of a project (or even worse, at the end) and crossing your fingers that the answer is yes, using human-centered design (HCD) makes this question the core of the entire design and development process.

Focus areas:

~~Design research~~

**Content design**

**Interaction design**

**Design systems**

Go to definition.

### Plays

#### 

#### 1. ~~Create group knowledge.~~ Do your research.

*Example:* [*Practical Design Discovery*](https://alistapart.com/article/practical-design-discovery)

***You can’t design a solution to a problem you don’t understand, which is why the discovery phase of design is so critical to the success of a project. Starting design with a [discovery plan](https://docs.google.com/document/d/1Wx-dtpz9erjwfcn0NoeFzlVXxZbDQ5L-E3O7gh878f0/edit?usp=sharing) and research are key to ensuring everything you build meets your users’ and customers’ needs. Software design teams have a host of information-gathering tools and activities at their disposal, including:***

* ***[Stakeholder and user interviews](http://theuxreview.co.uk/user-interviews-the-beginners-guide/): If you want to know what users want, ask them! Meeting diverse sets of project stakeholders and users where they are and getting their honest perspectives will allow you to create research artifacts such as*** [personas](http://www.ux-lady.com/introduction-to-user-personas/) ***[journey maps that visualize who your users are and how they might experience your solution from beginning to end](http://www.designkit.org/methods/63)***.
* **Competitor analyses:** Design doesn’t happen in a vacuum. It’s necessary to not only understand how your solution impacts users and stakeholders but also how others in the industry are addressing the same problems.
* [Heuristic analyses](https://uxdesign.cc/introduction-to-heuristic-evaluation-658705606518):
* **Content inventories:**

#### 

#### 2. Get consensus.

Example: ??

We can't say this one enough: good design is a collaborative process. Designers don’t magically know exactly what users want, what is technically possible, or what is operationally viable. Your team brings a lot of expertise to the table, so make sure that everyone is involved in defining the problem space, the users, and the end goal so the entire team is in sync about what needs to get built and why from day one. By coming to a shared understanding of the problem space at the beginning of the project, you’ll save yourself some major headaches down the road.

* ~~Affinity diagrams:~~

#### 

#### 3. ~~Visualize important systems.~~ Create visualizations.

*Example:* [Five Models for Making Sense of Complex Systems](https://medium.com/@cwodtke/five-models-for-making-sense-of-complex-systems-134be897b6b3)

Once you understand your users’ needs, it’s time to start thinking through the actual software design. This is where lo-fi visualizations like journey maps, user personas, empathy maps, storyboards, and affinity diagrams come into play. Sketching out your understanding of the problem space helps put you in the user’s shoes, so no matter what method(s) you choose to use, make sure your team takes the time to think through a typical user’s goals and movement through the app. After you’ve gotten a thorough understanding of how the features and site design should flow together, you can start creating your site maps and content models.

* [**Content model**](https://alistapart.com/article/content-modelling-a-master-skill): It’s hard to create design content if you don’t know how it will be organized. Content models are a great way to plot out all of the content elements required for your solution and the relationships between them.
* Site map
* Storyboarding
* Paper sketches: When in doubt, sketch it out! Drawing out your ideas with paper and a pencil (er, Sharpie) can be the first step to great design.
* [Lo-fi wireframes](https://methods.18f.gov/make/wireframing/)

#### 

#### 4. Test your rough ideas.

*Example:* [Prototyping for success](https://medium.com/@demianborba/prototyping-for-success-4bf3649fddff)

Before you take your model and run with it, it’s important to actually test it. The two key things to keep in mind during testing are speed and cost: don’t waste time and resources coding up an app to validate it when you could just as easily test it with sticky notes. Some useful testing tools include:

* **Hi-fi mockups: Once you’ve gained a thorough understanding of the bones of your solution, you can create and test hi-fi mockups to flesh it out. Hi-fi mockups are useful in content design and interaction design because they are a closer representation of the final user experience than wireframes and can guide the final stages of development.**
* [**Clickable prototypes**](https://blog.prototypr.io/14-prototyping-tools-how-each-can-be-used-1c804fab33a6)**: The tangibility of clickable prototypes makes them integral to interaction design. Clickable prototypes help users identify and describe their own pain points, accessibility needs, and workflows**
* **HTML and CSS templates:**

#### 5. Learn from feedback.

*Example:* [Improving Your App Design with User Testing](https://medium.com/soundwave-stories/validating-your-designs-with-user-testing-a5a4d8e9391d)

”Co-design” and “co-creativity” have become buzzwords in UCD, but trust us on this one: if you start thinking about design as a truly collaborative process, you’ll save yourself some major headaches down the road. Designers don’t magically know exactly what users want, what is technically possible, or what is operationally viable, so get your developers, stakeholders, and users involved in the design process. By keeping all the major players in the design loop, you’ll ensure that you’re actually solving emerging problems, rather than just assuming that you are. Using methods and tools like A/B tests, usability testing, Google Analytics, and CrazyEgg to test your prototypes will give you valuable information on how users interact with your software that you can then condense into product improvements.

* [A/B tests](https://www.smashingmagazine.com/2010/06/the-ultimate-guide-to-a-b-testing/)
* [Usability testing](http://uxmastery.com/beginners-guide-to-usability-testing/)
* Google Analytics

### 

### Definitions

**What makes good software design?**

Though the exact design methods used throughout a project depend on goals, team composition, and design objectives, good design work follows these three guidelines: 1) **involve the team in your process** by discussing design artifacts and user insights with everyone involved; 2) **review solutions and assumptions with users** by talking to them and learning what they really think; and 3) **build in time and processes for iteration** in order to incorporate user feedback into design and development. More importantly, good design is human-centered.

**What is human-centered design?**

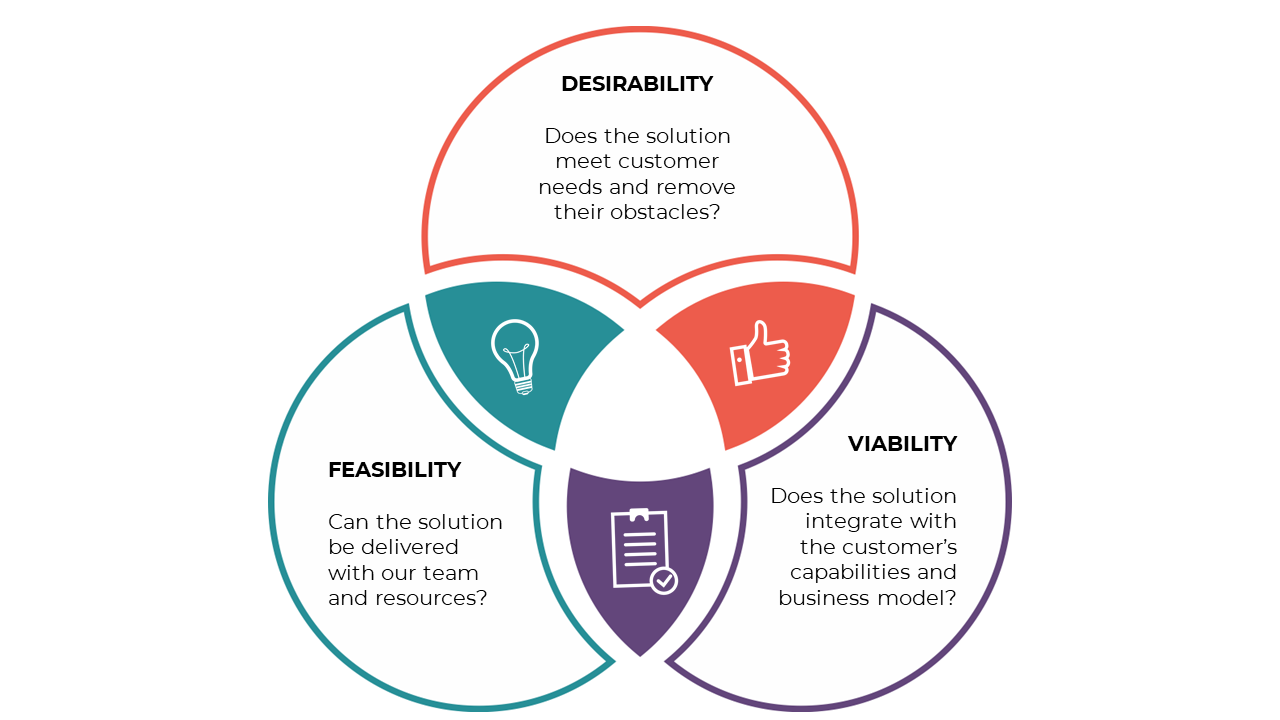
Human-centered design (HCD) is an approach to design and innovation that puts the needs of people affected by products and services at the heart of decision-making. HCD is at the heart of Fearless processes that build a deep empathy for users so that the final solution is intuitive, functional, and easy to use. It is often described as having three non-linear phases: 1) Inspiration; 2) Ideation; and 3) Implementation.

The human-centered process of Design Thinking is a tool used in many industries to balance **desirability**, **feasibility**, and **viability** throughout design. Design Thinking is often described as having six non-linear steps that, as shown in the below graphic, roughly map to the three phases of HCD: 1) Empathize; 2) Define; 3) Ideate; 4) Prototype; 5) Test; and 6) Implement.

# 

*HCD and Design Thinking phases*

**Why are HCD and Design Thinking important?**

****

*Desirable design satisfies user needs.*

*Viable design meets client goals.*

*Feasible design ships on time.*

As a community-rooted company dedicated to building software that solves problems and empowers people, Fearless keeps HCD at the heart of our design processes to build a deep empathy for users so that the final solution is intuitive, functional, and easy to use. We also bake Design Thinking methodologies into our work to balance the desirability, feasibility, and viability of our products. In other words, we want our software to fulfill users’ desires while remaining financially and technologically practical.

### Inspiration & Resources

* 18F Guides <https://guides.18f.gov/guides/>
* USDS Digital Services Playbook <https://playbook.cio.gov/>
* thoughtbot Playbook <https://thoughtbot.com/playbook>
* thoughtbot Product Design Sprint <https://thoughtbot.com/product-design-sprint/guide>
* Google Ventures Design Sprint <http://www.gv.com/sprint/>
* Google Developers Design Sprint <https://developers.google.com/design-sprint/>
* IDEO Design Kit <http://www.designkit.org/>
* Hyper Island Toolbox <http://toolbox.hyperisland.com/>
* Stanford D School Bootcamp Bootleg <http://dschool.stanford.edu/wp-content/uploads/2013/10/METHODCARDS-v3-slim.pdf>
* Circular Design Guide <http://circulardesignguide.com/>
* The Field Guide to Human-Centered Design <http://www.designkit.org//resources/1>