

Design Thinking Toolkit

A set of tools to create designs your users will love

Created for students using the UXsyllab.us site

Observation

Site visits show you real user pain



WHAT

See how people use your product. Watch the tasks they perform. Find the small annoyances and pain points.

WHY

Users lie. Unfortunately, users aren't very good at vocalizing their tasks and they have often become so worn down by the annoyances that they don't realize there could be an alternative.

So if you directly ask users what you should fix, or worse still how you should fix it, they'll give you answers that are incomplete and don't touch on the root cause of the problem.

Watching them work shows us their true task and exposes the pain points that need to be fixed.

HOW

Do:

- Write down what people say and do
- Take photos of the environment if you're allowed
- Smile
- Ask open-ended questions
- Ask for examples (times when "it" happened)

Don't:

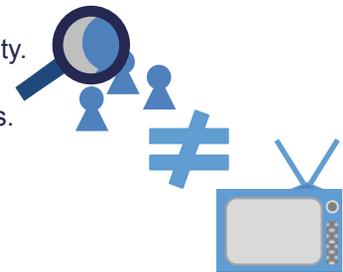
- Engage in conversation
- Write down solutions or bug fixes. If you're doing this, you're distracted.
- Sell them on your cool product idea
- Ask them to predict the future

Active observation: lean forward, write lots

User research is a lean-forward activity: you have to remain actively engaged. It's very different from watching TV: a lean-back, passive activity.

By the end of the observation session, your writing hand should be cramping up. Remember that the best notes are facts, not interpretations. Write down what users say and what they do. You can interpret those quotes and actions later.

It helps to think of yourself as an understudy to the person you are observing. If you had to act their role, what would you need to know?



TIPS



Know what task you want to watch

Go in with clear goals. What interactions are of interest? Ask participants to perform those tasks for you.



Let the participant do the talking

Ask participants to think out loud. Leave questions until a natural break in the work.



Observe in pairs

Different people see different things. One may grasp the technical issues, another might be good at reading body language.



Leave the video camera at home

You end up relying on the recording rather than taking notes. You'll never find time to watch it, so those notes are lost.



Bring a still camera

Take pictures of artifacts. Artifacts are anything that people use to help them do their work.

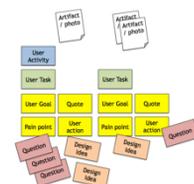


Clean up notes immediately

Share what you learned as quickly as possible. Much of the experience will still be in your head, not on paper.

WHAT'S NEXT?

Share the results of all the visits. Synthesize your findings on an *Experience Map* so the whole team can get the big picture.



Questioning skills



Listen, Probe and Validate

WHAT

Users will make lots of feature requests during research sessions. Listen, Probe and Validate to make sure you solve the underlying issue.

WHY

Users often give us feature requests without telling us the underlying problem they are trying to solve. Feature requests are problems bundled up as solutions, but in this instance the solutions have been designed by users who might not have the full picture.

To create a truly useful design solution we need to fully understand the problem, the context, and the user's goals. In other words, we have to understand their *intent*.

HOW

Listen	Pay attention to feelings (empathy), facts (versus what people say they do), process (actual versus official steps) and actors (who/what else is involved).
Probe	with open ended questions: "Tell me about the last time that happened", "Give me an example of...", "Was that what you expected?", "Does that happen often?" Dive deeper by asking for more detail. Move forwards by asking about the next steps.
Validate	with closed questions: "So exactly how long did that take?", or by paraphrasing what you just heard

Example: site visit to a shipping area of a warehouse:

Customer: "For items that I've already picked, I want a way to turn them all a different color so I know they're done."

Probe: "Can you describe the last time you wanted to do this?"

Customer: "It happens all the time - when I pick one order off the shelves, I'll grab the components I need for the other orders on the screen, because I don't want to have to come all the way back here each time."

Probe: "How do you keep track of it at the moment?"

Customer: "I print off the screen and take the paper with me. When I get the parts, I cross them off the printout."

Validate: "If I understand correctly, you want a way to minimize the effort of picking each item, regardless of what order the item belongs to."

Customer: "Exactly. But it's important to keep them in separate boxes otherwise you'll end up sending them out in the wrong orders."

By listening, probing and then reflecting the user's statements back to them, you can quickly get to the root cause of an issue.

You *could* decide that you need to create a checkbox and button combination that allows users to select items to change them red rather than green. However, does this interpretation of the feature request *really* solve the problem? Find out using the Listen, Probe, Validate process.

This leads to a totally different, more globally applicable interpretation of the initial feature request: You need a way to sort the picking list based on the location of items in the warehouse, rather than by individual orders, but with easy reference to the order numbers.

Experience Map

Pain point

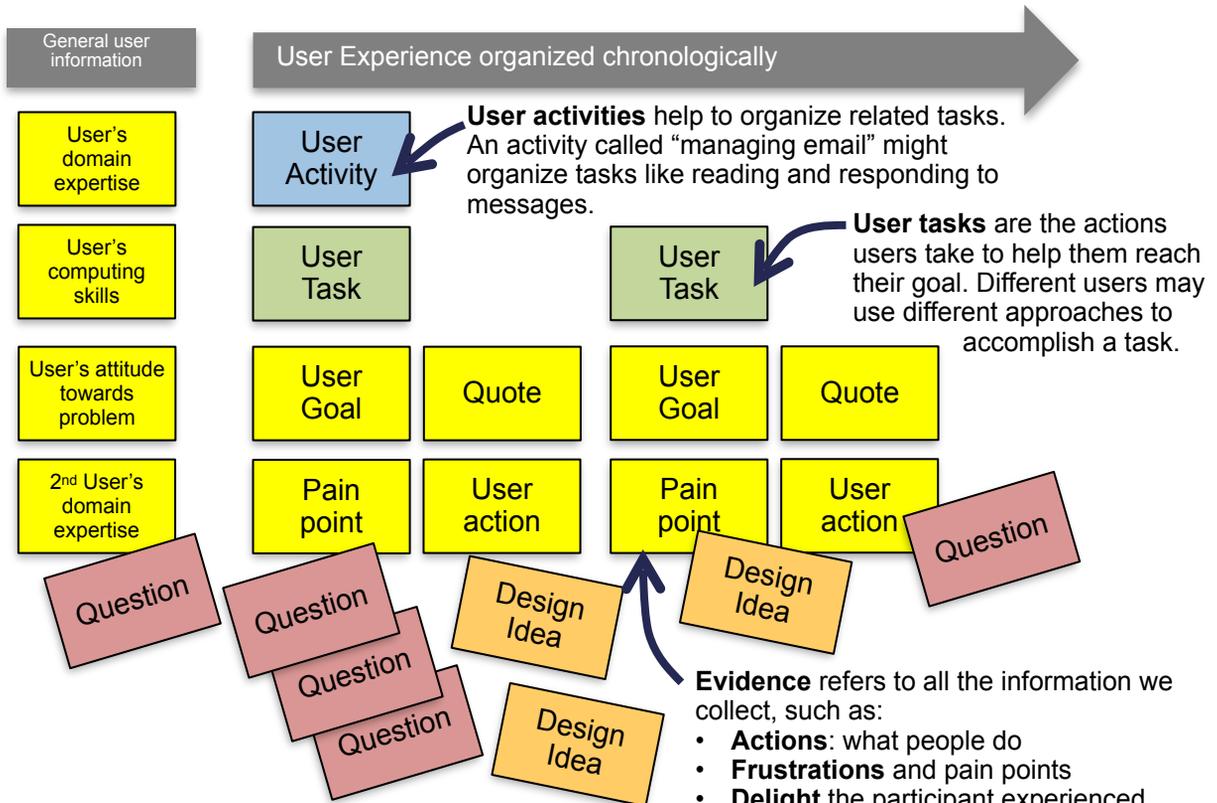
Question

Create a visual story of users' tasks and pain points

WHAT

An *Experience Map* organizes the data you gather from research and observation. Activities are arranged in the order users engage in them. The map incorporates users' motivations, frustrations and delights. It's also easy to add other relevant details like who they collaborate with and copies of any documentation they use.

As you build the map you'll identify questions to answer in subsequent research, and you may think of feature ideas as well.



HOW

- 1 Set up the mapping area**
 - Cover a whole wall in poster or butcher paper
 - Left hand space is for general observations about users
 - Identify the rough start and end points – from where users start thinking about their task until they have accomplished their goal
- 2 Comb notes for details**
 - Transcribe observations to yellow stickies
 - Use quotes wherever possible
 - Start with 10-20 notes
- 3 Start making the map**
 - In a group, add notes to the map
 - Place each note where you think it belongs
 - Group similar notes together
- 4 Reorganize and categorize**
 - Remove duplicates, agree on the groupings, and name them
 - Put task groupings in the right order
 - Create activity labels for task groups

WHAT'S NEXT?

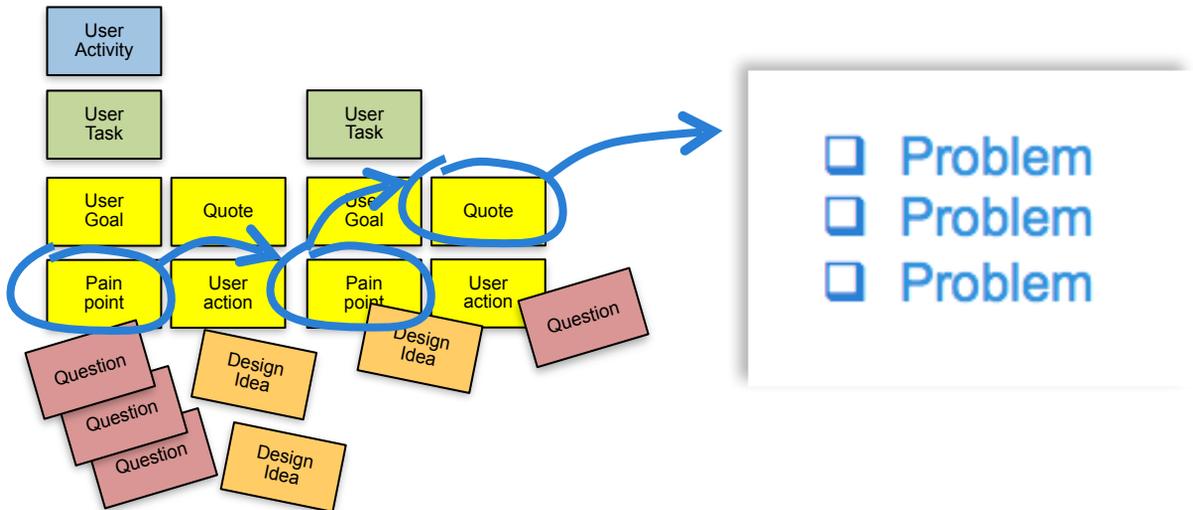
Identify the biggest pain points and create *Problem Statements* that describe issues that need resolution.

Problem Statements

Actionable product ideas

WHAT

Now that we know so much about our users, it's time to describe their pain. The *Problem Statement* moves us from problem identification to ideation.



WHY

It's often hard to know where to focus development effort. Using problem statements that are based on research means the solutions you create are more likely to be actionable and to resonate with users.

HOW

Walk the *Experience Map* following the process from beginning to end.

1. Highlight where the problems lie. Pay special attention to problems that are small enough for users to gloss over (or create work-arounds) but that still cause pain. If you can solve these problems, you are on the road to creating user delight.
2. Create a list of problem areas.
3. Write down the severity of each problem for your key persona.
4. Estimate the benefit to the business inherent in fixing each problem.
5. Prioritize the list, balancing severity and business benefit.

Although it's hard, remember to keep your discussion centered on problems and not solutions at this point.

WHAT'S NEXT?

Create *Personas* to serve as the focal point for our design and development effort.

Identify multiple potential solutions to each problem statement, using the *Design Charrette* technique.

Design Charrette



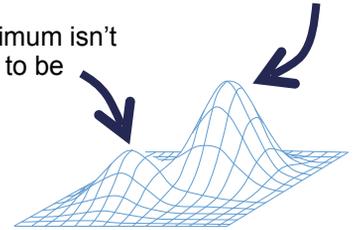
Ideate: identify many possible design solutions

WHAT

Quickly create and refine multiple potential solutions to arrive at the best option. Reduce “groupthink” and the local maxima problem.



Ideate to get past your preconceptions
Local maximum isn't best place to be



WHY

To reduce the risk of building a solution that won't succeed in the market, it's important that we consider multiple options and proceed with the most likely solution, or combination of them.

Our first solutions are rarely our best. A single person can only leverage their own individual understanding of the problem. Their design ideas are bound by personal experience. Working as a team to ideate and consider multiple solutions helps us get to a more well-rounded product.

HOW

Unlike brainstorming, for this exercise we start on our own and only come together once we've described our individual design solutions.

1. Review the problem statement and user research as a team.
2. On your own, sketch your product solution ideas using whichever technique works for you – written scenarios, design comics, storyboards, or UI sketches.
3. Together as a group, critique each design idea and identify the concepts you want to be sure to keep.
4. Work in pairs to synthesize the good concepts and come up with a next better design.
5. Together as a group critique these new designs, and make some final decisions about the solutions you'd like to carry through into prototypes.

WHAT'S NEXT?

Move from ideation to something more solid by describing the *User Scenarios* that your design ideas would enable.

User Scenarios

Refine a solution by imagining its user experience

WHAT

A *User Scenario* imagines a specific user (persona) working with a hypothetical product to reach a specific goal. An ideal scenario describes a typical path through the product, not just the simplest "happy path."

Bob. Pricing

PERSONARY

- Professional, Homeowner
- Determined, Frustrated
- Resourceful
- His Dad is bigger + head
- Consumerious
- Service oriented for son

ABOUT

- Computer savvy
- Self-proclaimed car. Buyer
- Expert
- Older
- Baby Boomer, Father
- Wants to get the best deal
- Uses multiple search engines for research
- Going to decrease sales
- Prefers price over
- Values TMV as a barrier

Scenario

- 1 Bob needs to buy a car for his son, who is leaving for college!
- 2 Bob is SAVVY, & KNOWS how much he wants to spend,
- 3 Bob gets minimum req's from his son.
- 4 His son wants a black car, with mpg info
- 5 Bob browses MULTIPLE sites, magazines & consumer reports articles on his options
- 6 ONCE RESOLUTE in his options, Bob + his son look at & test DRIVE EACH CAR
- 7 Bob is ready to research pricing info, goes to his home PC, & NAVIGATES to the COMPARED tool TO MAKE SURE his #1 choice, the Honda Fit, doesn't leave anything on the table to the OTHER CHOICES. After his review, he feels confirmed in his choice.
- 8 HE CLICKS ON THE LINK, TAKING HIM TO THE CONFIGURATOR..
- 9 Bob enters his zip & configures his car.
- 10 Bob notices two things as he configures his Fit.
A: TMV Price Adjusts AS HE configures the Fit.
B: The "Experience Shaping Widget" is also modifying on the fly, showing CARDS with info from Local Buyers who have recently bought A similarly configured vehicle!!
- 11 Bob Now has both a TMV + Reachable price for his Fit. He also notices the same individual who provided the price, also reviewed his dealership!
- 12 Bob clicks on the 'more' link on the card, & is presented with a SPLASH SCREEN, asking Bob to review + share his experience after he is done with his purchase.
- 13 Bob clicks the splash + collects the dealer phone # + specification who was well reviewed.
- 14 Bob calls, negotiates, & buys the Fit!
- 15 After buying the Fit, Bob reviews to Edmunds to input his experience.
- 16 HE is presented with the opportunity to become a user/login, or to simply enter his Anonymous pricing/demog info.
- 17 Bob chooses to simply enter his Anonymous data, so his experience helps reinforce the confidence of other Fit shoppers in his neighborhood, city or town!

Persona

Scenario

WHY

It's easy to mistake a good looking user interface design for one that effectively helps a user reach a goal. Thinking through the user experience in words first helps us imagine that experience without getting lost in UI details.

HOW

Work in pairs to imagine how your chosen persona would interact with your product.

- Imagine a product that has the characteristics you identified during the *Design Charrettes*. Leverage what you understand about your specific users and their goals, based on your *Site Visit* research and the *Experience Map*.
- Keep your scenarios at the level of behaviors, not specific UI components. For instance, say "Bob chooses the best option for him" rather than "Bob clicks on a drop-down list box to select an option." This way, you remain open to different design interpretations when you build the UI itself.

WHAT'S NEXT?

Understanding how users will accomplish their goals in your product sets you up for designing a user interface that allows them to succeed. You'll leverage your user scenarios as a recipe for building a testable *UI prototype*.

Paper Prototyping

Testing a product solution before you invest in code

WHAT

Pull all of your design ideas together into one interface to see how well they work together.



“The length of a healthy story is determined by the least number of steps absolutely essential to secure the hero’s objective.”
David Mamet “On Directing Film”

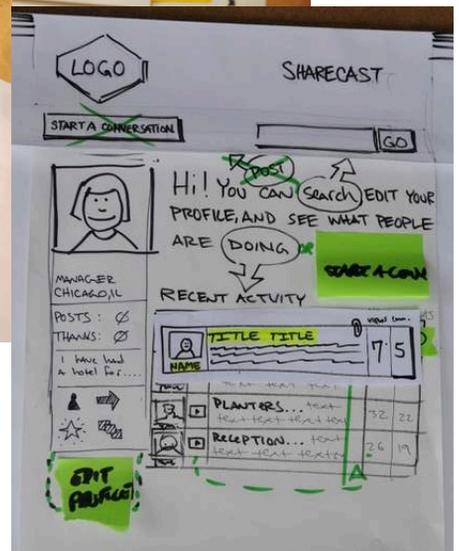


WHY

Computer-drawn designs can look too good, to the point where they fool you and your customers.

Creating something simple that allows you to walk through users’ tasks is a great way to iteratively build and fix your UI design.

Focusing attention on just the elements you want to validate means people don’t get caught up critiquing fonts or color choices.



HOW

- Take the best UI ideas from the *Design Charrette* and the behavior descriptions in your *User Scenarios*
- Draw just the interface elements needed to enable each scenario – nothing else. This way you avoid feature creep and create the minimum viable product
- Have one person read the scenario out loud while another works through the UI. Because the scenario describes behaviors and outcomes, it should be easy to see if the UI meets the criteria

TIP

Create each UI element on a separate piece of paper so that you can rearrange them or remove them without re-drawing everything.

WHAT'S NEXT?

You’ve walked through the prototype using your scenarios to do a sanity check. Now it’s time for the acid test: can real users perform their tasks with your prototype UI design? It’s time for a *Prototype Usability Test!*

Prototype Usability Testing

User feedback on a pre-alpha product

WHAT

How else can you get feedback on your ideas before they've even been coded? Users are really good at playing "let's pretend" with paper mock-ups of an interface. They can show you whether you're on the right track and what concepts need more work.



WHY

Validation. Although we started from user data, we've made several assumptions along the design path. Now it's time to find out whether those assumptions were correct. User testing – even at this early stage – shows us how our product ideas stand up to real user tasks. After testing we'll have the data we need to iterate on our designs to refine them.

HOW

The usability team will work with you to conduct paper prototype usability studies.

Tasks

- Your *User Scenarios* will form the basis for the usability study tasks
- Make sure that the wording of tasks doesn't give away the answer

Active Observers

- Team members watch (remember the duct tape)
- Write down quotes, observations. Save "solutions" until later

Output

- A list of issues with the prototype. Fix them (and re-test if necessary) before coding

WHAT'S NEXT?

After you've used the feedback from user testing to refine your prototype, it's time to plan out the incremental development cycle on a *Story Map*.

User Story Map

Break the solution down into incremental pieces

WHAT

We need to break our new design solution down into incremental parts – *Minimum Usable Releases* – that we can build and measure. A *User Story Map* uses agile user stories organized left to right by sequential user activities, then decomposed vertically from large stories about big capabilities at the top to small stories about particular aspects of the product at the bottom. This will look somewhat like the *Experience Map* but now it tells the ideal story rather than the messy current situation.

WHY

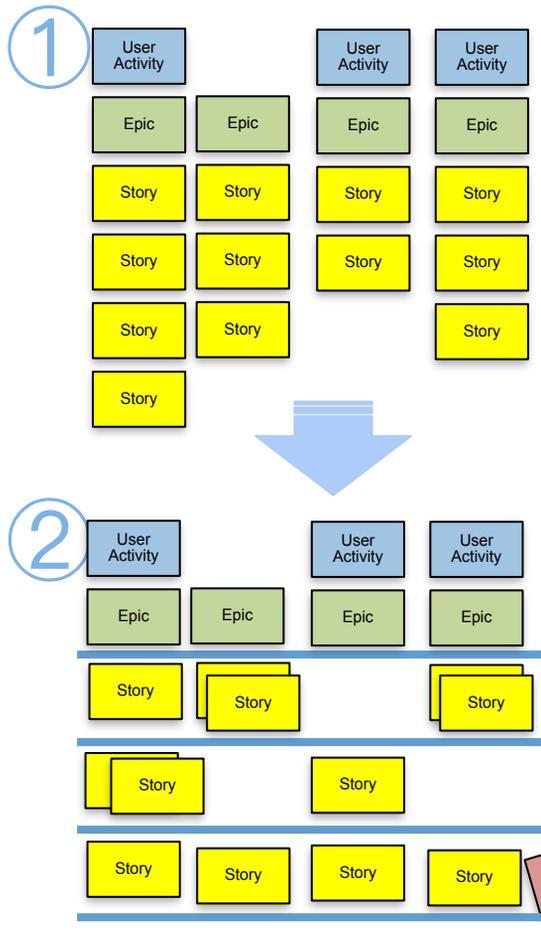
We want to build incrementally to our first minimum usable release, then iterate based on what we learn from user testing of that release. Not all of the ideas we've incorporated in our design need to be in that first release. Our sprinting strategy has to be prioritized based on delivering solutions to the *Problem Statements* we listed and ensuring a good user experience.

HOW

- Pin each page of your paper prototype across the wall. Leave plenty of space underneath for sticky notes.
- Concept: Start with top-level blue stickies. These should describe what the user needs to do, arranged step-by-step.
- Process: Next, add green stickies beneath each blue one, describing one part of the process used to achieve that goal.
- Execution: Now add the capabilities needed to create that process as yellow stickies under each green process area.

Next, so we can use it for planning purposes, we need to split this map into incremental releases.

- Add horizontal lines below your user story map to create delivery slices.
- Move the stories into each slice so that the top slice contains the minimum core functionality needed to let your chosen users achieve their goal and be happy with the product.
- The next slice down is an incremental release that adds more functionality in order to let users achieve more with the product.
- Label each slice with the problem statements it resolves and the business benefits it provides. Also say what the desired outcome is. This will be the thing you measure for the release.



Sneaky side-benefit

Creating this story map as a team means discussing the relative importance of each story towards meeting business goals. This process leads to greater understanding of what you'll be building and why.

WHAT'S NEXT?

We need to create a *Sprinting Strategy* for our first incremental release – the pieces we'll start building first.

Hat-tip to Jeff Patton (www.agileproductdesign.com), an avid promoter of user story maps.