

A workbook for designing legible AI products

The Legibility Workbook

Five worksheets to help you design AI explanations that people notice, understand, and learn from.

Want to try this on your computer or phone?

Work through the five principles on screen with the interactive version, an experiment on Prism.

[Open the interactive version →](#)

www.jordangulbronson.com/prism/legibility-workbook

Inside

01 Engage	02 Challenge habits	03 Guide attention	04 Introduce friction	05 Support learning
---------------------	-------------------------------	------------------------------	---------------------------------	-------------------------------

Credit

Framework adapted from Naiseh, M., Al-Thani, D., Jiang, N., & Ali, R. (2021). Explainable recommendation: when design meets trust calibration. World Wide Web, 24(5), 1857-1884. <https://doi.org/10.1007/s11280-021-00916-0>. Licensed CC BY 4.0.

Start here

Why legibility

A common answer to distrust in AI is more transparency: open the black box and show what’s inside. But showing more doesn’t mean people understand more. A novel of text explaining one decision is as opaque as no explanation at all. To be trusted, an AI system has to be transparent and legible: explanations people actually notice, read, and learn from.

How to use this workbook

- 1 Pick an AI feature. Choose a screen, a decision, or an output your users rely on.
- 2 Fill in the project box on this page.
- 3 Work through each principle sheet. Answer the questions, check the red flags you recognize, and sketch what you’d do instead.
- 4 Rate how well your product accomplishes each principle.

Where it comes from

The five principles come from Naiseh et al. (2021). The questions, red flags, and adjustments are mine, shaped by my work at Amazon and Tatari.

About your project

Product or feature you’re working on

What does the AI decide, generate, or recommend?

Who relies on it, and what do they do next?

What’s at stake if they misread it?

Credit and license

This workbook adapts the five principles from:

Naiseh, M., Al-Thani, D., Jiang, N., & Ali, R. (2021). Explainable recommendation: when design meets trust calibration. World Wide Web, 24(5), 1857-1884. <https://doi.org/10.1007/s11280-021-00916-0>

Naiseh et al. is published under a Creative Commons Attribution 4.0 license (creativecommons.org/licenses/by/4.0). Changes made: the principles are shortened to action names, and the questions, red flags, adjustments, and worksheets are added by Jordan Gulbronson. The original authors have not reviewed or endorsed this workbook.

The Legibility System

A framework for legible AI, adapted from Naiseh et al. (2021)

Principle	Questions to ask	What to watch out for	What to do instead
01 Engage Design for engagement	- Where might a user skip or skim past something they need? - What’s the minimum they need to notice before moving on? - What would make them curious enough to look closer?	- Long, text-heavy explanations that read as a wall - Static content that looks identical every time, so there’s nothing new to notice - Explanations placed after the decision, when attention has already moved on	- Structure for scanning first, depth second: headline plus expandable detail - Vary the surface for returning users (highlight what changed, not just what’s always true) - Place the explanation at the moment of the decision, not after it
02 Challenge habits Design for challenging habitual actions	- What’s the default or automatic thing users do out of routine? - Where does that default produce a worse outcome? - What would make them pause and reconsider without feeling accused?	- Treating “habit” as repetition: the risk isn’t the repeated task, it’s a bad default becoming invisible through repetition - A one-time warning, which habituates away just as fast as the original habit - Changing the behavior without explaining why, which just creates a new, unexamined habit	- Surface a specific consequence of the bad default in context before the consequence is realized - Explain the “why” once when the habit should break, not permanently where it can be ignored - Track whether the behavior changes over time, not just whether the consequence was viewed
03 Guide attention Design for attention guidance	- What’s the single most important thing and what does it look like? - What is the user actually trying to decide in this moment? - If they read only one line, which line should it be?	- Flat hierarchy: everything styled at the same weight, so nothing reads as more important - Hierarchy that reflects what the system has available rather than what the user needs - Emphasis that shifts arbitrarily between instances, so no stable expectation forms	- Rank content by user goal, not system capabilities, and build that goal into the output as a rule - Apply the rule consistently, keeping the structure clear even with different content - Check whether a user can state the “most important thing” back to you
04 Introduce friction Design for friction	- Where could a wrong or irreversible action happen without the user realizing? - Does the added friction match the consequence of getting this wrong? - Are users rushing past a step that matters?	- No natural stopping point before high-stakes actions, so flows complete on autopilot - Friction applied evenly everywhere, which trains users to click through without reading - Confirmation that requires no real engagement: a single “OK” teaches nothing	- Add a deliberate interruption at the specific point of highest stakes, not spread evenly through the flow - Require an active response (re-enter a value, check a box, view a preview), not a passive dismiss - Track skip/ignore rate on the interruption; near 100% means it’s annoying, not calibrated
05 Support learning Support training and learning	- What does the user not yet understand that they’ll need again later? - Is this building independent knowledge or is it something they will quickly forget? - What would let them predict the outcome themselves next time?	- Explanations that assume prior knowledge the user doesn’t have - Technically accurate explanations that don’t build the user’s mental model - No consistent place to find the explanation, so users don’t know where to look to learn	- Explain the reasoning behind the outcome, not just the outcome itself - Put explanations in the same place every time, so seeking them becomes habitual - Measure whether reliance on the explanation decreases over time: that’s evidence of real learning, not just exposure

Engage

Design for engagement

We’re busy and often distracted. Explanations should draw attention in, be easy to scan, and be genuinely helpful so they aren’t skipped or skimmed.

Product / feature	Team	Date
01 • Frame it Where does this principle show up in your product?	02 • Ask Where might a user skip or skim past something they need? What’s the minimum they need to notice before moving on? What would make them curious enough to look closer?	03 • Watch out Check any you recognize in your product. ☐ Long, text-heavy explanations that read as a wall ☐ Static content that looks identical every time, so there’s nothing new to notice ☐ Explanations placed after the decision, when attention has already moved on Where do you see it?
04 • Do instead Check any you’d try. ☐ Structure for scanning first, depth second: headline plus expandable detail ☐ Vary the surface for returning users (highlight what changed, not just what’s always true) ☐ Place the explanation at the moment of the decision, not after it Which will you try first, and how?	05 • Measure it How will you know it’s working? What will you watch? TodayTarget	06 • Rate it and plan the change How well does your product do this today? 1 2 3 4 5 1 = not at all5 = consistently One change to make OwnerBy when

Challenge habits

Design for challenging habitual actions

We slip into autopilot, auto-approving or ignoring useful information. An AI explanation should occasionally interrupt that muscle memory when it matters.

Product / feature	Team	Date
01 • Frame it Where does this principle show up in your product?	02 • Ask What’s the default or automatic thing users do out of routine? Where does that default produce a worse outcome? What would make them pause and reconsider without feeling accused?	03 • Watch out Check any you recognize in your product. ☐ Treating “habit” as repetition: the risk isn’t the repeated task, it’s a bad default becoming invisible through repetition ☐ A one-time warning, which habituates away just as fast as the original habit ☐ Changing the behavior without explaining why, which just creates a new, unexamined habit Where do you see it?
04 • Do instead Check any you’d try. ☐ Surface a specific consequence of the bad default in context before the consequence is realized ☐ Explain the “why” once when the habit should break, not permanently where it can be ignored ☐ Track whether the behavior changes over time, not just whether the consequence was viewed Which will you try first, and how?	05 • Measure it How will you know it’s working? What will you watch? TodayTarget	06 • Rate it and plan the change How well does your product do this today? 1 2 3 4 5 1 = not at all5 = consistently One change to make OwnerBy when

Guide attention

Design for attention guidance

Mundane and complex information are both hard to process. Use visual cues and hierarchy to highlight the information that matters most.

Product / feature	Team	Date
01 • Frame it Where does this principle show up in your product?	02 • Ask What’s the single most important thing and what does it look like? What is the user actually trying to decide in this moment? If they read only one line, which line should it be?	03 • Watch out Check any you recognize in your product. ☐ Flat hierarchy: everything styled at the same weight, so nothing reads as more important ☐ Hierarchy that reflects what the system has available rather than what the user needs ☐ Emphasis that shifts arbitrarily between instances, so no stable expectation forms Where do you see it?
04 • Do instead Check any you’d try. ☐ Rank content by user goal, not system capabilities, and build that goal into the output as a rule ☐ Apply the rule consistently, keeping the structure clear even with different content ☐ Check whether a user can state the “most important thing” back to you Which will you try first, and how?	05 • Measure it How will you know it’s working? What will you watch? TodayTarget	06 • Rate it and plan the change How well does your product do this today? 1 2 3 4 5 1 = not at all5 = consistently One change to make OwnerBy when

Introduce friction

Design for friction

With lots of information, it's hard to tell what's high stakes. Slow people down when necessary so they're paying attention where they should be.

Product / feature

Team

Date

01 · Frame it

Where does this principle show up in your product?

[illegible]

02 · Ask

Where could a wrong or irreversible action happen without the user realizing?

Does the added friction match the consequence of getting this wrong?

Are users rushing past a step that matters?

03 · Watch out

Check any you recognize in your product.

- ☐ No natural stopping point before high-stakes actions, so flows complete on autopilot
- ☐ Friction applied evenly everywhere, which trains users to click through without reading
- ☐ Confirmation that requires no real engagement: a single “OK” teaches nothing

Where do you see it?

04 · Do instead

Check any you'd try.

- ☐ Add a deliberate interruption at the specific point of highest stakes, not spread evenly through the flow
- ☐ Require an active response (re-enter a value, check a box, view a preview), not a passive dismiss
- ☐ Track skip/ignore rate on the interruption; near 100% means it's annoying, not calibrated

Which will you try first, and how?

05 · Measure it

How will you know it's working? What will you watch?

Today

Target

06 • Rate it and plan the change

How well does your product do this today?

Diagram showing five circles labeled 1, 2, 3, 4, and 5, representing the nodes of the graph.

1 = not at all 5 = consistently

One change to make

Owner

By when

People aren't static, and they're often curious. Explanations should help people learn and expand their understanding, not just restate what they already know.

Date

02 • Ask

What does the user not yet understand that they'll need again later?

What would let them predict the outcome themselves next time?

Check any you recognize in your product.

- Where do you see it?

05 · Measure it

How will you know it's working? What will you watch?

- Which will you try first, and how?

Target

How well does your product do this today?

1 = not at all 5 = consistently

One change to make

By when