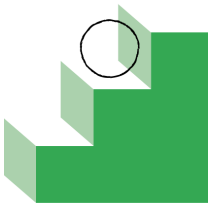
User Needs + Defining Success

Chapter worksheet



Instructions

Block out time to get as many cross-functional leads as possible together in a room to work through these exercises & checklists.

Exercises

1. Evidence of user need [multiple sessions]

Gather existing research and make a case for using AI to solve your user need.

2. Augmentation versus automation [multiple sessions]

Conduct user research to understand attitudes around automation versus augmentation.

3. Design your reward function [~1 hour]

Weigh the trade offs between precision and recall for the user experience.

4. Define success criteria [~1 hour]

Agree on how to measure if your feature is working or not, and consider the second order effects.

1. Evidence of user need

Before diving into whether or not to use AI, your team should gather user research detailing the problem you're trying to solve. The person in charge of user research should aggregate existing evidence for the team to reference in the subsequent exercises.

User research summary

List out the existing evidence you have supporting your user need. Add more rows as needed.

Date	Source	Summary of findings

Make a case for and against your AI feature

Meet as a team, look at the existing user research and evidence you have, and detail the user need you're trying to solve.

Next, write down a clear, focused statement of the user need and read through each of the statements below to identify if your user need is a potential good fit for an AI solution.

At the end of this exercise your team should be aligned on whether AI is a solution worth pursuing and why.

How might we solve _____**{ our user need }**_____?

Can AI solve this problem in a unique way?

AI probably better	AI probably **not** better
☐ The core experience requires recommending different content to different users. ☐ The core experience requires prediction of future events. ☐ Personalization will improve the user experience. ☐ User experience requires natural language interactions. ☐ Need to recognize a general class of things that is too large to articulate every case. ☐ Need to detect low occurrence events that are constantly evolving. ☐ An agent or bot experience for a particular domain. ☐ The user experience doesn't rely on predictability.	☐ The most valuable part of the core experience is its predictability regardless of context or additional user input. ☐ The cost of errors is very high and outweighs the benefits of a small increase in success rate. ☐ Users, customers, or developers need to understand exactly everything that happens in the code. ☐ Speed of development and getting to market first is more important than anything else, including the value using AI would provide. ☐ People explicitly tell you they don't want a task automated or augmented.

We think AI **{ can / can not }** help solve _____**{ user need }**_____, because

_____

_____

_____

2. Augmentation versus automation

Conduct research to understand user attitudes

If your team has a hypothesis for why AI is a good fit for your user's need, conduct user research to further validate if AI is a good solution through the lens of automation or augmentation.

If your team is light on field research for the problem space you're working in, contextual inquiries can be a great method to understand opportunities for automation or augmentation.

Below are some example questions you can ask to learn about how your users think about automation and augmentation.

Research protocol questions

- If you were helping to train a new coworker for a similar role, what would be the most important tasks you would teach them first?

- Tell me more about that action you just took, is that an action you repeat:
    - Hourly
    - Daily
    - Weekly
    - Monthly
    - Quarterly
    - Annually

- If you had a human assistant to work with on this task, what, if any, duties would you give them to carry out?

If going to meet your users in context isn't feasible, you can also look into prototyping a selection of automation and augmentation solutions to understand initial user reactions.

The Triptech method is an early concept evaluation method that can be used to outline user requirements based on likes, dislikes, expectations, and concerns.

Research protocol questions

- Describe your first impression of this feature.
- How often do you encounter the following problem: [insert problem/need statement here]?
    - Daily
    - Often (a few times a week)
    - Sometimes (a few times a month)
    - Rarely (a few times a year)
    - Never
- How important is it to address this need or problem?
    - Not at all important
    - Somewhat important
    - Moderately important
    - Very important
    - Extremely important

3. Design your reward function

Once your team has had a chance to digest your recent research on user attitudes towards automation and augmentation, meet as a team to design your AI's **reward function**. You'll revisit this exercise as you continue to iterate on your feature and uncover new insights about how your AI performs.

Use the template below to list out instances of each reward function dimension.

Reward function template

		Prediction	
		Positive	**Negative**
Reference	**Positive**	True Positive {Example 1} {Example 2} {Example 3}	**False Negative** {Example 1} {Example 2} {Example 3}
	Negative	**False Positive** {Example 1} {Example 2} {Example 3}	True Negative {Example 1} {Example 2} {Example 3}

Take a look at the false positives and false negatives your team has identified.

- If your feature offers the most user benefit for **fewer false positives**, consider optimizing for **precision**.

- If your feature offers the most user benefit for **fewer false negatives**, consider optimizing for **recall**.

Our AI model will be optimized for_____**{ precision / recall }**_____ because_____**{ user benefit }**_____ _____.

We understand that the tradeoff for choosing this method means our model will _____**{ user impact }**_____ _____ _____.

4. Define success criteria

Now that you've done the work to understand whether AI is a good fit for your user need and identified the tradeoffs of your AI's reward function, it's time to meet as a team to define success criteria for your feature. Your team may come up with multiple metrics for success by the end of this exercise.

By the end of this exercise, everyone on the team should feel aligned on what success looks like for your feature, and how to alert the team if there is evidence that your feature is failing to meet the success criteria.

Success metrics framework

Start with this template and try a few different versions:

> If **__{ specific success metric }__**
>
> for **__ { your team's specific AI driven feature }__**
>
> **{ drops below/goes above }__ { meaningful threshold }__**
>
> we will **__{ take a specific action }__.**

Version 1

Version 2

Version 3

Statement iteration

Take each version through this checklist:

- [ ] Is this metric meaningful for all of our users?
    - [ ] How might this metric negatively impact some of our users?
- [ ] Is this what success means for our feature on day 1?
    - [ ] What about day 1,000?

Final version

Schedule regular reviews

Once you've agreed upon your success metric(s), put time on the calendar to hold your team accountable to regularly evaluate whether your feature is progressing towards and meeting your defined criteria.

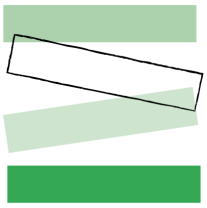
Success metric review

Date:

Attendees:

Data Collection + Evaluation

Chapter worksheet



Instructions

Use the exercises below as needed throughout your product's development.

Exercises

1. Get to know your data [~1 hour]

Decide what data you need, whether or not it already exists, and understand the sources.

2. Speak with a domain expert [~1 hour]

Use these questions as a starting point to speak with an expert in the domain.

3. Data collection considerations matrix [~1 hour]

Examine the goals of your collection effort and how you will know when you have the right data.

4. Data Labelers + Task Design [~3 hours]

Work with your labelers to ensure they have the right tools for this critical work.

5. Write data disaster/diligence headlines [~1 hour]

Avoid data disasters before they happen with this brainstorming activity.

1. Map user needs to data requirements

The first task your team has to complete is to identify the type and scope of data needed to train an ML model that can meet your users' needs.

Use the template below for each unique user need your ML model will impact.

Example: building a recipe recommendation service that suggests new dishes to cook.

User needs & data needs	
Users	*Home chefs*
User action (core value prop)	*Cook a new dish using the recipe based on recommendation*
ML system output	*Recommendations for new recipes*
ML system learning	*Patterns of behavior around choosing recipe recommendations*
Training dataset needed	*Set of recipes user has previously found, used, and liked*
Key **features** needed in dataset	*Ingredient cost* *Cuisine type* *Allergens* *Dietary restrictions*
Key **labels** needed in dataset	*Home cook's accept / reject of recommended recipe* *Home cook's feedback as to why suggestion rejected (user-generated label)* *Recipe ratings from other users*

Data **formatting**	*dish_name (all lower case)*
Real world data considerations	*Does your recipe dataset…* *...account for speciality holiday dishes?* *...reflect dietary and allergen concerns?* *...account for different cooking equipment?*
Data source key user questions	*“How does the app know what I like?”* *"Where do these recipes come from?"*

Synthesize your core data needs with the template below.

Our product/service uses:

- _____**{ data source }**_____
- _____**{ data source }**_____
- _____**{ data source }**_____

to provide _____**{ user type }**_____ with____**{ core value prop }**_____.

Critical labels for our data include:

- _____**{ data label }**_____
- _____**{ data label }**_____
- _____**{ data label }**_____

We’re aware of how the real world (e.g. time of year, changing trends) can impact the data used in our model.

To reflect the dynamism of the real world we made sure our data includes:

- **_____{ real world data consideration }_____**
- **_____{ real world data consideration }_____**
- **_____{ real world data consideration }_____**

2. Speak with a domain expert.

Once your team has the user-data needs template complete, identify **domain experts** who can give you feedback on your initial data hypotheses.

A domain expert is someone with a specialization in your ML model's subject area (not necessarily a ML expert) and can give you insights into the real-world implications of your data.

Questions for domain experts

- What data are important in your domain for <target use case>?
    - What makes data usable vs. unusable in your domain?
- How are data collected in your domain <target use case>? (e.g. in person, on paper, over the phone, online, a mix?)
    - Do you have recommendations for data collection and/or labeling organizations?
- What problems occur with the data (e.g. reporting, representation, capturing, updating)?
- Are there any environmental and/or temporal circumstances that impact data collection (e.g. type of sensor used, time of day/year)?

- How easy or difficult is it to reuse data in your domain?
- What are the top 3-5 things people should be aware of when it comes to working with data in this domain?

3. Data Collection Weighted Matrix[1]

Once your team knows what data will be required to train your model based on your answers to in the user + data needs template from exercise 1 and you've consulted with domain experts, you'll need to determine if you can get those data from:

- An existing dataset
- A new dataset

Use the weighted matrix below with your team to gain consensus on your data collection plan *(example matrix filled in below for a team with 6 people voting)*:

1. Have each team member vote for which dataset type is the best option for each row
    - The dataset criteria are suggested, you can change the criteria based on your team needs, but we strongly recommend always including 'fit for use case' and 'maintainability'
2. Multiply the number of votes for each option by the associated weight
3. Total the weighted number of votes per dataset option to give direction to your data collection plan.

[1] This exercise is adopted from the weighted matrix exercise featured in Martin, Bella, and Bruce M. Hanington. **Universal Methods of Design**: 100 **Ways** to Research Complex Problems, Develop Innovative Ideas, and **Design** Effective Solutions. Beverly, MA: Rockport Publishers, 2012.

Dataset options → Data criteria ↓	Weight	**Existing dataset (no transformations)**	**Existing dataset (with transformations)**	**New dataset + Existing dataset**	**New dataset**
Fit for use case *Is this data appropriate for your users and use case? Consider PII and Protected Characteristics: in some regions it's illegal to use them to make certain predictions.* *Are there any risks of the dataset excluding certain user groups?* *Have you used the Facets tool or some other tool/technique to evaluate the dataset for bias?*	3	(1x3)	(1x3)	(1x3)	(3x3)
Legality / Compliance *What data standards are in place for compliance, licensing, documentation?* *See if you the dataset has a Data Card (or whether your team would need to create one)*	3	(2x3)	(1x3)	(1x3)	(2x3)
Maintainability *Does your team have a plan for maintaining the data post launch?* *How will data stay up to date over time?*	2	(1x2)	(1x2)	(1x2)	(2x2)
Data collection effort *How will the data be collected?* *How will your team ensure ethical data collection practices?*	2	(1x2)	(3x2)	(1x2)	(1x2)
Cost *What are the costs of choosing the most expedient data vs. the best data?*	1	(1x1)	(3x1)	(1x1)	(1x1)
Total		14	17	11	22

4. Data Labelers + Task Design

If your feature uses supervised learning and you are using a new dataset, you need to understand the people who will be teaching or evaluating your model, also known as "raters", (or “oracles”, “labelers”, or “analysts”).

Labelers can be:

- Employees at a labeling company
- Volunteers
- Your own team members
- Or a combination of all of the above!

Use the questions below to get to understand potential mental model mismatches between your labelers vs. your users.

4.1 Who are your labelers?

- What are the particular perspectives or biases that labelers may be bringing to this task that could impact the quality of the labels?

Consider what contextual knowledge would be important for a person labeling data for:

- An AI music recommendation system
- An AI predicting likelihood of depression
- An AI for recommending job candidates

- How will you compensate labelers fairly for their work?

Consult with your domain expert for advice.

4.2 Task Instructions checklist

Help your labelers master a task by creating easy to use instructions.

DRAFT AND PILOT

- ☐ Draft instructions and budget time to get feedback from labelers on any aspects of the instructions that are unclear. *If you have already made instructions, don't worry! You can ask for feedback at any point.*

BITE-SIZE

- ☐ Break down instructions into manageable chunks by using bullets for steps, data items, or rules.
  - ☐ In house labeling teams and 3rd party companies may have the benefit of doing in person/remote trainings, but that doesn't mean instructions shouldn't be broken down into easily referenceable chunks

EXAMPLES/IMAGES

- ☐ Add at least 3 positive, negative, and ambiguous examples to illustrate expectations.
- ☐ If you are advertising a task on an open crowd platform, use images to capture worker interest in your task.

EXPLANATIONS

- ☐ Explain the overall goal of the effort to provide context and get labeler investment.
- ☐ Explain criteria for acceptance, and clearly state what errors would trigger a rejection of the task. Allow for a feedback mechanism for labelers to flag ambiguous cases.

ACCESSIBILITY

- ☐ Highlight if the task is fully accessible or requires specific abilities to complete.

4.3 Task design and usability

In case you missed it - read the article First: Raters to understand how different types of labeling impact the design of labeling tools.

- ☐ **Do the task yourself!**
  - ☐ Catch and correct any usability issues prior to testing with labelers.
- ☐ **Observe people completing your task**
  - ☐ Can labelers complete key tasks quickly and without errors?
  - ☐ Note: make it clear you are evaluating the task and not the individual's performance.
- ☐ **Plan for unsures**
  - ☐ Is your labeling UI forcing labelers to label prematurely or in error?
  - ☐ How are you thinking about inter-rater reliability?
    - ☐ Will labelers be able to periodically indicate their level of confidence for a given task submission? (This technique can help reduce the need for multiple ratings)
    - ☐ Can the data be labeled in more than one way?
- ☐ **Welcome feedback on your task/tool**
  - ☐ What incentives are there for labelers who speak up about discrepancies or interesting insights beyond the scope of the task?
- ☐ **Provide feedback to labelers in a timely manner**
  - ☐ How will labelers know they are doing a good job and that their feedback is valued?

Additionally, you can use / modify the following questionnaire to evaluate the usability of your task:

Please evaluate the usability of the task you are working on.

	Agree	Disagree	Not applicable	Comments
1. The goal of the task is clear				
2. The task instructions are comprehensive				
3. The task instructions are easy to reference				
4. The task was easy to learn				
5. The steps to complete the task are in a logical sequence				
6. The task shortcuts are useful				
7. The task shortcuts are logical				
8. It is easy to ask questions and get answers about the task				
9. The time to complete the task is appropriate				

5. Data disaster/diligence headlines

Write data disaster (and diligence) headlines to spot problematic data issues before they happen. Use these headlines to identify any data concerns to follow up with your engineering partners.

Data privacy	Customers of **{product}** upset to learn it uses **{sensitive data}**.
	{Product} champions essential data by limiting use of **{sensitive data}**.
Guiding questions ● How do you get access to the data? Do you have permission? ● What anonymization and/or aggregation techniques does your product use?	

Data exclusion	Uproar over **{product}'s** lack of **{data type}** that excludes **{user group}.**
	Praise for **{product}'s** inclusion of **{data type}** that benefits **{user group}.**
Guiding questions ● What is the downstream, real-world effect of this model's performance? ● What data is missing that would adversely impact certain user groups?	

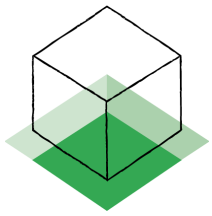
Data ethics	Calls to boycott **{product}** over unfair treatment of **{humans involved in data collection/labeling}.**
	{Product} sets the bar for **{humans doing data collection/labeling}** by **{action taken to compensate fairly}**.
Guiding questions ● Who are the humans involved collecting and/or labeling your data? ● How are you compensating them for this critical work?	

Data transferability	**{Product}** cancelled over faulty **{data}** used from **{inappropriate source}**.
	{Product} innovates in leveraging **{data}** from **{source}** by **{action taken to transform data}.**
Guiding questions ● What risks are present for using data not originally intended for your use case?	

Data fragility	**{Product}** down as team struggles to fix **{key data input sources}.**
	{Product} outperforms competitors thanks to including **{data}** that accounts for **{real world consideration}.**
Guiding questions ● Does your data reflect the real world? e.g. for image based systems does it include off center/blurry images?	

Mental Models

Chapter worksheet



Instructions

Block out time to get as many cross-functional leads as possible together in a room to work through these exercises & checklists.

Exercises

1. Existing vs. new mental models [~2 hours]

Determine existing user mental models to understand how your product will break or reinforce them.

2. Creating onboarding [~1 hour]

Craft your onboarding message and test user comprehension of cause and effect.

1. Existing vs. new mental models analysis

Discuss the following questions as a group, then capture answers in the boxes below. Review your answers as a team to determine what approaches your product will need to take to help users establish good mental models.

Example product: AI that automatically prioritizes new emails and sorts inbox according to their priority

Key questions

Who are your different user groups? Add more boxes as needed.

User group A

Example: employee at large company using email for work

User group B

Example: Everyday consumer using a free email service

What **primary goal** will each user group have?

User group A

Example: Goal - Prioritize tasks and communications received through email in order to do well at their job.

User group B

Example: Goal - Not miss the few important emails received among the flood of promotional emails.

What is the step-by-step process that **novice users** from each group currently use to accomplish the task that the AI system will accomplish? How uniform or variable is this process?

Note: user research may be needed to answer this question

User group A

Example:

Process - Frequently check email, individually triage each message.
Uniformity - Highly variable.

User group B

Example:

Process - Scan inbox for important mail, ignoring the rest
Uniformity - Highly variable.

What is the step-by-step process that **expert users** from each group currently use to accomplish the task that the AI system will accomplish? How uniform or variable is this process?

Note: user research may be needed to answer this question

User group A

Example:

Process - Set up multiple custom filters, notifications, and labels and folders
Uniformity - Highly variable.

User group B

Example:

Process - Set up filters, systematically unsubscribe from lists to free up inbox.
Uniformity - Highly variable.

What **mental models** might already be in place based on the step-by-step process and any non-AI-driven tools used by each group?

User group A

Example: Use sender, subject, and knowledge of my existing work to prioritize new email

User group B

Example: Use sender, subject, and knowledge of what messages I might be expecting (e.g., online order notifications) to pick out important new email

Based on existing mental models, are there potential **places where the user's mental model could break** when encountering the realities of the AI's functionality?

User group A

Example: AI system can't account for user knowledge of their wider context (e.g., they just changed roles at work; they are expecting an email from someone not in their contacts list)

User group B

Example: The AI system can't account for infrequent and variable but important emails from friends or loved ones

Given all the above, what **cause and effect relationships** does the user need to understand — even in simplified terms or by analogy — to successfully use the AI product?

User group A

Example: Priority of email varies by:

- *Number of recipients (just user or large group)*
- *Frequency of sending emails to contact*
- *Speed at which user opens and replies to email*

User group B

Example: Priority of email varies by:

- *Contact's membership in a specific group*
- *Active orders or subscriptions*
- *Length of communication*

Given the mental model we want users to have, how might anthropomorphizing the product alter the mental model?

User group A

Example: Making the system seem "human" might imply that the AI actually does have the same knowledge and context as the user, which conflicts with the key cause and effect relationships the user needs to understand.

User group B

The biggest risks to users developing good mental models for our product are:

User group A

User group B

List the key points in product where messaging is critical for creating or updating a good mental model. For example: “onboarding”, “inboarding”, or “reboarding".

User group A

User group B

What if anything might need to change about how the AI works in order to accommodate mental models?

All users

Example: AI works as a binary yes / no categorizer for censoring content in an online forum, but users expect gradations of control.

2. Creating onboarding

Start crafting your onboarding message using this template, and try a few different versions:

1. Onboarding template

> This is __**{ your product or feature }**___, and it'll help you by __**{ core benefits }**__.
>
> It's NOT able to __**{ primary limitations of AI }**__.
>
> Over time, it'll change to become more relevant to you.
>
> You can help it get better by __**{ actions users can take to help the system learn }**____.

Version 1

Version 2

Version 3

2. Messaging checklist

Take each version of your messaging through this checklist:

- ☐ Does the description focus on the benefits to the user and not the technology?
- ☐ Are we introducing the product at the right level, or are we overloading the description with things that should be saved for “inboarding”?
- ☐ In the product, do we make it easy to experiment with the process we describe in the “You can help it get better by…” phrase?
- ☐ Is the description specific and explicit about how the user will interact with and improve the AI over time?
- ☐ Are we specific and explicit about how the system will change over time and how that will benefit the user?

3. Demonstrating cause and effect

Outline what actions the user will do next in order to reinforce the information described in the message you wrote above.

- Will they complete set-up tasks?
- See examples of what the AI can do?
- Simply start trying it?

4. Test user mental models

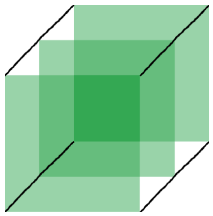
Pick your best draft onboarding messaging + next action concepts, or pick several to test, then conduct user research.

Research protocol questions

- First, show users your initial onboarding concepts, then ask them questions like:
    - Explain in your own words what [product] is.
    - Explain in your own words how [product] works.
    - Based on what you saw, describe what using [product] will be like.
    - Based on what you saw, how useful do you expect [product] to be for you?
    - Any additional expectations you have about [product] based on what you read?
- Next, if you have any wireframes or demos or working prototypes of your product or feature, show it to the user after walking through your onboarding experience concepts.
- Lastly, after interacting with both the design concepts and the AI prototype, have users describe how the AI experience compared to their expectations.

Explainability + Trust

Chapter worksheet



Instructions

Block out time to get as many cross-functional leads as possible together in a room to work through these exercises & checklists.

Exercises

1. Trust calibration [~1 hour]

Imagine situations where users could under-trust or over-trust your feature.

2. Explanation strategy[~30 minutes]

Determine which user interactions require an explanation, and what kind.

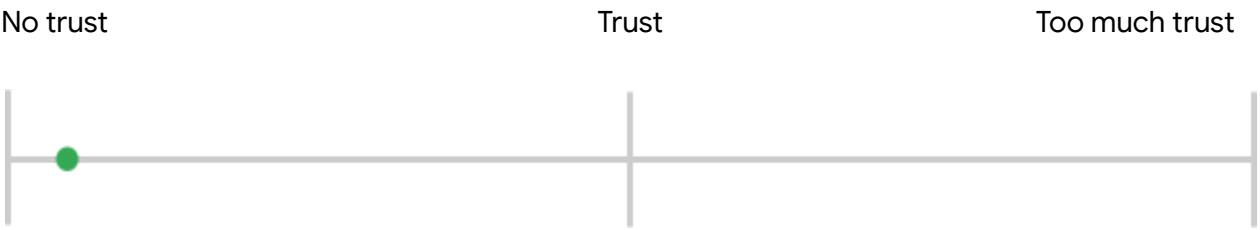
3. Test with user research [multiple sessions]

Make sure that the explanation visuals and messaging make sense and are helpful for your users.

1. Trust calibration

As a team, brainstorm what kinds of experiences and interactions would decrease, maintain, or inflate trust in your feature's AI. Identify the underlying data sources, system data and user knowledge, that could impact the calibration.

Example product: AI that classifies a skin condition.



User Group A *Example user group: Doctors of patients using the AI system* *Example scenario: Patients see doctor with a condition that was mis-classified by the AI.*	
System data impacting calibration	*Example: image of current condition submitted by user, label of skin conditions in training data*
User knowledge impacting calibration	*Example: user's prior medical history*

User Group B

Example user group: Patients using the AI system

Example scenario: Patients see doctor with a condition that was misclassified by the AI.

System data impacting calibration	
User knowledge impacting calibration	

No trust | Trust | Too much trust

User Group A *Example user group: Patients using the AI system* *Example scenario: Patient uses AI system to identify a common and temporary condition, like poison ivy, and receives a recommended treatment that works.*	
System data impacting calibration	
User knowledge impacting calibration	

User Group B	
System data impacting calibration	
User knowledge impacting calibration	

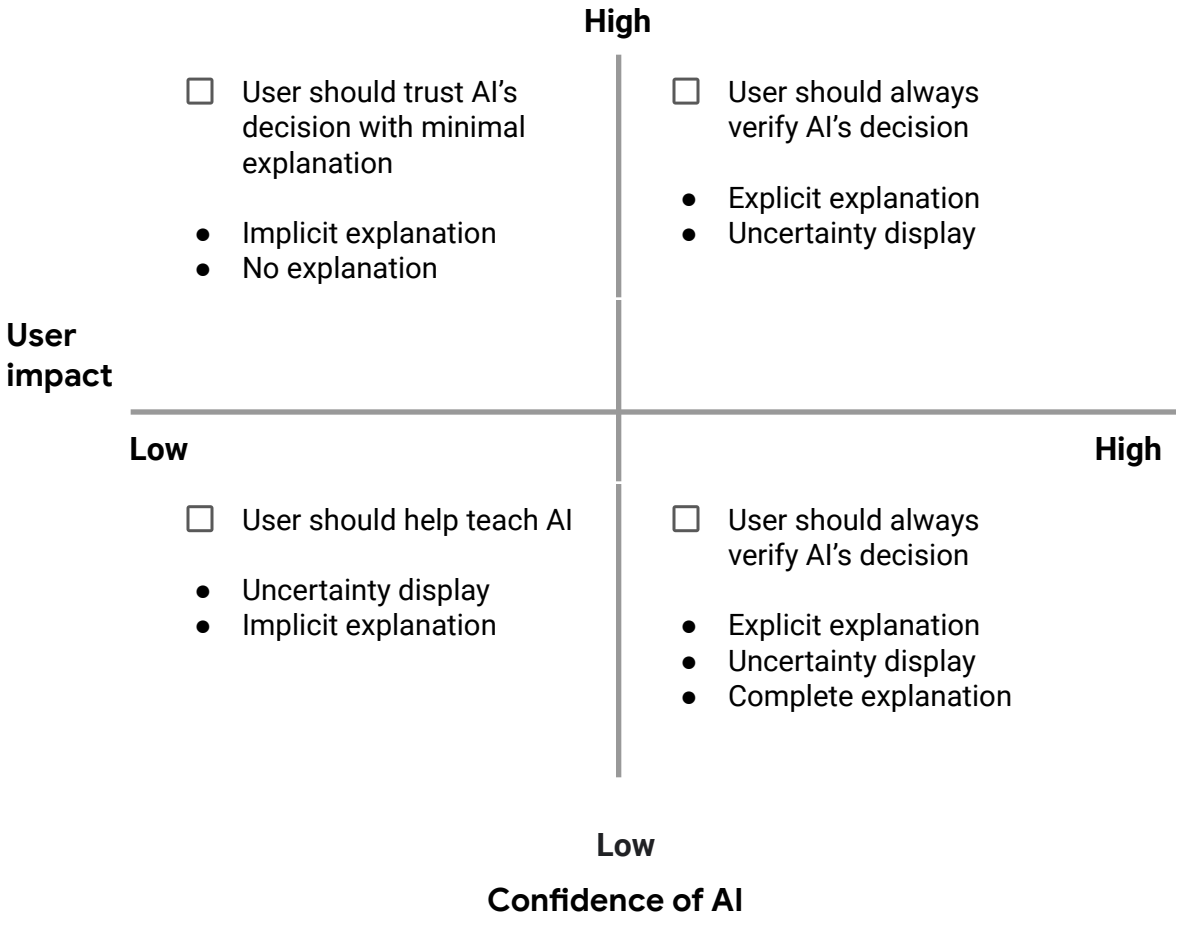
No trust | Trust | Too much trust

User Group A	
Example user group: Patients using the AI system *Example scenario: Patients with pre-cancerous cells doesn't double check the app's diagnosis*	
System data impacting calibration	
User knowledge impacting calibration	

User Group B	
System data impacting calibration	
User knowledge impacting calibration	

2. Explanation strategy

After mapping the range of interactions that could decrease, maintain, or inflate trust, decide which interactions require an explanation. Use the 2x2 template of “User impact” vs. “Confidence of AI” to help narrow down which explanations you want to test with users. Once you have consensus on the interactions that require an explanation, use the templates below to draft the explanation copy for user testing.



Explanation messaging

Product interaction	
Users	
Technical understanding	☐ expert ☐ non-expert
Explanation type	
Explanation text	

Messaging templates

Explicit explanation

Example:
“You are seeing this video recommendation because you often watch cooking videos.”
“This is most likely ____X_____, because _____Y_____.”

Uncertainty display - Confidence level

Example: “Prediction: [category] XX%”

Uncertainty display - N-Best

Example: “Most likely X, Y, or Z”

3. Test with user research

Use the explanations you have drafted in user research. Multiple research efforts will be needed to understand any trust concerns users anticipate, and arrive at the ideal explanations for your user groups.

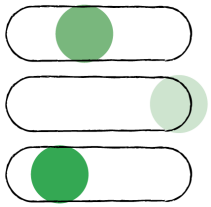
Research protocol questions

- On this scale, show me how trusting you are of this recommendation. [show scale]
- What questions do you have about how [name of product / feature] came to this recommendation?
- What, if anything, would increase your trust in this recommendation?
- How satisfied or dissatisfied are you with the explanation written here?

Once you're through with the above, ask additional questions to gauge user understanding of your specific AI system and the actions users can take.

Feedback + Control

Chapter worksheet



Instructions

Block out time to get as many cross-functional leads as possible together in a room to work through these exercises & checklists.

Exercises

1. User expectations for control [~30 minutes]

Review situations or events in your app where your users want or need higher levels of control.

2. Feedback audit & prioritization [~2 hours]

Audit feedback mechanisms, define the priority and specificity of feedback signals, and determine how you'll integrate user feedback in your product.

3. Validate feedback mechanisms [multiple sessions]

Conduct user research to understand whether or not your feedback mechanisms are working as intended and set aside time to review feedback from users.

1. User expectations for control

Take time to reflect on what aspects of the system the user expects to have control over and why. Use the checklist and template below to help identify the right level of control and customization. Make sure you validate your hypotheses with user research.

Note: You might want to complete this activity after some initial user testing depending on your familiarity with your user groups. You may also want to revisit any exercises you did from User needs + defining success chapter.

Higher need for user control

- ☐ The AI cannot yet accommodate a wide range of user abilities and/or preferences.
- ☐ The AI deals with highly-sensitive domains: money, business, relationships, or health.
- ☐ The AI might take a long time to get to the target level of accuracy and usefulness. User controls could be a good stop-gap measure in the meantime.
- ☐ AI is used in a high-stakes situations for which AI presents a new interaction paradigm. The user might want the option to be in full control at first as they build trust in the system.
- ☐ System limitations or other potential errors will require user control to correct the system. More detail in the Errors + graceful failure chapter.
- ☐ There are likely scenarios in which changes in the user's circumstances require them to "reset" or otherwise "take over" for the model. For example: user is now vegetarian, so previous restaurant recommendations are no longer relevant.

Mapping a user's need for control

Use the template below to map out how much control your user needs across different interactions with your app. Refer to your user groups from the Mental Models worksheet.

Example product: AI that recommends recipes based on the contents of a picture the user submits.

Users	Event / Task / Feature	Level of user control needed
First time users	Users have unique dietary needs	☑ ~~high~~
		☐ low ☐ medium ☐ high ☐ unsure
		☐ low ☐ medium ☐ high ☐ unsure
		☐ low ☐ medium ☐ high ☐ unsure
		☐ low ☐ medium ☐ high ☐ unsure
		☐ low ☐ medium ☐ high ☐ unsure

2. Feedback

The person responsible for your product's user research should block out time to take stock of existing or potential feedback mechanisms.

Consult with engineers, designers, product counsel, and other stakeholders as needed to generate a truly comprehensive audit. Use the template below to map current and potential feedback mechanisms in your product.

Feedback audit

List out events and corresponding feedback opportunities that could provide data to improve the AI in your product. Cast a wide net and list as many as possible: App Store reviews, Twitter, email, call centers, push notifications, etc. Add more rows as needed.

Example product: AI that recommends recipes based on the contents of a picture the user submits.

Event	Feedback	Type of Feedback
User completes a recommended recipe	User prompted to rate relevance of completed recipe	☐ implicit ☑ ~~explicit~~
User rejects recommendations including a certain ingredient	Specific ingredient is outside of users preferences	☑ ~~implicit~~ ☐ explicit
		☐ implicit ☐ explicit
		☐ implicit ☐ explicit

		☐ implicit ☐ explicit
		☐ implicit ☐ explicit
		☐ implicit ☐ explicit

Feedback prioritization

After auditing all the potential sources of feedback, prioritize what feedback will need to be collected to impact the AI.

When prioritizing feedback opportunities to improve the user experience with the AI ask yourselves:

- ☐ Do all of your user groups benefit from this feedback?
- ☐ How might the level of control that the user has over the AI that the user has (or wants to have) influence the user's desire to provide feedback?
- ☐ How will the AI change from this feedback?

Feedback mission statement

After discussing the above questions, use the template below to write mission statements for each source of feedback that your team agrees can be used to alter the AI. Revisit this exercise as features get added or deprecated that impact the user's interaction with the AI.

We want to collect _____**{ type of user feedback }**_____,

so we can improve_____ **{ user interaction with AI }**_____

By_____**{ change made to the AI }**_____.

Feedback mission statement 1

Feedback mission statement 2

Feedback mission statement 3

Feedback specification

Take the feedback mission statements your team has prioritized through the template below to get more specific about the feedback lifecycle. Feel free to add categories relevant to your product as necessary.

Subject of feedback	
Feedback Mission Statement We want to collect _____ _____, so we can improve_____ _____ by_____ _____.	☐ Improves the system as a whole ☐ Communicates with others ☐ Improves personalization Reviewed by: ☐ Engineering ☐ Legal ☐ Design ☐ PM ☐ User ☐ Other
Timing and frequency *Are we asking for feedback at the right moment for the user or the AI? What are the trade-offs?*	

Type of feedback

- ☐ implicit
- ☐ explicit
- ☐ qualitative
- ☐ quantitative

Ambiguity risks

Any risks of feedback being misinterpreted or having a dual meaning?

Feedback motivation

User motivation for feedback:

User value / benefit in giving feedback:

Timing until feedback value / user benefit / impact:

Alternative user value for feedback:

Feedback mechanism ideas

Example: a daily toast notification, an annual survey sent to linked email account

Acknowledgment

Is the acknowledgement specific and explicit about how and when the user's feedback will improve the AI?
Opt out & dismissal *Do we give users a way to opt out or dismiss the feedback mechanism?*

3. Validate feedback mechanisms

Once your team has a detailed proposal for collecting feedback, test your proposal with users.

It can be hard to concept test feedback mechanisms in a realistic way when the user is not actually using the product as part of their day-to-day life. So, you may need to be creative in how you design a user study that will create the context under which the user will encounter the feedback mechanism. You may be able to do this also through diary studies or other forms of longitudinal studies with beta versions of your app.

User research methods

"Wizard of Oz" prototypes can also be a great method for collecting meaningful data around the feedback experience. Allocate time in your research plan for participants to submit content or examples ahead of the study session to use in your prototype feedback experience. Users will be more invested in critiquing the feedback and level of control if they can relate it back to their own data.

Example: If the recipe app plans to survey users each week about what recipes they cooked, you should consider planning a study that is a week long with a prototype version of the app.

Depending on the nature of your study, you might be able to observe user responses to feedback prompts and results, then query them with questions like this:

Research protocol questions

- Why do you think you're being asked for this feedback?
- What might influence your decision as to whether or not you'd provide feedback here?
    - How, if at all, do you expect your experience to change after you provide a response?
    - What other factors, if any, do you think will be used to change your experience over time? [probing on implicit feedback]

- [Towards the end of the session] You'd like to update a preference so that you see [more of / less of X]. Show me how you would do that.

Lastly, after validating your proposed feedback mechanisms, schedule time after the launch of your feature for your team to review actual feedback from users.

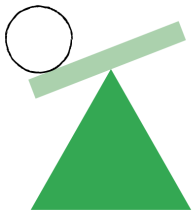
Feedback review session

Date for review of feedback:

Attendees:

Errors + Grace Failure

Chapter worksheet



Instructions

Block out time to get as many cross-functional leads as possible together in a room to work through these exercises & checklists.

Exercises

1. Error audit [~1 hour]

Collect canonical error examples to define existing and potential errors and solutions.

2. Quality assurance [~30 minutes]

Prioritize how you'll test and monitor errors and reporting so you can hear from your users early and often.

1. Error audit

As a team, brainstorm what kinds of errors users could encounter. If your team has a working prototype of your feature, try to add current examples.

Use the template below to start collecting error examples so your team has a shared understanding about the different error types and solutions your model could produce.

Error	**Users**
Add screenshots, pictures, or logs to show what the user sees when encountering the error	**User stakes** ☐ low ☐ high
Error type ☐ **System limitation** - Your system can't provide the right answer, or any answer at all, due to inherent limitations to the system. ☐ **Context** - The system is “working as intended,” but the user perceives an error because the actions of the system aren't well-explained, break the user's mental model, or were based on poor assumptions. ☐ **Background** - Situations in which the system isn't working correctly, but neither the user nor the system register an error.	

Error sources

Take each error identified above through these questions to determine the source of the error:

Input error signals

- [ ] Did the user anticipate the auto-correction of their input into an AI system?
- [ ] Was the user's habituation interrupted?
- [ ] Did the model improperly weigh a user action or other signal? If yes, likely a context error.

Relevance error signals

- [ ] Is the model lacking available data or requirements for prediction accuracy?
- [ ] Is the model receiving unstable or noisy data?
- [ ] Is the system output presented to users in a way that isn't relevant to the user's needs?

System hierarchy error

- [ ] Is your user connecting your product to another system, and it isn't clear which system is in charge?
- [ ] Are there multiple systems monitoring a single (or similar) output and an event causes simultaneous alerts? Signal crashes increase the user's mental load because they have to parse multiple signals to figure out what happened and what to do next.

Failure state

- [ ] Is your feature unusable as the result of multiple errors?

Error resolution

Once you have identified the source or sources of the error, complete the sections below for each of the errors in the template with your team's plan for improving / reducing the identified error: Create as many copies as you need to cover all your identified errors.

Error rationale	**Solution type**
Why the user thinks this is an error:	☐ Feedback ☐ User control ☐ Other:

Error resolution

User path:

Examples: User sees errors, gives feedback, completes task; User sees error, takes over control, completes task

Opportunity for model improvement:

Example: User's feedback logged for model tuning

2. Quality assurance

Getting your feature into users' hands is essential for identifying errors that your team, as expert users, may never encounter. Meet as a team to prioritize how you want to monitor errors reported by users so that your model is being tested and criticized by your users early and often.

As you have this discussion, consider all potential sources of error reporting:

- Reports sent to customer service
- Comments and reports sent through social media channels
- In-product metrics
- In-product surveys
- User research (out-of-product surveys, deep dive interviews, diary studies, etc.)

QA template

<table>
<tr><td>Goal</td><td rowspan="3">Review frequency
☐ Daily
☐ Weekly
☐ Monthly
☐ Other:</td></tr>
<tr><td>Method</td></tr>
<tr><td>Start date:
Review / End date:</td></tr>
</table>