

Design Thinking

CS 346 Application
Development

Building exceptional products

We all want to produce *exceptional products that people will use*.
What does that mean?

Our product should be useful: it should satisfy a need or desire for our users. However, that is not enough. Success often means an *outstanding* user experience.

“The first requirement for an exemplary user experience is to meet the exact needs of the customer, without fuss or bother. Next comes simplicity and elegance that produce products that are a joy to own, a joy to use.”

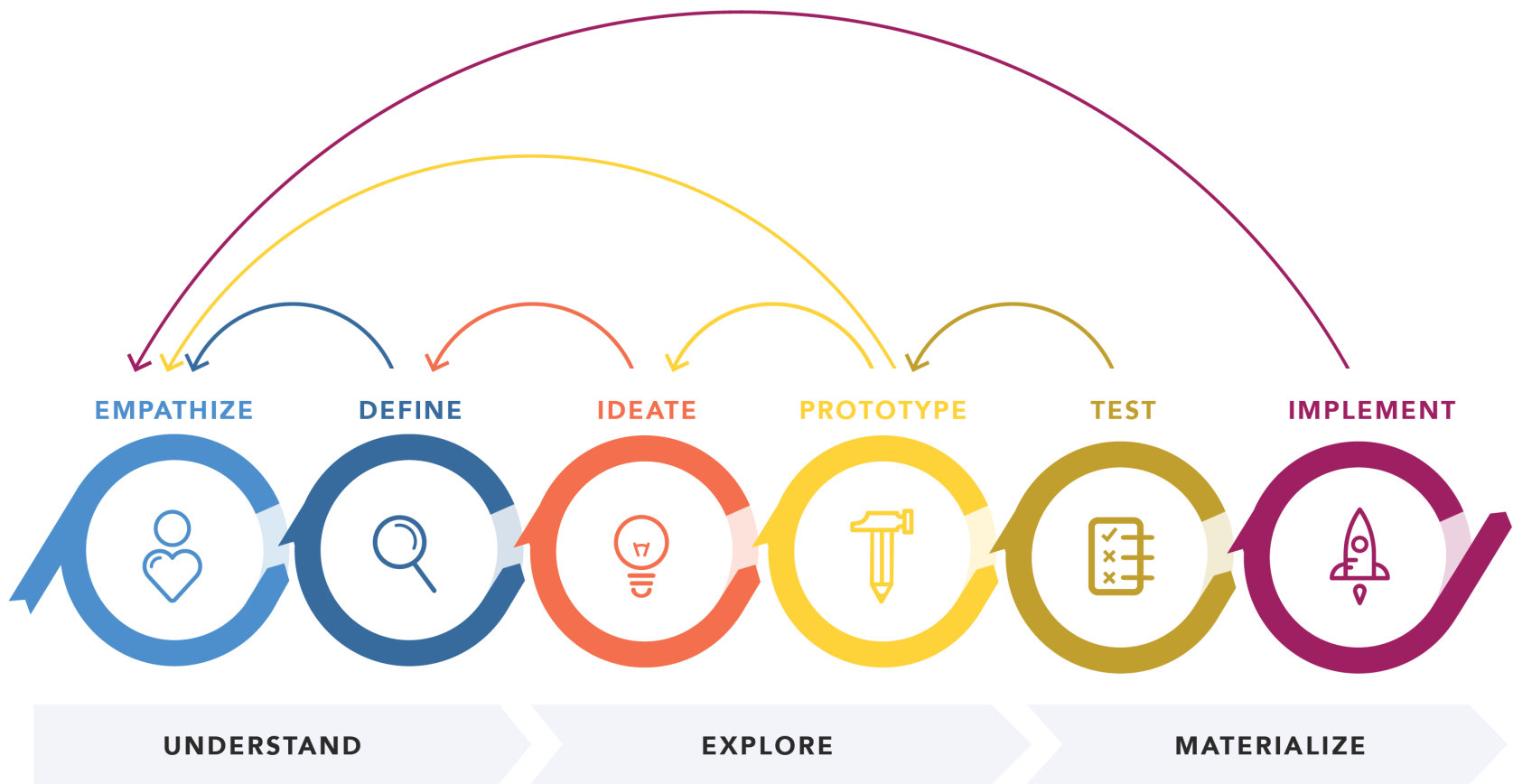
– Don Norman & Jacob Nielsen

Design thinking

Design thinking is a way to think about designing for people.

Don Norman characterizes design thinking this way:

1. It's fundamentally people-centred.
2. The goal is always to solve the right problems and address the root cause.
3. It considers everything as a system. You can't solve one little piece, because everything is interconnected. You really have to look at the nature of the system altogether.
4. It focuses on small and simple interventions. With complex systems, we probably won't get everything right, so build small and simple, check the results, change what we're doing and continually improve.



1/ UNDERSTAND phase

We take steps to understand the user, their context and potential problems that they might be experiencing.

1. Empathize: Develop knowledge about your users, and their context. Your goal is to gather enough information that you understand your users, how they work, and what motivates them.

2. Define. Look at the information that you've gathered, and observe where your user problems exist. Look for opportunities.

2/ EXPLORE phase

Brainstorm ways to solve their problems. Consider alternative solutions.

3.Ideate. Brainstorm ideas to solve the problem! You want lots of ideas, from any source (including competing products, things that “sound like a good idea”, or direct suggestions from users). Discuss them all and narrow down to the ones that work best.

4.Prototype. Work together to build mock-ups of your “best attempt” at a solution. Use the process to determine which components of your solution work and which ones need further refinement. Some ideas might not work together and need to be rejected - that’s ok too.

3/ MATERIALIZE phase

Iterate over potential solutions with your users and collect their feedback.

5.Test. Test with users by showing them the prototype and collect their feedback. Identify problems and areas to improve. Iterate on your designs!

6.Implement. Implement your final design solution.

PHASE 1: Understand

Understand your users, including what challenges and issues they might face.

Step 1. Empathize

The first step to building a solution for someone is to understand who they are, what challenges and constraints they face. You want to be able to clearly state:

- “Who are my users?”
- “What are they trying to accomplish?”
- “What motivates them?”
- “What discourages them?”

Your goal is to understand and empathize with your users and their perspectives.

Action: Conducting interviews

The best way to understand a set of users is to talk to them! You need to interview people to understand them, their perspective (how they work, what they like, what they dislike and so on).

Interview guidelines

- Interview 5–6 users (Nielsen 2000)
- Ask open-ended questions i.e., what they do, how they work.
- Continue asking questions to identify issues or challenges that they might have.
- Use follow-up questions to make sure that you understand their goals and motivations.

Interview Questions

Good interview questions

- “What is something that you do in this role that takes a lot of time”.
- “Describe a problem that you frequently encounter”.
- “How would you accomplish X”?
- “What do you like about this existing solution? What do you dislike?”

Poor interview questions:

- “I built this. Isn’t it cool?”
- “You want *what*? That’s a dumb idea.”

Action: Generating Personas

A **persona** is an archetypal description of a user of your product.

Personas are meant to be generalized descriptions that represent a category of people with similar characteristics. We use personas as stand-ins when discussing product goals and features.

- Think of personas as people with specific goals (i.e., someone who needs to accomplish something specific) or roles (i.e., someone with a series of responsibilities and expectations based on that).
- You will likely have multiple personas, where each persona represents a different type of user. It's not uncommon to have two or three personas, even for a small solution.

Example Personas

- **Bio:** Fictional name, job title, company, job description, family status, and more.
- **Demographic info:** Fictional age, gender, income, education, location.
- **Image:** A vivid image can help your team picture your users clearly and help establish a consistent understanding of the target.
- **Goals & motivations:** Make it clear what your audience wants to get or to do with your product.

Max Banker

BIO

Max is a business student who has had multiple internships at various institutions. He's a capable computer user for browsing the internet.

GOALS

Max has a lot of trouble keeping track of his work. Anything that would help him would be useful.

FRUSTRATIONS



QUOTES

"I'm too busy to learn now."
"I hate my computer (Windows)."

DOMAIN AWARENESS:
Not much. He scribbles on his paper notebook at work.

Jessica Coder

BIO

Jessica is a professional software developer. She graduated from UW with a BCS and a minor in CO. She is currently the tech lead for her current project at Shopify.

GOALS

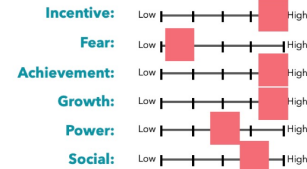
She is a packrat for saving technical information - code snippets, articles online.

FRUSTRATIONS

She finds it difficult to keep track of everything that she needs to work with in a day.

MOTIVATIONS

Anything to make her more efficient.



QUOTES

"If I have to use a mouse, that just annoys me. I'm faster with a keyboard. VIM all the way."
"Information should be easy to access, and always accessible. Having everything searchable without breaking the flow would be amazing."

DOMAIN AWARENESS:
Currently uses markdown notes and manually "grets" based on keyword.

TECH KNOWLEDGE:
Expert in software, technology. Expects an advanced, polished solution.



Age: 26
Occupation: Software Developer
Status: Single
Location: Toronto
Archetype: Developer
Personality: "Gadget person"; loves new tech; Android over iOS because she can tweak it.

Step 2. Define

This step involves identifying problems that your users may have disclosed in the interview.

- In reviewing your interview notes, you should be able to identify the needs that people are trying to meet.
- One way to do that is to underline *verbs* in the interview notes. These represent activities they are performing.
- Watch for actions that are unfulfilled or frustrating to them.

These actions or stories represent things the user needs to accomplish. You should focus on these!

Action: Capturing User Stories

A **user story** is an informal, general explanation of a software feature written from the perspective of the end user.

- From your interviews, you should extract these user stories. They are the foundation for what you will build.
- You do NOT need to build everything you identify. Look for related actions or stories.
- User stories should be few sentences in simple language that outline the desired outcome. They don't go into detail.
- We will often pair user stories with personas to talk about "hypotheticals" when discussing potential solutions.

Action: User Story Examples

- “As a manager, I need to track my staff’s hours worked. To do that, I need to query their start and stop times for each shift and do the math; it’s time-consuming”.
- “I spend a lot of time in meetings, and taking detailed minutes. I mostly use MS Word, but it’s really difficult to format them during the meeting.”
- “As a software developer, I spend a lot of time collaborating with my team. This has become much more difficult since we work remotely a lot of the time. I wish I could share my sketches more easily online.”

Action: Define a Problem Statement

You should be able to capture an over-arching statement of the problem that you are trying to address:

- Look for the theme of your user stories. What is the common problem that you are addressing?

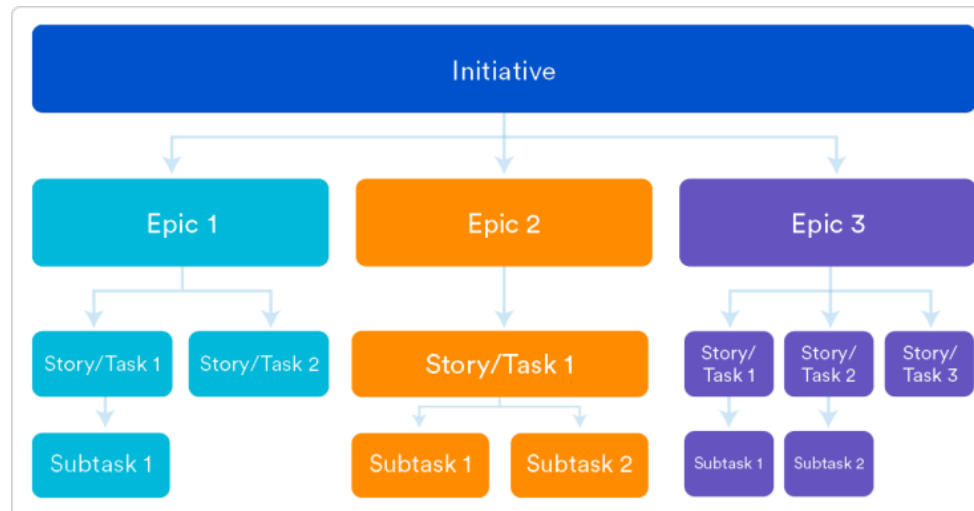
e.g., Lawyers need to track their time carefully. Existing solutions fail to capture all of the required data, or pose billing challenges. We propose a software solution to help them easily capture time worked by client, project and category on their phone or computer, and generate summary and billing reports.

Terminology: Initiatives, Epics, User Stories

Stories, also called “user stories,” are short requirements or requests written from the perspective of an end user.

Epics are large bodies of work that can be broken down into a stories.

Initiatives are collections of epics that drive toward a common goal.



PHASE 2: Explore

Brainstorm solutions to your problem statement!

Step 3: Ideate

You have a problem statement, and a bunch of user stories.

This step is all about brainstorming solutions!

- Bring your team together and brainstorm ideas you have that could contribute towards a solution.
- No idea is too far-fetched! Share ideas with one another, mixing and remixing, building on others' ideas.
- Reduce the set of solutions to a single “best solution” i.e., a set of user stories and features to address them.
- You will want to document every idea that you have, not just the “winning ideas”.

Action: Affinity Diagram

One way to collect ideas is to build a large [affinity diagram](#).

1. Write down each idea on a sticky note.
2. Put them on a whiteboard or large wall.
3. Organize them into clusters of related ideas. This can help you create hierarchies, or figure out dependencies.
 - You can use this to cluster use cases.
 - You can also do it to cluster features/solutions.

Goal: identify features that address the problem.



Step 4: Prototype

Next, you will build a prototype of your solution. A **prototype** is essentially a mock-up of your solution, that is built to demonstrate functionality and elicit feedback from your users.

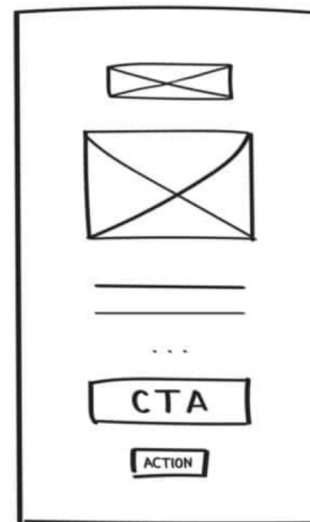
- The goal of this phase is to understand what components of your ideas work, and which do not! In an ideal world, everything would work perfectly, but you probably will need to iterate a few times.
- You want to show this prototype to your user and walk through all of the features to get initial “buy in”.
- You will not get everything right! The goal is *early feedback*.

Prototypes

Focus on low-fidelity prototypes.

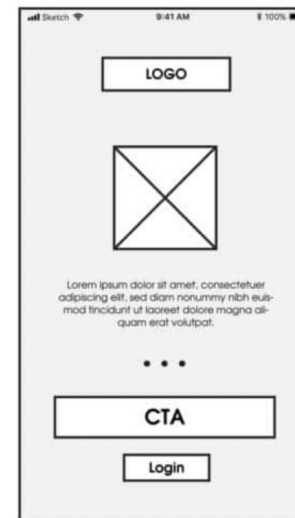
Low-fidelity prototypes are deliberately simple, low-tech, and represent a minimal investment.

- You can sketch something on paper.
- Many online tools help you build wireframe diagrams that you can demo e.g., Figma.
- You can even make them semi-interactive to test progression through the interface.

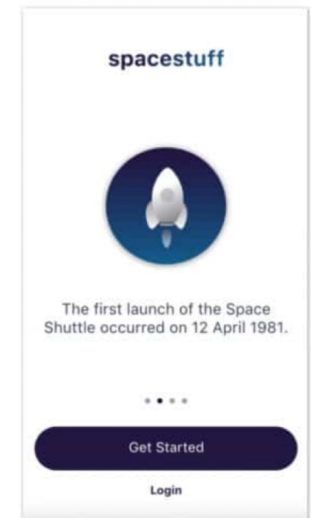


Low

X



Medium



High

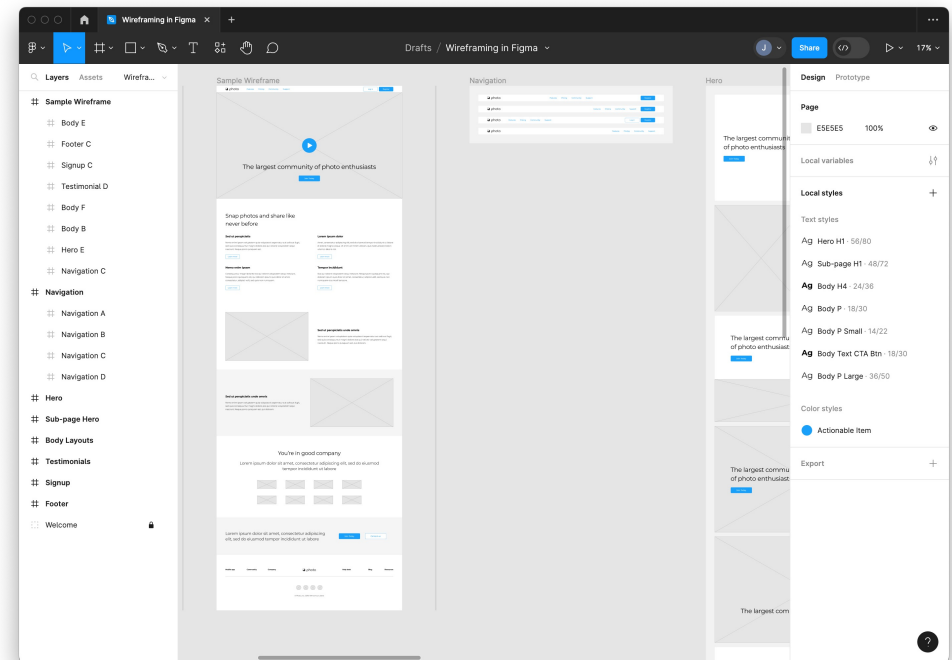
Recommended Tools

Figma is a very popular prototyping tool (desktop, mobile, web).

- Mock screens & interactions.
- Can build specific UI designs (e.g., iPhone, iPad, desktop).
- Iteration is much easier compared to paper prototyping.

Other options

- Omnigraffle (mac)
- Balsamiq (win, mac, web)
- Hand-drawn diagrams



<https://www.figma.com/>

PHASE 3: Materialize

Iterate on your solutions with your users.

Step 5: Test

This step involves taking your prototype back to your users and allowing them to walk through it. Your goal is to collect their feedback! You want to address questions like:

- Does this solve your problem?
- Is it easy to understand? To use?
- Does (this functionality) make sense to you?
- What would you change?

Write down feedback and take it back to your team. Iterate!

Step 6: Implement

At some point, you will have enough feedback to be able to proceed with the actual implementation. You should still demo occasionally to users, but you have enough information to proceed.

Tip: Keep your prototypes in-sync with your code as you are developing it. If you need to change anything, it's easiest to be able to put a prototype in front of you user (again) to get more feedback.