

UNIT IV WIREFRAMING, PROTOTYPING AND TESTING

Sketching Principles - Sketching Red Routes - Responsive Design – Wireframing - Creating Wireflows - Building a Prototype - Building High-Fidelity Mockups - Designing Efficiently with Tools - Interaction Patterns - Conducting Usability Tests - Other Evaluative User Research Methods - Synthesizing Test Findings - Prototype Iteration

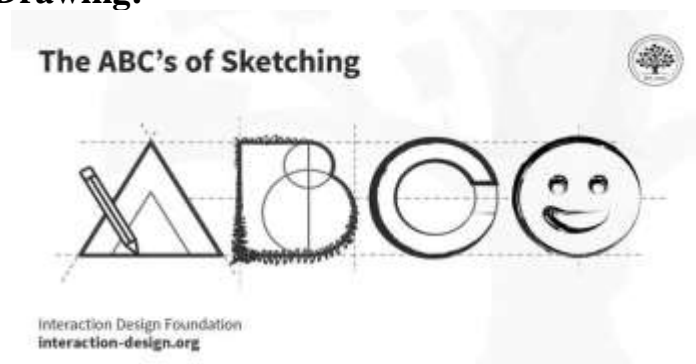
SKETCHING PRINCIPLES

Sketching:

- Sketching is a distinctive form of drawing which designers use to propose, explore, refine and communicate ideas.
- Sketches are easy, fast, and cheap to create, iterate, and if needed, even discard without much effort. Unlike written or verbal communication, sketches sidestep rules of grammar and help clearly communicate ideas, all but eliminating misunderstandings.



The Five Elements of Drawing:



Features of Sketch:

Sketches are:

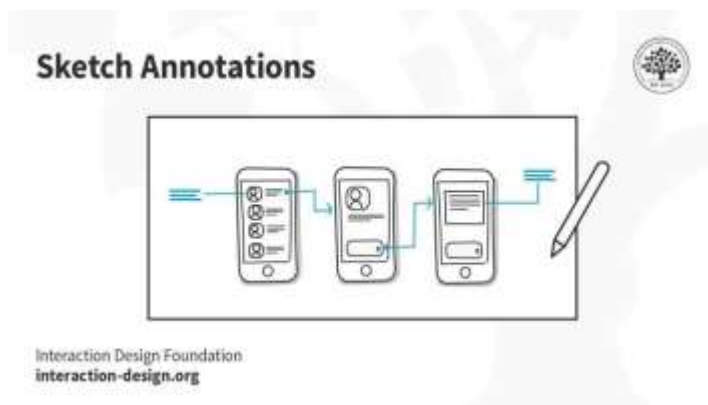
Quick: Don't invest a long period producing them.

Timely: Produce them when and as the need arises.

Disposable: Rely on their usefulness to explore a concept and not on their production costs.
Plentiful: Produce sketches as a collection that explores different aspects of interaction over time.

Minimalist: Use sketches to clarify one concept at a time.

Enhancing Sketches — Annotations, Arrows and Notes



Annotations, arrows and notes increase the communicative power of your sketches.

Annotations – These are names, labels and explanations located next to different parts of a sketch to expand and clarify the meaning of any element depicted. Tie annotations to different elements in your sketch using arrows, braces, numbering and spatial proximity. Write your annotations using a different color that contrasts with the sketch proper.

Arrows – Apart from pointing to specific elements in a sketch, use arrows to illustrate interaction flow, a sequence of events, movement and direction.

Notes – Any text, long or short, that provides additional insight into your sketch is a note. Use notes to do the following:

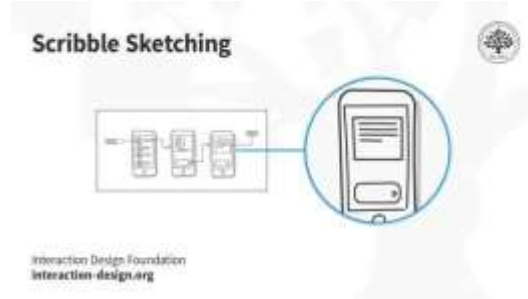
- Provide detailed explanation of the action or sequence illustrated.
- Describe an idea derived from an illustration.
- List unresolved issues.
- Explore design elements not depicted in the sketch.
- Clarify the purpose of each element you present, especially non-static ones.
- Keep a record of your thought process when you first draw a sketch.

Types of Sketching:

1. Scribble Sketching

The idea behind this technique is to capture, as fast as possible and with the broadest of strokes, the essence of the object, design or action you are trying to preserve. Leave out non-

important details, decorations, text and other non-essential elements.



2. Sampling with Cameras

The goal of this technique is to use still photos and video to capture some features of the world. Capture objects, designs and actions that delight, inspire and irritate you.

Pros and Cons of Sketches

Pros of sketches

- They are extremely cheap and fast to create. As such, you can sketch out a large number of ideas in a short amount of time.
- You can do it anywhere: with pen and paper or digitally on your smartphone, tablet or desktop computer.
- They are disposable, so you won't get attached to sketches that turn out to be bad ideas.

Cons of sketches

- Sketches lack detail and are ambiguous by design. As such, you cannot use sketches to convey complex interactions of an app, for example.
- Sketches are almost never of high enough fidelity to be useful with people outside of the team, since they rarely have the context to understand what the sketch is meant to convey.
- Sketches are not very helpful in convergent processes where you want to select a few best ideas—other forms of prototypes, such as paper prototypes or wireframes, are more helpful.

When to Use Sketches

Use sketches in early, divergent stages of your design process.

Sketch out your rough ideas so you can discuss them with team-mates.

Sketching Principles:

Best Practices and Tips for Sketching

- Always sketch out your ideas, rather than store them in your head! Design thinking emphasizes a bias towards action. Whenever you have an idea, sketch it out, no matter how silly it seems—you will be able to evaluate it much better when it's on paper rather than in your head.
- Use the right amount of detail: remember that a sketch should be rough and quick. Don't spend extra time adding details which are not required for your quick sketch.
- Draw diagrams to map out complex ideas or use cases, where many factors and players affect one another. Journey maps, behavior maps, system flow diagrams and a range of other mapping methods are at your service to help you scope out complex situations.
- Invite other team-mates to join in your sketching sessions, when appropriate. Because sketches are so easy to create, they are great opportunities for you to involve other stakeholders in the design process.

The best way to gain confidence in sketching is to practice.

Create a Record of Failed Design Efforts

Keep a record of failed design efforts. Gather objects and situations that annoy you. Write down the reasons behind this negative effect.

Create a Record of Successful Design

keep a record of successful design. This will help you draw inspiration from multiple domains to ground your design work on features other than those of the digital realm.

How to Perform a Sketch Test

The procedure is simple:

1. Print a copy of the deliverable
2. Recruit a sketch-test participant from your target audience. For example, if you're preparing a wireframe for developers, recruit a front-end developer.
3. Give the sketch-test participant the printed copy of the deliverable. Place a blank pad of paper and a pen or pencil right next to the person.
4. Invite the participant to write directly on the deliverable, but also make scratch paper available. Explain that the printed version of the deliverable is just a scratch copy, and that you welcome any comments.
5. Ask the participant to explain the concepts in the deliverable.
 - Explain that the current version of the deliverable is a work in progress, and that you are still developing the format of the document. While content suggestions can be welcome, at this point do not ask for them explicitly — keep the focus on making the

deliverable clear and direct.

- Note that you want to make this document easier to understand.
- Give the participant time to read the deliverable. Ask your participant to use the think-aloud protocol during the initial reading of the document.
- Ask the participant to pretend that she is presenting this document, and have her explain the deliverable to you.

6. Watch and listen as your participant explains the concepts in your deliverable.

7. Ask your participant open-ended follow-up questions

8. At the end of the session, explain the intended —correct meaning of the document to your participant, to avoid having him leave with an erroneous understanding of your work.

Feedback to Look for During a Sketch Test

The sketch test can help surface two types of problems with deliverables: (1) content that is not easily perceptible in the document (e.g. using blue text to annotate a screenshot made it difficult to notice against a similar background), and (2) content that is not comprehensible

During a sketch test, notice if your participant exhibits any of the following behaviors,

- Circling or underlining elements
- Drawing something to represent content that your deliverable presents in text, or employing a different visual metaphor than you currently use for data visualization or infographics
- Making notes or edits on your sketch, or suggesting terminology or wording for concepts other than those shown in the deliverable
- Reading the same passages or studying the same images several times
- Struggling to explain something to you
- Explaining something incorrectly to you

Improving the Deliverable Based on the Sketch Test

If your participant struggles to explain your deliverable easily, you will likely need to revise it before sharing it with your colleagues.

Feedback about the Content of the Deliverable

During a sketch test, you may receive feedback not only about the format of your deliverable, but also about its content.

SKETCHING RED ROUTES

Dr. David Travis of User focus identified that the analogy of the red route. By identifying the core paths users will take on a website should be identified as the red routes.

- Red Routes are majorly derived from —Important roads in London.
- Transport for London do everything in their power to make sure passenger journeys on these routes are completed as smoothly and quickly as possible
- London's Red Routes as a metaphor for frictionless user journeys.

How does Red route help in Designs ?

—When applied to design, these red routes are the critical and frequent paths that users take to complete their tasks

Red Route principle

Define red routes for your product and you'll be able to identify, prioritise and eliminate any usability obstacles on key user journeys.

Identifying Red Routes

- Critical
- End-to-end tasks with multiple steps or actions
- Frequently utilised
- Built for scale Key value drivers
- Objectively successful
- Tied to critical product metrics

Red route usability

—Focusing on your product's red routes provides the key constraint you need to ship a high value product from version 1

Red routes improve speed and effectiveness:

We can ruthlessly eradicate any usability obstacles on the key user journeys.

Red routes describe frequent and critical activities :

We need to map out the red routes for our site it's important to consider both the frequency and critical nature of the activity

Red routes should reflect key business objectives :

These red routes are clearly important from the perspective of the organisation: you need to make sure these routes are trouble-free to make money

Red routes should reflect key customer objectives :

Visitors to your site will have their own goals that your site needs to support. there will be others that may not seem that important to your organisation but that are critical if customers are going to do business with you

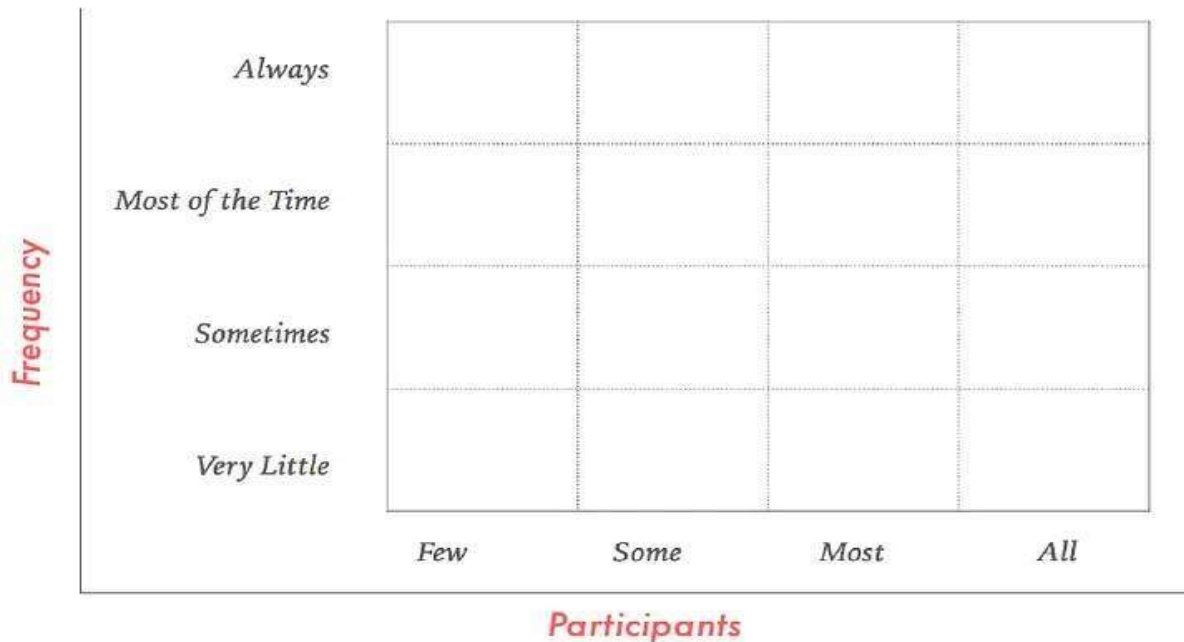
Red route Analysis

—Red routes are the key tasks that users want to carry out with the product —

We can decide if one of the product backlog items lies on a red route by asking two simple questions :

- How many users need this function?
- How often do users need this function?

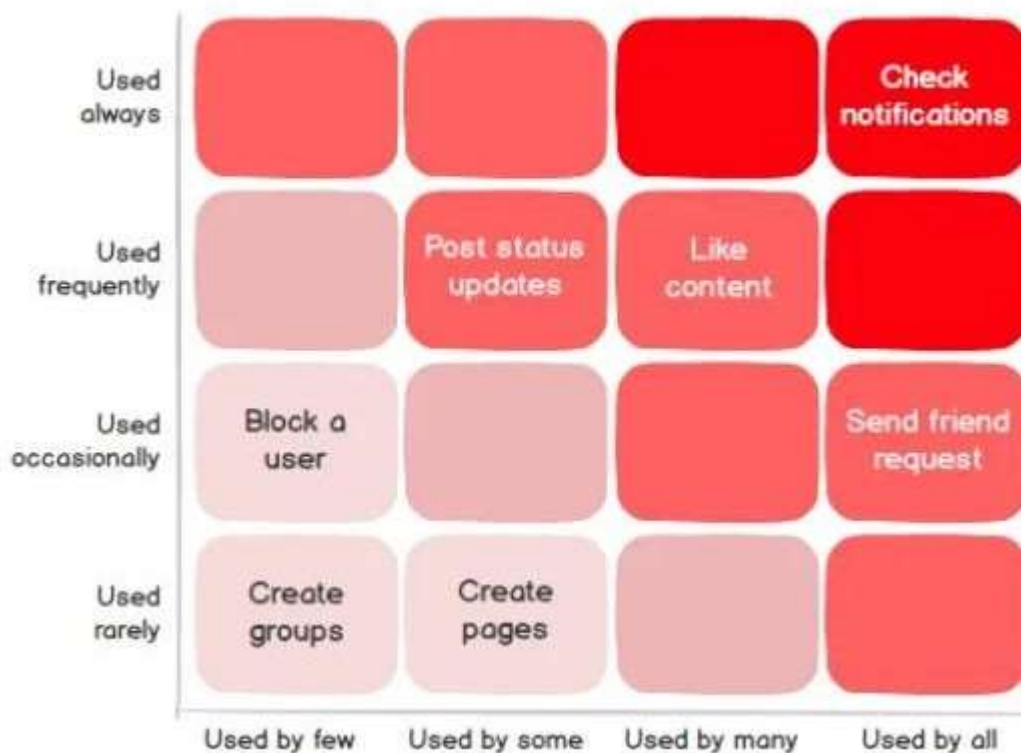
** Base your red routes on real customer data "*



By identifying what the top tasks of your users are it allows you to:

- Anticipate user needs
- Guide usability testing
- Target essential website pages
- Design website with user needs in mind
- Identify your website's mission

Example : Facebook



Benefits of Identifying Red Routes :

—Defining crisp red routes for your users is like allowing them to travel from Point A to Point B without any obstacles

- Identifying red routes helps your team prioritise user needs and facilitate alignment amongst your stakeholders.
- It also helps avoid scope creep and the introduction of extraneous/ancillary features.
- Most importantly, it helps your team build and optimise product features that deliver the most value to your customers and drive your key metrics.

Route Type	Customer objective	Business objective
Uber - Red route	Requesting a ride	Accepting a ride
Uber - Normal route	Adding a payment method	Changing profile settings
Red bus - Red route	Booking a bus	Managing reservations
Red bus - Normal route	Leaving a review	Updating trip details

RESPONSIVE DESIGN

Responsive web design (RWD) is a web development approach that creates dynamic changes to the appearance of a website, depending on the screen size and orientation of the device being used to view it. RWD is one approach to the problem of designing for the multitude of devices available to customers, ranging from tiny phones to huge desktop monitors.

RWD uses so-called breakpoints to determine how the layout of a site will appear: one design is used above a breakpoint and another design is applied below that breakpoint. The breakpoints are commonly based on the width of the browser.

Definition:

Responsive design is an approach to web design in which the interface adapts to the device's layout, facilitating usability, navigation and information seeking.

Benefits of Responsive design:

- Reach a larger audience: nowadays, more users access the web through mobile devices.
- Save time and development efforts since designers and developers only focus on one design version.
- Improve SEO, as search engines reward mobile-friendly websites with better search positions.
- Ensure brand and design consistency across devices, as there is no chance to modify guidelines to fit different design boxes.

Why Responsive Design is so Popular?

More users were starting to access web material on handheld devices than on desktops. There were two main design approaches to deal with designing across devices:

- Designers could craft several versions of a design optimized for different devices and make each have fixed dimensions (adaptive design approach).
- They could work on a single, flexible design that would stretch or shrink to fit the screen (responsive design approach).

Understanding the Language of Responsive Design

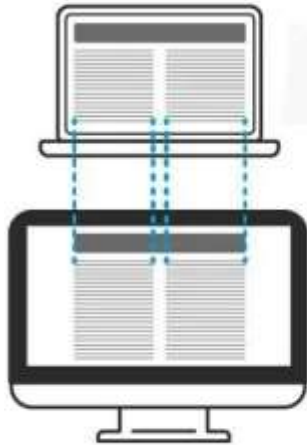
Responsive design has three core principles:

Fluid Grid System

Grid systems are aids designers use to build, design, arrange information and make consistent user experiences. In interaction design, multi-column, hierarchical and modular are the most widely-used types of grids.

The principle of a grid is simple: every element occupies the same percentage of space, however large or small the screen becomes, which means that the components can be scaled up and down as the user switches devices.

Fixed Version

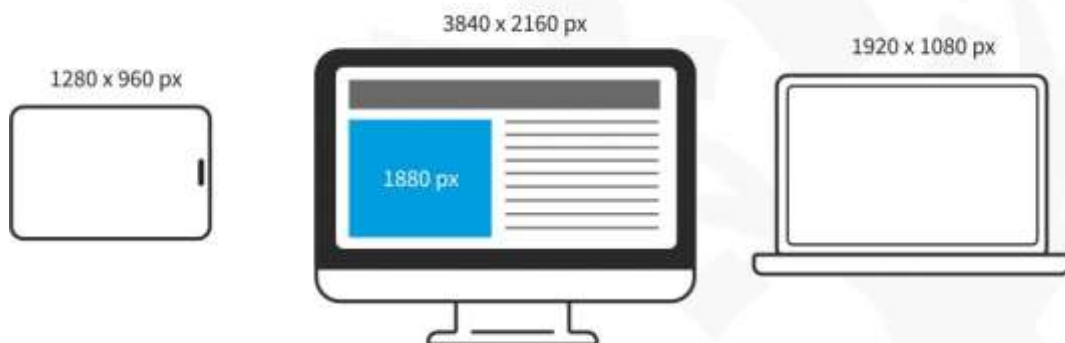


Fluid Version



In this image, you can see that the fixed version of the content has the same width regardless of the device whereas in the fluid version the content fills the available space depending on the device size.

Fluid Images



- In responsive design, fluid images are images that scale to fit their container, meaning that when the browser reaches a breakpoint, the image will scale up or down to the current window size.
- For non-photographic images, such as icons, you can use SVG files—these file formats are lightweight, and you can scale them to any resolution without losing quality.

Media Queries and Breakpoints

- Media queries are filters that detect the browsing device's dimensions and make your design appear appropriate regardless of the screen size. To aid media queries, you have

breakpoints: these are the values where the content of your website will be rearranged to provide the user with the best possible experience.

- Media queries and breakpoints go hand in hand, and both can be defined in your CSS style sheets. For designers, a breakpoint is a boundary where the design will change to accommodate the features to the new size.

Best Practices & Considerations for Responsive Design

With responsive design, you design for flexibility in every aspect—images, text and layouts. So, you should:

1. Assume a "mobile-first" mentality: Whether you start designing from the smallest screen or the desktop version, assuming a mobile-first mentality helps you practice the mobile-first approach, a design principle with simplicity at its core. A mobile-first mentality means prioritizing content, leaving complex graphs and images for the desktop version
2. Use Scalable Vector Graphics (SVGs). These are an XML-based file format for 2D graphics, which supports interactivity and animations.
3. Include three or more breakpoints (i.e., design for 3+ devices).
4. Prioritize and hide content to suit users' contexts. Check your visual hierarchy and use progressive disclosure and navigation drawers to give users needed items first. Keep nonessential items (nice-to-haves) secondary.
5. Aim for minimalism.
6. Apply design patterns to maximize ease of use for users in their contexts and quicken their familiarity: e.g., the column drop pattern fits content to many screen types.
7. Aim for accessibility with font sizes/styles. Use contrast and background effectively.

Demerits of Responsive design:

Responsive design can still run into difficulties if you use it without caution. For example, it can restrict your control over the design's screen sizes

WIREFRAMING

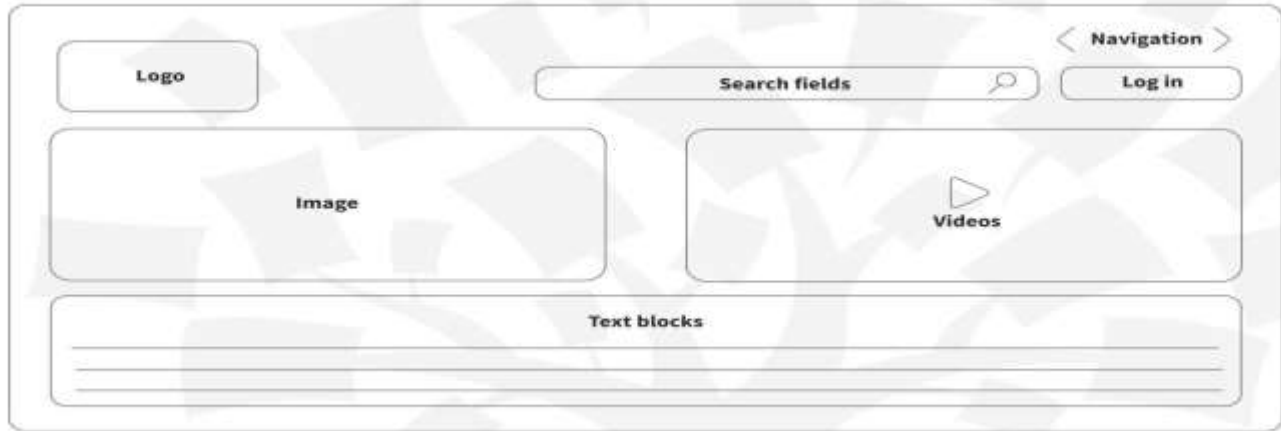
Definition:

Wireframing is a visual representation of the workflow of a website or mobile application. This preliminary design step illustrates the page's structure, layout, and functionality. It helps in planning the user experience without distractions from color, graphics, or content.

The Elements of a Wireframe



Elements Of A Wireframe



Interaction Design Foundation
interaction-design.org

1. The logo signifies brand identity. It usually sits at the top corner of a page and anchors the brand's presence.
2. Navigation helps users explore different parts of the site or app. Commonly seen as a menu bar or sidebar, it directs users to primary sections or features.
3. Search fields allow users to find specific content or features. It's especially useful for content-heavy sites.
4. Text blocks on a wireframe represent the content placement. It can be headlines, paragraphs, or bullet points.
5. Buttons prompt users to take actions like "Submit," "Read More," or "Buy Now."
6. Image placeholders show where you will place your visuals. They help in understanding the content-to-visual balance on a page.
7. Videos mark the areas on the wireframe where multimedia content, such as videos or animations, will appear.

Different Types of Wireframes

1. Low-fidelity Wireframes

Low-fidelity wireframes are the first you create. They provide the basic visuals of the design, but lack scale, grid, or pixel accuracy. The main aim is to strip away distractions.

These wireframes ease discussions, help in setting navigation, and outline user paths.

2. Mid-fidelity Wireframes

Mid-fidelity wireframes are the most prevalent. They offer a clearer layout view, though they still omit visuals like images or specific typography.

3. High-fidelity Wireframes

High-fidelity wireframes are detailed. They provide pixel-specific layout views. Unlike the low-fidelity version, which might use placeholder text and symbols, these wireframes present actual images and relevant content.

When to Use Wireframes?

Wireframes play a role throughout a project. They're essential for clarity, feedback, planning, and testing. But they serve a specific purpose. Here are the scenarios where creating wireframes can help.

- **Explore the initial idea:** At the start of a project, wireframes help visualize rough ideas. They bring abstract concepts to life.
- **Collect meaningful feedback:** Before you get into the details of a design, wireframes help you gather initial feedback. Stakeholders, members of the design team, and potential users can provide valuable insights before you move on to the next step.
- **Plan functionality:** Wireframes map out where you will place functional elements like buttons or interactive features. It helps you understand how to create an interaction design for usability.
- **Structure content:** Wireframes help plan where content will appear. For instance, positioning text, images, or multimedia is easier.
- **Customer journey mapping:** Wireframes are tools to plot user journeys. They enable designers to envision how users will navigate a site or app.
- **Usability testing:** Before final designs, wireframes can undergo usability tests. It helps identify and rectify usability issues early.

Choosing the Right Wireframing Tools

Benefits of Digital Wireframing

- **Precision:** Digital tools offer accurate measurements. This ensures elements align and match the intended design.
- **Efficiency:** Speed up the wireframing process. Features like copy, paste, and templates save time.
- **Collaboration:** Many digital tools have built-in collaboration features. Teams can review, comment, and iterate together in real time.
- **Version control:** Digital wireframing can help you track changes. Most tools offer version history and allow designers to revert or compare versions.
- **Easy sharing:** You can share digital wireframes with a link. It eases the process of gathering feedback.
- **Interactivity:** Some tools enable clickable wireframes. This simulates user interactions by offering a dynamic preview.
- **Scalability:** Adjusting wireframe sizes for various devices is straightforward. It ensures designs are responsive.
- **Integration:** Many tools integrate with other design software. This streamlines the transition from wireframing to high-fidelity designs.
- **Asset management:** It's easy to organize and store assets. Icons, components, and UI elements remain at your fingertips.
- **Professional presentation:** Digital wireframes look polished. Presenting them to stakeholders or clients gives a professional impression.

Wireframing tools and software

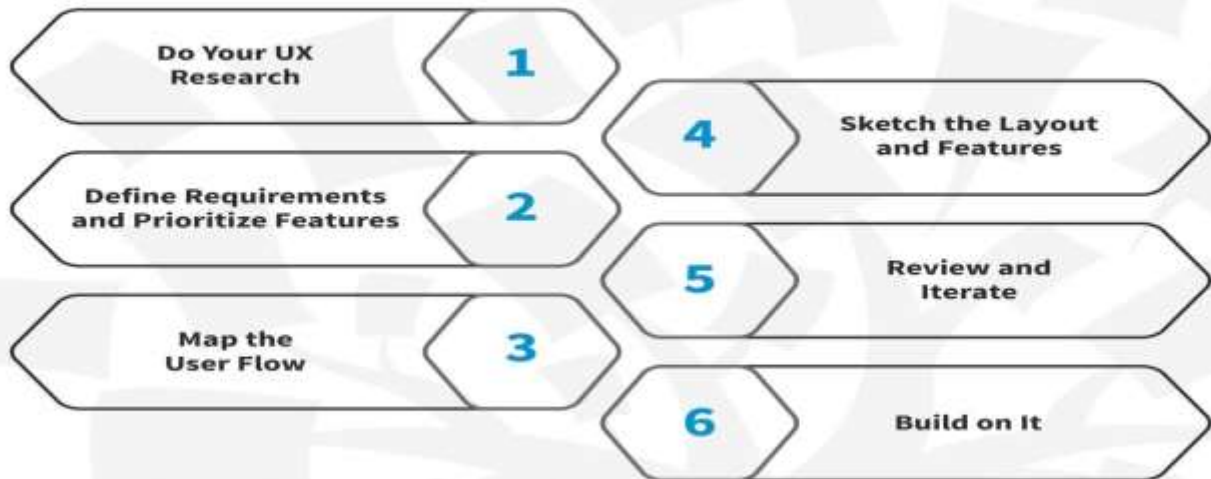
Wireframing tools help you map out the user experience, layout, and overall flow. It streamlines the design process and ensures everyone is on the same page.

1. Figma
2. Mockflow
3. Lucidchart

4. Miro

How to Create Wireframes: A Step-by-Step Guide

How To Create Wireframes



Interaction Design Foundation
interaction-design.org

Step 1: Do Your UX Research

Before sketching, you need to have a good understanding of your audience and business. Start your UX research from the perspective of your target audience.

What motivates them?

What holds them back?

Which products resonate with them?

Utilizing tools like user personas or the jobs-to-be-done (JTBD) framework can be invaluable here.

Step 2: Define Requirements and Prioritize Features

After you complete your UX research, the next step is narrowing down what to build. At this stage, involve stakeholders or the product owner. They will help translate broad user needs into specific features.

For example, if your e-commerce site aims to improve user engagement, it may require a 'Recommended Products' section. Stakeholders can provide insights on what features align with both user needs and business goals.

Step 3: Map the User Flow

User flow is the blueprint of a user's journey on your platform. Taking an e-commerce site as an example, a user's path might start from the homepage, leading to a product search, then product selection, payment, and finally, order confirmation.

Recognizing these steps will highlight the necessary features and inform your design decisions.

Step 4: Sketch the Layout and Features

With the user research and project requirements in hand, sketch your wireframe. Select an appropriate canvas size and start positioning elements.

- Information architecture: Focus on how you organize the content and prioritize information based on its significance.
- Interactive elements: Highlight buttons, links, and other clickable items.
- Static elements: Remember foundational parts like headers, footers, or menus.
- Fidelity choice: Decide on the detail level. Early stages might use low-fidelity sketches, while detailed projects lean towards high-fidelity designs. In between, you find medium-fidelity designs.

Step 5: Review and Iterate

- Share your wireframes. Gather input from stakeholders, including business professionals and developers. Consider guerrilla usability tests, which involve testing the wireframe with users to gather insights.
- When you collect feedback, frame your questions to include broad insights and specific details.

Step 6: Build on It

Upon reaching a mutual agreement on the wireframe design, progress to the next stages. Create detailed mockups and interactive prototypes and transform them into minimum viable products

What is Wireframing in UX?

Wireframing in UX refers to creating a skeletal layout for a digital product. It's the blueprint or foundation upon which you create the design

Different platforms have specific wireframe sizes. These are:

- Smartphone screen: 1080 px x 1920 px
- 8" Tablet screen: 800 px x 1280 px
- 10" Tablet screen: 1200 px x 1920 px
- Desktop screen: 768 px x 1366 px

The Necessity of Wireframing

- Clarity: Wireframing interprets abstract ideas to convert them into a visual format.
- Feedback: Early-stage wireframes gather valuable feedback from stakeholders and users.
- Cost-efficient: It's easier and cheaper to make changes to a wireframe than to a fully designed product.
- Focus on Functionality: Without distractions like color or graphics, the core user journey and functionality remain at the forefront.

Approaches to Wireframing

- Hand-drawn: Quick sketches on paper or a whiteboard. Ideal for brainstorming.
- Low fidelity: Basic visual design that focuses on structure and layout.
- High fidelity: Detailed, closer to the final design. They showcase interactions and more precise placements.

Wireframing Principles

- Simplicity: Keep it straightforward. Focus on structure and functionality.
- User-centered: Prioritize user needs and the intended user journeys.
- Clarity: Every element should have a clear purpose.
- Feedback loop: Constantly gather feedback and iterate.
- Consistency: Maintain a uniform structure and design language.

Wireframing for User Testing

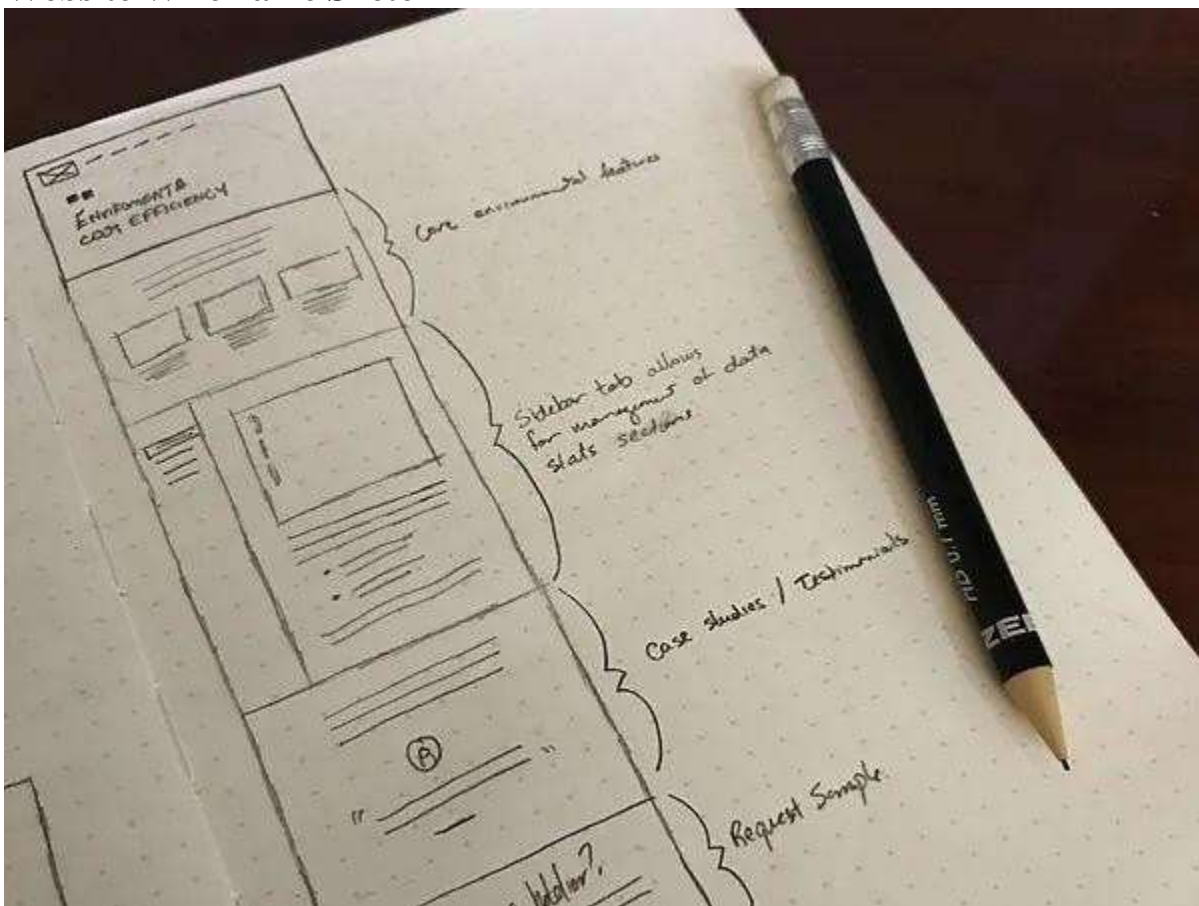
Testing with wireframes is invaluable. Before getting deep into design or development, wireframes provide a tangible product for users to interact with. This early-stage testing identifies potential usability issues, layout problems, or unclear navigation paths.

Wireframe Page Layout: Structuring Content

- Header: The topmost section, usually containing the logo, navigation links, or contact details.
- Main content area: The primary section where the core content resides. It's the focal point of the user's attention.
- Sidebars: Secondary content areas, often containing links, ads, or additional information.
- Footer: The bottom section displays copyright information, additional links, or contact details.

UX and UI Wireframe Examples

1. Website Wireframe Sketch



A basic homepage wireframe sketch. It outlines sections and highlights essential details.

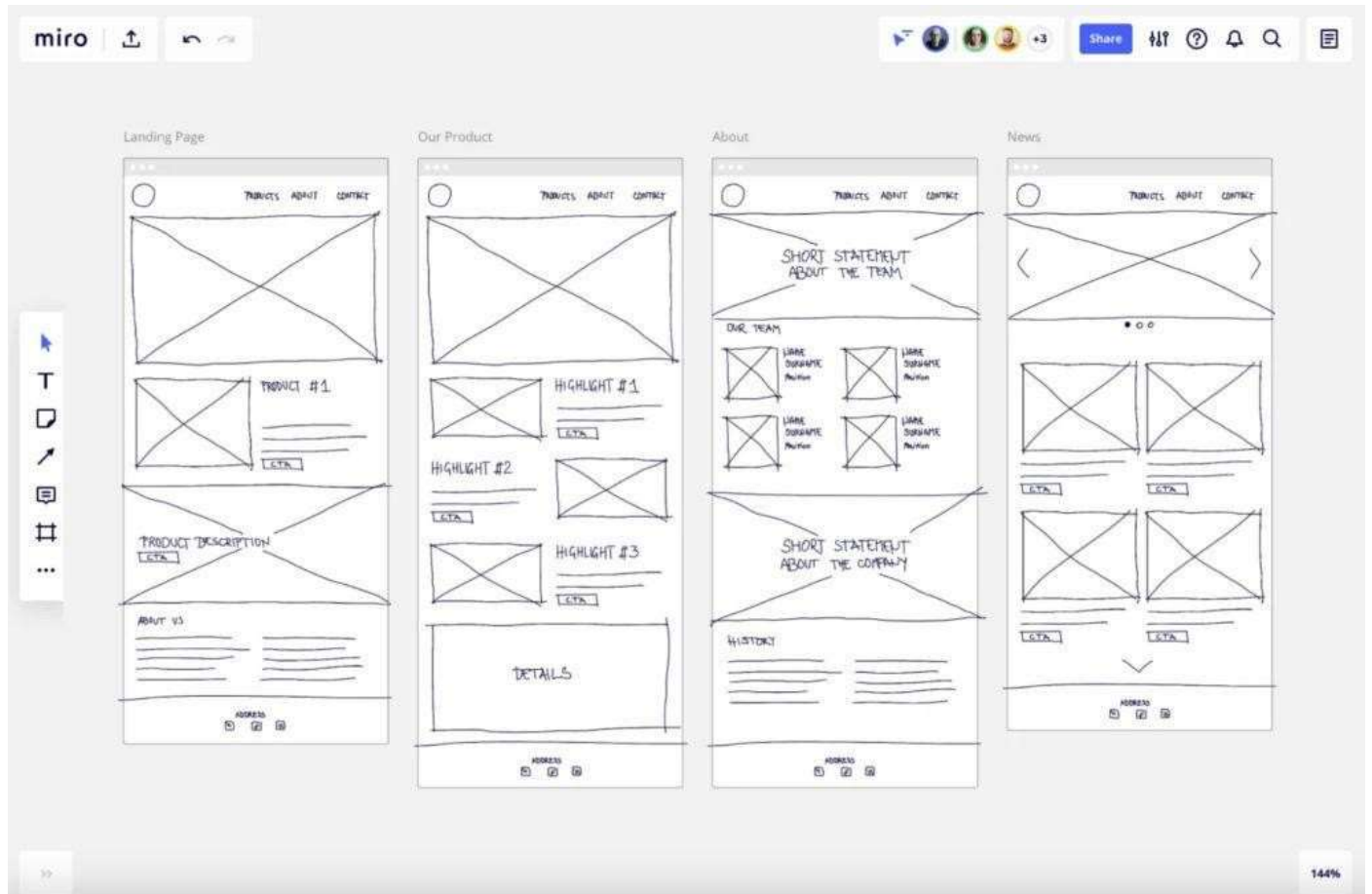
Strengths

- Allows for quick ideation without tool constraints.
- Encourages collaboration; anyone with a pen can contribute.
- Sparks creativity and is less restrictive than digital tools.

Weaknesses

- Not as precise or scalable as digital versions.
- Difficult to share and edit in remote settings.
- Lacks interactivity features present in digital tools.

eCommerce Mobile Wireframe Sketch



The digital sketch shows the main parts of a mobile eCommerce site

- Allows designers to quickly visualize product listings, calls to action, and user journeys.
- Encourages a focus on essential eCommerce elements like product images, pricing, and checkout flow.
- Great for brainstorming promotional placements or special features.

Weaknesses

- Doesn't capture the detailed intricacies of product variations or filters.
- Limited in representing interactive elements like dropdowns or sliders.
- Often lacks detailed space allocation for promotional content or ads.

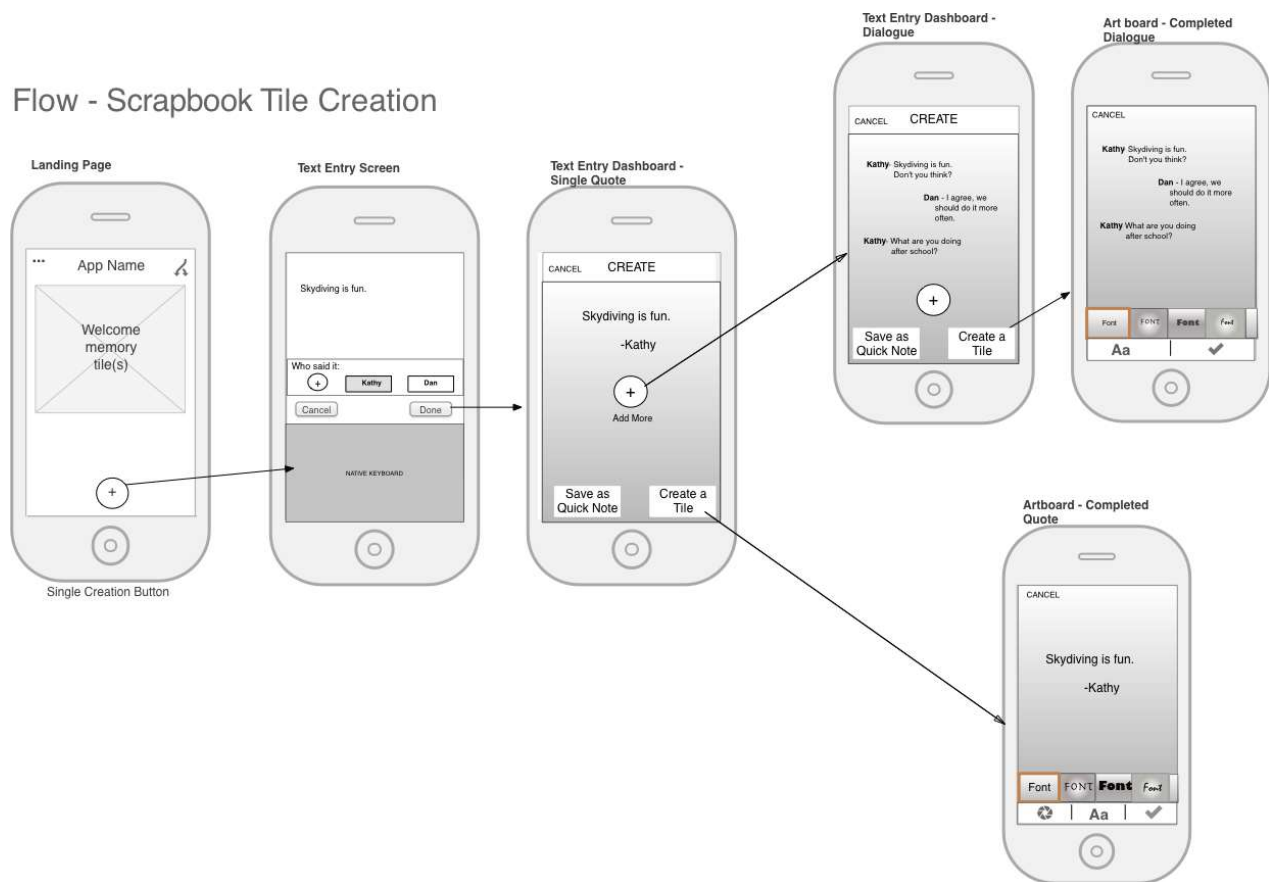
Creating Wireflows

- Wireflows are a combination of wireframes and flowcharts. They can document workflow and screen designs when there are few pages that change dynamically.
- In the UX field, wireframes are a common deliverable to show page-level layout ideas, whereas flowcharts are useful for documenting complex workflows and user tasks.
- Wireflows are used when documenting mobile, desktop, or web apps that don't have many unique pages, but instead feature a few core pages which change content (or layout) dynamically based on user interaction.

Wireflows as a Deliverable for Workflows

Definition: Wireflows are a design-specification format that combines wireframe-style page layout designs with a simplified flowchart-like way of representing interactions.

Flow - Scrapbook Tile Creation



This low-fidelity wireflow shows a simple user task. The use of screen designs, rather than abstract flowchart symbols, keeps focus on the product with which users will be interacting. While wireflows can be created in high fidelity for the purposes of communicating detailed design specifications, they are just as useful as lower-fidelity documents to discuss and communicate interaction design and user workflows.

Wireflows Document Interactions

The classic use-case for wireflows is to document the process of a user working through a common task on the product (e.g. —send a direct message to someone in your network on a social media app). At each step in the workflow a simple wireframe or high fidelity screen mock-up shows the screen available to users.

Wireflows for Collaborative Ideation

In addition to being a useful form of communication with project stakeholders and developers, wireflows also work well as a tool for collaboration between team members. Especially in Agile environments, being able to collaborate and communicate well among a crossfunctional team is critical.

BUILDING A PROTOTYPE

What is Prototyping?

Prototyping is an experimental process where design teams implement ideas into tangible forms from paper to digital. Teams build prototypes of varying degrees of fidelity to capture design concepts and test on users.

What is a Prototype?

A prototype is a simple experimental model of a proposed solution used to test or validate ideas, design assumptions and other aspects of its conceptualisation quickly and cheaply, so that the designer/s involved can make appropriate refinements or possible changes in direction.

Remarkable Reasons for Prototyping

The advantages of prototyping are that you:

- Have a solid foundation from which to ideate towards improvements—giving all stakeholders a clear picture of the potential benefits, risks and costs associated with where a prototype might lead.
- Can adapt changes early—thereby avoiding commitment to a single, falsely-ideal version, getting stuck on local maxima of UX and later incurring heavy costs due to oversights.
- Show the prototype to your users so they can give you their feedback to help pinpoint which elements/variants work best and whether an overhaul is required.
- Have a tool to experiment with associated parts of the users' needs and problems—therefore, you can get insights into less-obvious areas of the users' world (e.g., you notice them using it for additional purposes or spot unforeseen accessibility issues such as challenges to mobile use).
- Provide a sense of ownership to all concerned stakeholders—therefore fostering emotional investment in the product's ultimate success.
- Improve time-to-market by minimizing the number of errors to correct before product release.

The three primary types of prototypes based on fidelity are:

1. Low-Fidelity Prototypes: Simple and often hand-drawn representations, such as sketches or paper interfaces. They're quick to produce and useful for initial concept validations.
2. Mid-Fidelity Prototypes: More detailed than low-fidelity, these often use digital tools and give a clearer understanding of the product's aesthetics and functionality.
3. High-Fidelity Prototypes: Highly detailed and interactive, resembling the final product closely. They're used for advanced user testing and to get a realistic feel of the final design.

Low-fidelity

Example: Paper prototypes

Pros: Fast and cheap; disposable; easy to make changes and test new iterations; allow a quick overall view of the product; anyone can produce them; encourage design thinking since prototypes are visibly not finalized.

Cons: Lack of realism, so users might have a hard time giving feedback; hard to apply results from crude early versions; may be too basic to reflect the user experience of the finished product; can oversimplify complex issues; lack of interactivity deprives users of direct control; users must imagine how they would use the product.

High-fidelity

Example: Digital prototypes created on software such as Sketch or Adobe XD

Pros: Engaging—all stakeholders have the vision realized in their hands and can judge how well it matches users' needs and solves their problems; testing will yield more accurate, more applicable results; versions closest to the final product enable you to predict how users will take to it in the marketplace.

Cons: Longer/costlier to create; users are more likely to comment on superficial details than on content; after hours of work, you the designer are likely to dislike the idea of making changes, which can take considerable time; users may mistake the prototype for the finished product and form biases.

Why We Need to Prototype

Early Research isn't Everything

Research conducted during the early stages of your Design Thinking project does not tell you everything you need to know in order to create the optimal solution. Use prototyping as a form of research even before other phases in Design Thinking, allowing you to explore problem areas in interfaces, products or services, and spot areas for improvement or innovation.

Prototype to Empathise, Define, Ideate, and Test

Some of the purposes that prototypes fulfil are:

Exploring and Experimentation

You can use prototypes to explore problems, ideas, and opportunities within a specific area of focus and test out the impact of incremental or radical changes.

Learning and Understanding

Use prototypes in order to better understand the dynamics of a problem, product, or system by physically engaging with them and picking apart what makes them work or fail.

Engaging, Testing, and Experiencing

Use prototyping to engage with end users or stakeholders, in ways that reveal deeper insight and more valuable experiences, to inform design decisions going forward.

Inspiring and Motivating

Use prototypes to sell new ideas, motivate buy-in from internal or external stakeholders, or inspire markets toward radical new ways of thinking and doing.

How Prototyping Works

Bias Towards Action

One of the essential mindsets for Design Thinking is having a bias towards action:

Learning by Doing

One of the most important aspects of Design Thinking is exploring unknown possibilities and uncovering unknown insights. This is the reason the discipline places emphasis on learning and on activities that increase the learning potential of the team.

Prototyping to Test

This will be the most common prototype you will create in a design project. Create iteratively improved prototypes in order to test out solutions quickly, and then use the test results to improve your ideas.

Prototyping to Decide

Sometimes in your design project, you may face conflicting ideas from different team-mates or stakeholders. Prototyping can be an effective tool for enabling your team to compare the ideas and prevent any disagreements from developing.

When building a prototype to decide, you can see how each of the solutions will work better. You will be able to see whether the prototypes lack in some areas; for example, you may realise that the prototype would not work in the natural environment of users. Your team will also be able to see the different ideas tangibly, and hence discuss the ideas and build on them, or suggest ways to merge the best aspects of each prototype.

BUILDING HIGH-FIDELITY MOCKUPS

Mock-ups are used by designers mainly to acquire feedback from users about designs and design ideas early in the design process. Mock-ups are 'very early prototypes' made of cardboard or otherwise low-fidelity materials. The user, aided by the designer, may test the mock-up (imagining that it works) and thus provide valuable feedback about functionality/usability/understanding of the basic design idea/etc.



Mockups look like screenshots from a completed, real app, they are little more than images. Mockups are fully polished visual designs sometimes rendered in realistic devices that include branding, colors, images and typography.

Advantages:

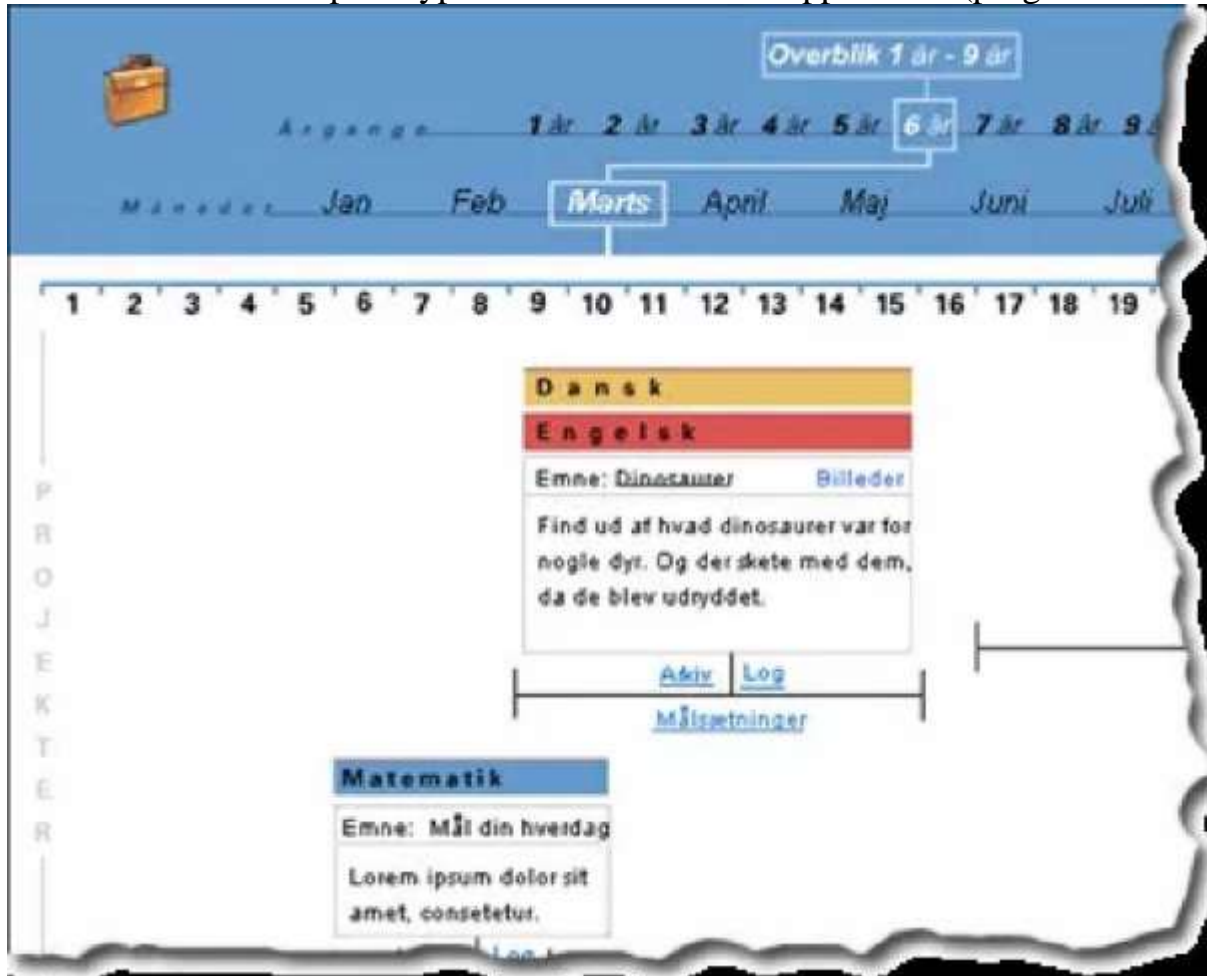
- Mock-ups incite criticism from users because they are low-cost (can be made of cardboard) and low-fidelity. If a user is presented with an early version of a system that has required substantial work, he/she is likely to be more reluctant (as well as able) to criticise it.
- When using mock-ups of cardboard or similar materials, the user and designer can collectively change the design using familiar tools such as pens, scissors etc. As such, mock-ups are a discussion medium and a discussion facilitator between designer and user.
- Not only can the mock-up function as a discussion medium between designer and user but also between the members of the design team. Thus, mock-ups may help facilitate work across disciplinary borders, bringing together a disparate team.
- Mock-ups make it possible to do usability testing early in the development process.
- Mock-ups incite and legalise experimentation as they are inexpensive to alter.
- Mockups focus on content and functionality and turn attention away from details of graphic design.

Examples of mockups



Figure 1: Cardboard Mockup of a calendar application for school kids

After a couple of tests with the mock-up, it was made into a prototype programmed in Macromedia Flash (figure 2). The prototype was more 'polished' and allowed for more interactivity. Using prototype, various scenarios of use (use cases) were tested with the users, after which the Flash prototype was built into a real application (programmed in Java).



The same calendar application, this time as a prototype made in Macromedia Flash.



A mock-up of a website

DESIGNING EFFICIENTLY WITH TOOLS

What are UX Tools?

User experience designers use UX tools—often purpose-built software—at different stages of their work. For example, designers test their assumptions using prototyping software (e.g., Balsamiq) and do usability testing with other software (e.g., Loop11). To stay competitive, brands make UX tools extremely easy to learn.

Types of tools

Questionnaire tools – To gather users’ information and feedback in quantitative research: E.g., Google Forms, SurveyMonkey.

Flowcharting tools – To help predict users’ needs, thoughts and actions and align these with business needs through flowcharts: E.g., LucidChart, OmniGraffle.

Low-fidelity prototyping/Wireframing tools – To realize more-basic visualizations of potential solutions to problem statements: E.g., Marvel, Balsamiq.

High-fidelity prototyping and user interface (UI) design tools – To realize sophisticated visualizations of design solutions: E.g., Sketch, Figma.

Usability testing tools – For qualitative research and to test the usability and accessibility of solutions (e.g., as prototypes): E.g., Loop11, UserTesting.

Handoff tools – To send completed design work to developers: E.g., Mockplus iDoc, Zeplin. Microsoft Visio is a feature-rich flowcharting tool that has a similar UI to the Microsoft Office suite of apps, which means you probably don’t have to learn to use it.

How to Approach the Right UX Tools

Before reaching for any UX tools, you should consider:








What you do matters more than the tool you choose – UX tools are instrumental to—not guarantees of—your success. As a designer, you can only make impressive and useful solutions if you know what goes where and why. UX experts and recruiters agree that skills come first. So, you should always start with important UX considerations and know what you want to achieve. Then, you select the best tool for the job at hand, even if it’s only pencil and paper. Otherwise, you’ll constrict your vision because the tool can frame your ideas and blind you to users’ needs. Beautiful, trendy-looking products won’t necessarily prove you empathize with users.

UX tools are constantly evolving – New tools and third-party add-ons to existing ones keep appearing on the market. So, it’s vital to stay grounded in the timeless principles of human psychology and your craft, instead of chasing the fleeting shadows of software’s freshest updates. Software will keep changing; users’ brains won’t.

Organizations use different tools –Company practices—including choice of UX tools—vary. Moreover, with time, they’ll switch to the next best thing for them.

Key Features, Strengths, and Use Cases of Popular UI and UX Design Tools



UI UX Design Tools	Features	Strengths	Use Cases
 Figma	Supports advanced drawing tools and reusable components, and real-time collaboration.	Cloud-based nature, seamless collaboration.	Create graphics, interactive prototypes, websites, and more with real-time collaboration.
 Adobe XD	Supports voice prototyping, auto-animate features, and repeat grid tool	Quick mockups, extensive asset, and integration with other Adobe tools.	Create digital design and prototyping user interfaces (UI) and user experiences (UX).
 Sketch	Reusable components, plugins, and shared libraries for teams.	Extensive plugin ecosystem.	Create concept pages, icons, and other web elements.
 InVision	Responsive design, rapid prototyping, and advanced animations.	Assets sharing, giving and gathering real time feedback.	Basic prototyping, collaboration (Freehand), and handoff (Inspect) capabilities.
 Balsamiq	Extensive library of UI components, drag-and-drop functionality, and low-fidelity wireframes.	Good for beginners or quick, low-fidelity wireframes.	Create quick wireframes and mockups, ideal for sketching.
 Marvel	User testing features like user flows, heatmaps, and screen recordings.	Get valuable insights with help of user testing features.	Create all sorts of mockups, prototypes, and wireframes.
 axure	Conditional logic, variables, adaptive views and automated documentation.	Advanced interaction features for complex projects and detailed prototyping.	Built for interaction design and functional prototyping.

Interaction Design Foundation
interaction-design.org

Adobe XD: This all-in-one UX/UI design software allows for wireframing, prototyping, and collaboration in the same platform.

Figma: Known for its cloud-based interface, Figma enables real-time collaboration, making it a favorite among design teams.

Sketch: Although the Sketch app is only available for macOS, Sketch has a robust set of features and an extensive plugin ecosystem, making it a powerful tool for user experience design.

InVision: InVision offers a suite of tools for prototyping, collaboration, and workflow management, making it a comprehensive solution for design teams.

Axure RP: This tool stands out for its advanced prototyping capabilities, allowing designers to create highly interactive and dynamic prototypes.

Wireframing Tools

Wireframing tools are the blueprint machines of the design process. They allow designers to create a skeletal layout of a web page or app, laying the groundwork for more detailed design work. The primary function of a wireframe is to illustrate the page structure, content arrangement, and intended functionalities.

Benefits: Wireframing software expedites the initial design phase by providing simple drag-and-drop elements, making it easy to visualize the layout. They facilitate rapid iterations and allow you to explore different layout options. Also, they establish the basic structure before you get to the detailed design. Wireframes also have the advantage of looking malleable. Colleagues and users are less reluctant to suggest changes compared to high-fidelity prototypes.

Limitations: Wireframes lack the visual design appeal of a fully designed interface. This may make it challenging for stakeholders unfamiliar with the design process to envision the final product.

Examples: Balsamiq, Sketch, and Adobe XD are popular choices for wireframing due to their simplicity and efficiency.

Prototyping Tools

Once the basic structure of the design is in place, prototyping tools come into play. These tools allow designers to create interactive mockups and showcase how the final product will work. This includes everything from button interactions and animations to transitions and user flow.

Prototyping tools are vital for usability testing, as they provide a hands-on experience of the design before any coding begins. They help identify potential issues and improvements to reduce the risk of costly changes later in development.

Benefits: Prototypes give a real feel for the user experience, enabling you to identify and resolve usability issues early in the process. They also help communicate design ideas more effectively to stakeholders for better feedback and alignment.

Limitations: Creating high-fidelity prototypes can be time-consuming. Moreover, some prototyping tools may have limitations in simulating complex interactions. It may lead to discrepancies between the prototype and the final product.

Examples: Figma, InVision, and Axure RP are renowned for their prototyping capabilities. They provide a wide array of interactive elements and transition effects.

Graphic Design Software

For the visual refinement of a design, graphic design software is the tool of choice. They offer a high level of control over the visual elements of a design, from typography and color to imagery and iconography. They are essential for creating the final look and feel of a product.

Benefits: Graphic design software offers extensive features that help create and manipulate visual elements. They allow you to achieve the exact look and feel they envision. They also support various file formats to help you work with multiple assets.

Limitations: Graphic design tools often have a steep learning curve due to their complex functionality. Moreover, they typically focus on creating static assets and may lack the collaborative and interactive features found in UX/UI design software.

Examples: Adobe Illustrator and Photoshop are industry standards for graphic design. Affinity Designer is another powerful alternative that is gaining popularity.

User research tools

User research tools are critical instruments used to gather insights about users' behaviors, needs, and motivations. These tools help understand the user journey, identify pain points, and craft a user experience that is intuitive and satisfying for the end user.

Benefits: User research tools provide data-driven insights that can inform every stage of the

design process to create user-centric products. They allow researchers to test hypotheses, validate designs, and understand user needs and preferences in a detailed manner.

Limitations: The effectiveness of user research tools largely depends on the sample size and representativeness of the user group involved in the research. These tools also require time and resources to set up and analyze. Sometimes, the data can be overwhelming and challenging to interpret without the right expertise.

Examples: SurveyMonkey and Google Forms are simple, accessible tools for conducting surveys and Lookback for user interviews. UserTesting provides audio and video feedback from users interacting with your designs. UX research tools like Hotjar offer heatmaps, session recordings, and other tools for understanding user behavior on a website. For early design, products such as Optimal Workshop provide card sorting, tree sorting and first-click testing.

Criteria for Choosing UI and UX Design Tools

- Ease of Use
- Features
- Cost
- Platform Compatibility

INTERACTION PATTERNS

An interaction design (ID) pattern is a general repeatable solution to a commonly-occurring usability problem in interface design or interaction design. An ID pattern usually consists of the following elements:

- **Problem:** Problems are related to the usage of the system and are relevant to the user or any other stakeholder that is interested in usability.
- **Use when:** a situation (in terms of the tasks, the users and the context of use) giving rise to a usability problem. This section extends the plain problem-solutions dichotomy by describing situations in which the problems occur.
- **Principle:** a pattern is usually based on one or more ergonomic principles such as user guidance, or consistency, or error management.
- **Solution:** a proven solution to the problem. A solution describes only the core of the problem, and the designer has the freedom to implement it in many ways. Other patterns may be needed to solve sub problems.
- **Why:** How and why the pattern actually works, including an analysis of how it may affect certain attributes of usability. The rationale (why) should provide a reasonable argument for the specified impact on usability when the pattern is applied. The why should

describe which usability aspects should have been improved or which other aspects might suffer.

- Examples: Each example shows how the pattern has been successfully applied in a real life system. This is often accompanied by a screenshot and a short description.
- Implementation: Some patterns provide implementation details.

Example interaction design pattern: Multi-level Undo

Problem Users do actions they later want reverse because they realized they made a mistake or because they changed their mind.

Use when You are designing a desktop or web-based application where users can manage information or create new artifacts. Typically, such systems include editors, financial systems, graphical drawing packages, or development environments. Such systems deal mostly with their own data and produce only few non-reversible side-effects, like sending of an email within an email application. Undo is not suitable for systems where the majority of actions is not reversible, for example, workflow management systems or transaction systems in general.

Both novice and expert users may want to reverse their actions, either because of mistakes or changes in intention. Expert users may want to use the history of their actions for more specific manipulation of the data in the application. For example, in a graphical modeling application, users may want to undo work on some specific object while keeping later work done on other objects.

Principle Error Management (Safety) (Norman, 1988)

Solution Maintain a list of user actions and allow users to reverse selected actions.
Each 'action' the user does is recorded and added to a list.

This list then becomes the 'history of user actions' and users can reverse actions from the last done action to the first one recorded. This is also called a Linear Multi-level Undo.

- **Interacting with the history**

There are two variations on how to show the history of actions to the users. First there is the standard 'office-like' way where the 'Edit' menu contains both 'Undo' and 'Redo' functions with their keyboard shortcuts. Often there is also a widget in the toolbar that can show the last items in the history. By dragging the selection in the list, actions can be undone. A second variant is to work with primarily with the history list itself and moving a slider or scrollbar to move back in history and undo actions. Photoshop uses such a variant.

- **Displaying actions**

Actions in the history are usually displayed using a text label such as 'Create circle', 'Typing', 'New contact'. Such labels only name the function and not the object the functions work on. In some applications it may be better to include the object and the parameters as well, for example 'Change-color Circle12 to Red'.

- **Granularity of actions**

When designing Undo it is important to determine the desired granularity of actions. For example, it is usually not desired to record each key press in a text editor as an action. Instead, typing a word is used as a unit of action. Designers need to determine what unit of action is appropriate in the application.

- **Non-reversible actions**

Although most actions in the application may be reversible, it is very likely that some actions will not be

reversible. For example, printing, saving, doing a payment, or downloading an object. For actions that are non-reversible and 'negative' of nature (like paying or destroying something), need to show the user a Warning Message and not add the action to the history.

- **Selective undo**

In some cases, it can be meaningful to allow single actions from the history to be deleted. This is the case when a certain 'episode' of work must be deleted or undone while keeping work that has been done later on. Selective undo is conceptually much more difficult than linear undo since there is a notion of 'dependency between actions' that determines the consequences of undoing a particular action. For example, if a 'create circle' action is undone at some point in the history, subsequent actions in the history working on that object lose their meaning and must be deleted. There are many semantic issues with selective undo

Object-based Undo

Object-based Undo can sometimes be considered as an alternative to Selective Undo. With Object-based Undo, each object has its own action history. Upon selecting the object, the users can undo actions done on the object. Naturally, this requires the application to have a clear concept of an 'object' and is therefore not applicable for bitmap editors.

Multi-user undo

If the application is a multi-user application and uses undo, the application must distinguish between local actions and global actions. That leads to multiple histories and requires special semantics for what happens when undoing actions.

Why

Offering the possibility to always undo actions gives users a comforting feeling. It helps the users feel that *they* are in control of the interaction rather than the other way around. They can explore, make mistakes and easily go some steps back, which facilitates learning the application's functionality. It also often eliminates the need for annoying warning messages since most actions will not be permanent

Examples

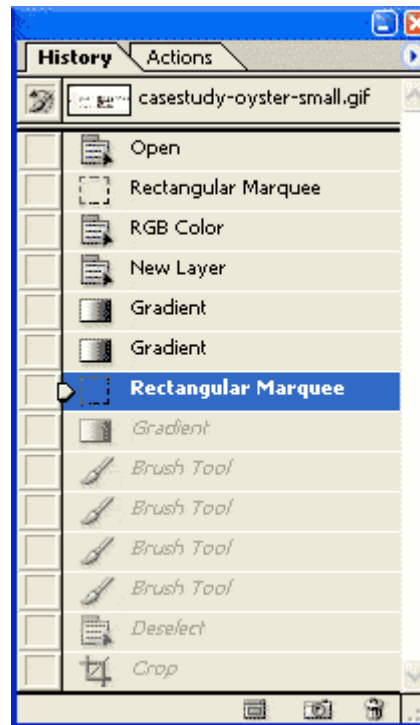
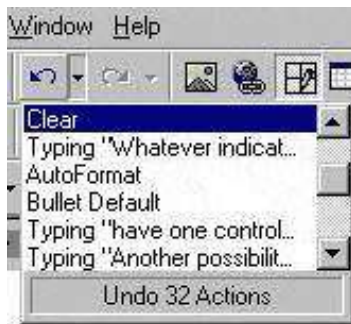
As in all MS Office applications, in Word 2000 the users can see the history of their actions and undo one or more of them. The actions are briefly described and the users can select a range of actions to be undone. After selecting undo, users can even redo the actions.

In Photoshop a selective undo is also possible. By moving the slider, users can do the normal multi-level undo but they can also drag an action into the trashcan and thereby do a selective undo. Operations that depended on that action are

automatically deleted
as well of they are
not relevant
anymore.

Implementation

Most implementations of multi-level undo are based on the Command (Gamma et al 1995) pattern. When using the Command pattern, most functionality is encapsulated in Command objects rather than in other controlling classes. The idea is to have a base class that defines a method to "do" a command, and another method to "undo" a command. Then, for each command, you derive from the command base class and fill in the code for the do and undo methods. The "do" method is expected to store any information needed to "undo" the command. For example, the command to delete an item would remember the content of the item being deleted



CONDUCTING USABILITY TESTS

What is Usability Testing?

Usability testing is the practice of testing how easy a design is to use with a group of representative users. It usually involves observing users as they attempt to complete tasks and can be done for different types of designs. It is often conducted repeatedly, from early development until a product's release.

Usability Testing Leads to the Right Products

Through usability testing, you can find design flaws you might otherwise overlook. When you watch how test users behave while they try to execute tasks, you'll get vital insights into how well your design/product works. Then, you can leverage these insights to make improvements. Whenever you run a usability test, your chief objectives are to:

- 1) Determine whether testers can complete tasks successfully and independently.
- 2) Assess their performance and mental state as they try to complete tasks, to see how well your design works.

- 3) See how much users enjoy using it.
- 4) Identify problems and their severity.
- 5) Find solutions.

While usability tests can help you create the right products, they shouldn't be the only tool in your UX research toolbox. If you just focus on the evaluation activity, you won't improve the usability overall.

There are different methods for usability testing. Which one you choose depends on your product and where you are in your design process.

Usability Testing is an Iterative Process

To make usability testing work best, you should:

1) Plan –

- a. Define what you want to test. Ask yourself questions about your design/product. What aspect/s of it do you want to test? You can make a hypothesis from each answer. With a clear hypothesis, you'll have the exact aspect you want to test.
- b. Decide how to conduct your test – e.g., remotely. Define the scope of what to test (e.g., navigation) and stick to it throughout the test. When you test aspects individually, you'll eventually build a broader view of how well your design works overall.

2) Set user tasks –

- a. Prioritize the most important tasks to meet objectives (e.g., complete checkout), no more than 5 per participant. Allow a 60-minute timeframe.
- b. Clearly define tasks with realistic goals.
- c. Create scenarios where users can try to use the design naturally. That means you let them get to grips with it on their own rather than direct them with instructions.

3) Recruit testers – Know who your users are as a target group. Use screening questionnaires (e.g., Google Forms) to find suitable candidates. You can advertise and offer incentives. You can also find contacts through community groups, etc. If you test with only 5 users, you can still reveal 85% of core issues.

4) Facilitate/Moderate testing –Set up testing in a suitable environment. Observe and interview users. Notice issues. See if users fail to see things, go in the wrong direction or misinterpret rules. When you record usability sessions, you can more easily count the number of times users become confused. Ask users to think aloud and tell you how they feel as they go through the test. From this, you can check whether your designer's mental model is accurate: Does what you think users can do with your design match what these test users show?

If you choose remote testing, you can moderate via Google Hangouts, etc., or use unmoderated testing. You can use this software to carry out remote moderated and unmoderated testing and have the benefit of tools such as heatmaps.

Keep usability tests smooth by following these guidelines.

1) Assess user behavior – Use these metrics:

Quantitative – time users take on a task, success and failure rates, effort (how many clicks users take, instances of confusion, etc.)

Qualitative – users' stress responses (facial reactions, body-language changes, squinting, etc.), subjective satisfaction (which they give through a post-test questionnaire) and perceived level of effort/difficulty

2) Create a test report – Review video footage and analyzed data. Clearly define design issues and best practices. Involve the entire team.

Overall, you should test not your design's functionality, but users' experience of it. Some users may be too polite to be entirely honest about problems. So, always examine all data carefully.

To conduct usability testing effectively:

Start by defining clear, objective goals and recruit representative users.

Develop realistic tasks for participants to perform and set up a controlled, neutral environment for testing.

Observe user interactions, noting difficulties and successes, and gather qualitative and quantitative data.

After testing, analyze the results to identify areas for improvement.
Usability testing is a popular UX research methodology.

In a usability-testing session, a researcher (called a —facilitator‖ or a —moderator‖) asks a participant to perform tasks, usually using one or more specific user interfaces. While the participant completes each task, the researcher observes the participant’s behavior and listens for feedback.

The phrase —usability testing‖ is often used interchangeably with —user testing.‖

(One objection sometimes raised against the phrase —user testing‖ is that it sounds like researchers are testing the participant — we never test the user, only the interface. However, the term is intended to mean testing with users, which is exactly the point of empirical studies.)

Why Usability Test?

The goals of usability testing vary by study, but they usually include:

- Identifying problems in the design of the product or service
- Uncovering opportunities to improve
- Learning about the target user’s behavior and preferences

Usability testing helps us to uncover problems, discover opportunities, and learn about users. Why do we need to do usability testing? Won’t a good professional UX designer know how to design a great user interface? Even the best UX designers can’t design a perfect — or even good enough — user experience without iterative design driven by observations of real users and of their interactions with the design.

There are many variables in designing a modern user interface and there are even more variables in the human brain. The total number of combinations is huge. The only way to get UX design right is to test it.

Elements of Usability Testing

There are many different types of usability testing, but the core elements in most usability

tests are the facilitator, the tasks, and the participant.

A usability-testing session involves a participant and a facilitator who gives tasks to the participant and observes the participant's behavior.

The facilitator administers tasks to the participant. As the participant performs these tasks, the facilitator observes the participant's behavior and listens for feedback. The facilitator may also ask followup questions to elicit detail from the participant.

In a usability test, the facilitator gives instructions and task scenarios to the participant. The participant provides behavioral and verbal feedback about the interface while he performs those tasks.

Facilitator

The facilitator guides the participant through the test process. She gives instructions, answers the participant's questions, and asks followup questions.

The facilitator works to ensure that the test results in high-quality, valid data, without accidentally influencing the participant's behavior. Achieving this balance is difficult and requires training.

(In one form of remote usability testing, called remote unmoderated testing, an application may perform some of the facilitator's roles.)

Tasks

The tasks in a usability test are realistic activities that the participant might perform in real life. They can be very specific or very open-ended, depending on the research questions and the type of usability testing.

Examples of tasks from real usability studies:

Your printer is showing -Error 5200ll. How can you get rid of the error message?

You're considering opening a new credit card with Wells Fargo. Please visit wellsfargo.com and decide which credit card you might want to open, if any.

You've been told you need to speak to Tyler Smith from the Project Management department. Use the intranet to find out where they are located. Tell the researcher your answer.

Task wording is very important in usability testing. Small errors in the phrasing of a task can cause the participant to misunderstand what they're asked to do or can influence how participants perform the task (a psychological phenomenon called priming).

Participant

The participant should be a realistic user of the product or service being studied. That might mean that the user is already using the product or service in real life. Alternatively, in some cases, the participant might just have a similar background to the target user group, or might have the same needs, even if he isn't already a user of the product.

Participants are often asked to think out loud during usability testing (called the —think-aloud method). The facilitator might ask the participants to narrate their actions and thoughts as they perform tasks. The goal of this approach is to understand participants' behaviors, goals, thoughts, and motivations.

Types of Usability Testing

Qualitative vs. Quantitative

Usability testing can be either qualitative or quantitative.

Qualitative usability testing focuses on collecting insights, findings, and anecdotes about how people use the product or service. Qualitative usability testing is best for discovering problems in the user experience. This form of usability testing is more common than quantitative usability testing.

Quantitative usability testing focuses on collecting metrics that describe the user experience. Two of the metrics most commonly collected in quantitative usability testing are task success and time on task. Quantitative usability testing is best for collecting benchmarks.

The number of participants needed for a usability test varies depending on the type of study. For a typical qualitative usability study of a single user group, we recommend using five participants to uncover the majority of the most common problems in the product.

Remote vs. In-Person Testing

Remote usability tests are popular because they often require less time and money than in-person studies. There are two types of remote usability testing: moderated and unmoderated.

Remote moderated usability tests work very similarly to in-person studies. The facilitator still interacts with the participant and asks her to perform tasks. However, the facilitator and participant are in different physical locations. Usually, moderated tests can be performed using screen-sharing software like Skype or GoToMeeting.

Remote unmoderated remote usability tests do not have the same facilitator-participant interaction as an in-person or moderated tests. The researcher uses a dedicated online remote-testing tool to set up written tasks for the participant. Then, the participant completes those

tasks alone on her own time. The testing tool delivers the task instructions and any followup questions. After the participant completes her test, the researcher receives a recording of the session, along with metrics like task success.

Remote Unmoderated: Flow of Information



SYNTHESIZING TEST FINDINGS

Synthesis, is about creatively putting together your analysis and research in order to form whole ideas.

Methods which will Help you Synthesise Your Research

Share Inspiring User Stories

One way of making sense of your data is to share with your team the most inspiring stories you've heard from the people you're designing for. Think about user stories or experiences that have stuck with you: stories which surprised you, made you curious, or verified or falsified your assumptions.

Best Practice: Construct Your User Story Madlib

As a [who are they], he/she wants to [what do they want to do], so that [their end goal].

Example: As a freelance consultant, Peter wants to easily schedule meetings, so that he can ensure his schedule is always organised and effective.

Example: As a corporate web designer, Lisa wants to improve the company's website, so that users can easily find and get access to what they need.

Make sense in your Research by Creating Maps

Mapping helps plot experiences, customer journeys, thought processes, a series of activities or actions and other related behaviour, as well as feelings, in one place. It would be useless having reams of data and not be able to understand the essence of it by extracting significant meaning from the data.

Affinity Diagrams and Space Saturate and Group – Clustering and Bundling Ideas and Facts

The —space saturate and group|| method's goal is to get all of your observations and findings into one place: Immerse yourself in the chaos of information you've gathered during your research Empathise phase.

Write on post-its, draw your insights, tell stories, and share artifacts.

The term —saturate|| relates to the way everyone covers or saturates the —space|| with their images and notes in order to create a wall of information to inform and start —grouping|| the following problem-defining process.

Empathy Map

An Empathy map will help you understand your user's needs whilst developing a deeper understanding of the person you are designing for. There are many techniques available for you to develop this type of empathy. An Empathy Map is one method that will help you define and synthesise your observations from your fieldwork and research phase, and draw out insights about your users' needs

Personas

Personas are fictional characters, which, based on your research, you create to represent the different user types that might use your service, product, site, or brand in a similar way. When you create personas you will understand your users' needs, behaviours, and goals – it will help you step out of yourself. It helps you recognise that different people have different needs and expectations, and it helps you to identify with the user you're designing for.

Stories

User stories are short stories that aim to insert the Persona into a situation in which he or she is using your product or aiming to fulfill the goal you seek to match. At this point, you should ignore minor details and get to the root of the who, what and why of the situation

Scenarios

Scenarios expand on user stories and fill in many valuable human and environmental factors, which flesh out the story providing much more meaning for exploring various aspects of the solution. Scenarios provide us with the context within which our personas function, and how their experiences and needs play out.

Define Your Point Of View (POV)

By the end of your Define mode, you should end up creating an actionable problem statement, also known as your Point of View (POV). You form your Point of View by extracting the most important insights about your users' core human needs that you should fulfill within the problem area you're investigating and designing for

Frame Your POV with How Might We Questions

When you've defined your design challenge in a POV, you can start opening up for ideas to solve your design challenge by asking —How Might We. You're now moving on to the next phase of the Design Thinking process, the third phase: Ideation. You start by rephrasing and framing your POV as several questions by adding —How might we at the beginning of the POV.

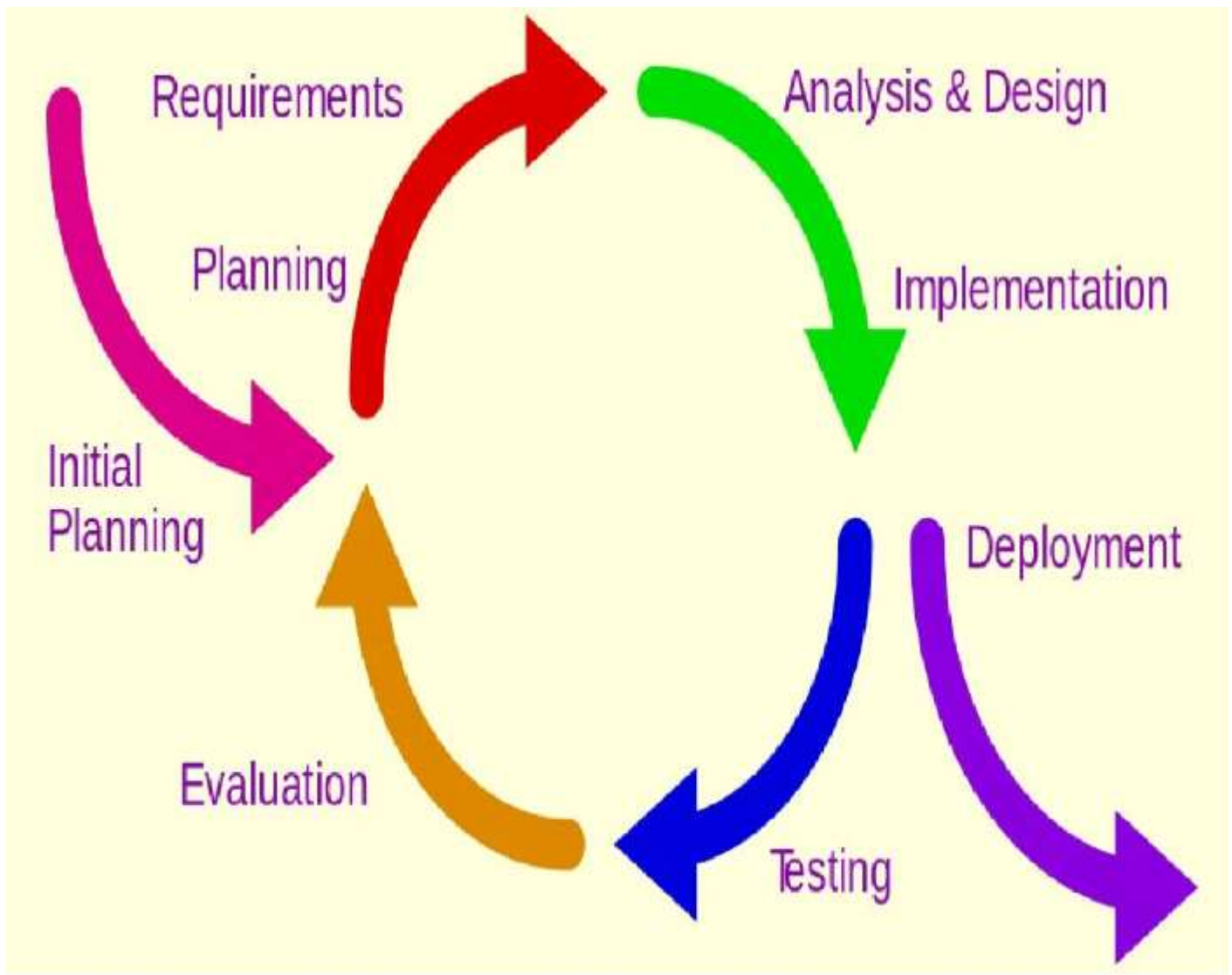
Why-How Laddering

Designers will use why to progress to the top of the so-called Why-How Ladder where the ultimate aim is to find out how you can solve one or more problems. In other words, you could say that the Why-How Laddering starts with asking Why to work out How they can solve the specific problem or design challenge.

PROTOTYPE ITERATION

The iterative design process is a simple concept. Once, through user research, you have identified a user need and have generated ideas to meet that need, you develop a prototype. Then you test the prototype to see whether it meets the need in the best possible way. Then you take what you learned from testing and amend the design. Following that, you create a new prototype and begin the process all over again until you are satisfied that you've reached the best possible product for release to the market.

This iterative process is often called —rapid prototyping‖ or —spiral prototyping.‖



When Do I Use Iterative Design?

Iterative design can be used at any phase of the design process, including when the product has already been launched in the market and you are looking to create improvements in that product.

Why Does Iterative Design Exist?

- User research prior to product development has to deal with what people say they will do in most cases. There's no product available to examine what they actually do. So the first model of any given product is likely to reflect the —theoretical use of a product rather than its actual use.

- Iteration recognizes that this is the case and allows you to quickly develop a prototype (which is simply a model of the finished product) and then test that prototype with users to determine what they actually do when confronted with the product.
- Because prototyping is relatively cheap to do (for example the lowest fidelity prototype might consist of nothing but pen and paper sketches) and relatively quick to create, it makes for a cost-effective system to improve designs without doing all the hard work (development) and then discovering that the difference between what users said they would do and what they actually do is large and expensive to overcome.

The Benefits of Iterative Design

- It allows for rapid resolution of misunderstandings within the project team and established clarity early in the development lifecycle.
- It brings out user feedback to ensure that system requirements meet user needs.
- It can help with client relationships to show the evolution of a design rather than —dumping a finished product on them.
- It gives the development team some certainty that their efforts are being focused on adding value for users.
- It provides regular testing which can provide a strong desired performance framework for acceptance testing.
- It allows for easy incorporation of —lessons learned in the final product.
- It gives stakeholders better visibility of progress at each iteration.

Examples of Iterative Design

Best example of iterative design on the web is the use of Wikis. Wikipedia, for example, contains user generated content. Anyone is free to come along and improve on that content at any time. It's easy for a reviewer (or editor) to visit that improvement and make a decision as to whether the change is an improvement or whether it takes something away instead.

