

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
10-22-25	CASE ID #101006332910	Delay in illegal parking case. User expects faster enforcement update
10-21-25	CASE ID #101006330791	Abandoned vehicle not resolved, cross-department coordination issues.
10-20-25	CASE ID #101006330628	Graffiti removal pending, no visible progress or status update to user.
10-26-25	CASE ID #BCS-00107351	Street light outage still open, lack of ETA communication causes concern.

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 citizens' need to quickly get accurate updates on 311 service requests and help city workers prioritize tasks efficiently?

Can AI solve this problem in a unique way?

AI can solve this by understanding user queries, giving quick personalized answers, and predicting high-priority cases for faster action

AI probably better	AI probably not better
<ul style="list-style-type: none"> <input checked="" type="checkbox"/> The core experience requires recommending different content to different users. <input checked="" type="checkbox"/> The core experience requires prediction of future events. <input checked="" type="checkbox"/> Personalization will improve the user experience. <input checked="" type="checkbox"/> User experience requires natural language interactions. <input checked="" type="checkbox"/> Need to recognize a general class of things that is too large to articulate every case. <input checked="" type="checkbox"/> Need to detect low occurrence events that are constantly evolving. <input checked="" type="checkbox"/> An agent or bot experience for a particular domain. <input type="checkbox"/> The user experience doesn't rely on predictability. 	<ul style="list-style-type: none"> <input type="checkbox"/> The most valuable part of the core experience is its predictability regardless of context or additional user input. <input type="checkbox"/> The cost of errors is very high and outweighs the benefits of a small increase in success rate. <input type="checkbox"/> Users, customers, or developers need to understand exactly everything that happens in the code. <input type="checkbox"/> Speed of development and getting to market first is more important than anything else, including the value using AI would provide. <input type="checkbox"/> People explicitly tell you they don't want a task automated or augmented.



We think AI **can** help solve help solve citizens' need to quickly get accurate updates and report issues through the Boston 311 system, because AI can understand natural language, personalize responses, and analyze large volumes of service request data in real time, because

It can also predict which cases need urgent attention and provide insights through dashboards to help city workers act more efficiently.

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?
I'd teach them how to review new 311 requests, identify urgent cases, update residents, and use the dashboard to track and prioritize issues efficiently.
- Tell me more about that action you just took, is that an action you repeat:
 - Hourly
 - Daily [reviewing and prioritizing new 311 request over last 28 days data]
 - Weekly [reviewing and prioritizing new 311 request across all time data]
 - 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?
I'd ask the assistant to help monitor incoming 311 cases, flag overdue or high-priority issues, update case statuses, and prepare weekly summaries for the dashboard.



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.
It's simple, useful, and saves time by giving quick updates and clear issue tracking.
- How often do you encounter the following problem: [insert problem/need statement here]?
 - Daily [new 311 service requests and user queries]
 - 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 Quick, accurate chatbot replies Dashboard helps workers act faster and prioritize well.	False Negative Chatbot gives vague or delayed answers. Dashboard shows incorrect priorities or slow updates
	Negative	False Positive Chatbot gives vague or delayed answers. Dashboard shows incorrect priorities or slow updates	True Negative Chatbot gives wrong info confidently. Dashboard displays outdated or misleading data

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** _____
because _____ **user benefit most from accurate, trustworthy
information and correct case prioritization**

We understand that the tradeoff for choosing this method means our
model will _____ **user impact sometimes skip or delay uncertain
responses instead of guessing, which may slightly reduce coverage or
speed but maintains user trust**

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

If **chatbot response accuracy**
for **the citizen-facing query system**
drops below 90%,
we will **review model intent classification and retrain using new user logs**.

Version 2

If **dashboard data freshness**
for **the real-time case monitoring feature**
exceeds a 1 day delay,
we will **investigate Airflow and BigQuery pipelines to restore real-time sync**.

Version 3

If **average response time for high-priority service requests on the worker dashboard fails to improve by at least 15% within 3 months**, we will **review the prioritization model and improve weighting for urgency factors**

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

If **either chatbot accuracy or dashboard freshness falls below the defined thresholds for two consecutive review cycles**, we will **pause automatic responses, notify the operations team, and perform a full audit of model performance and data pipelines.**

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: November 18

Attendees

Google

- Project Lead
- Data Analyst
- Data Scientist
- Data Engineer I
- Data Engineer II
- Tableau Developer