

1) Product background (existing context)

Pear's mission + ecosystem reality

Pear is positioning itself around seamless integration, intuitive design, low cognitive load, and privacy, across a connected ecosystem (PearOS + PearCloud) serving leisure users, working professionals, and creatives/makers.

Why this matters strategically

Pear's stated challenges include market saturation and retention/differentiation via experiences users can't replicate elsewhere—and it calls out “hero products” like PearLens and PearDesk as strategic levers.

Current adoption baseline (helps pick where to innovate)

The usage chart shows PearLens is the smallest installed base (~10M) vs PearPods (~50M), PearPads (~35M), PearBand (~20M), PearDesk (~15M). That suggests PearLens needs a killer differentiated use case to grow/retain.

2) User segmentation (who to build for first)

Pear already names 3 big segments. A sharper segmentation (demo + psychographic) that leads naturally to your solution:

Segment A — “Creative Power Users (and aspiring creatives)”

- Demographic: 16–35 (students + early-career), creative majors, hobbyist creators
- Psychographic: identity-driven (“I want to be a creator”), impatient with friction, learns by doing
- Context: edits in Photoshop/Lightroom/Premiere/Figma; lots of tool switching and hidden features

Segment B — “Workflow-Heavy Professionals”

- Demographic: 22–45 knowledge workers
- Psychographic: time-poor, outcome-driven, hates “figuring it out,” optimizes workflows
- Context: Excel/Sheets, slide tools, dashboards, enterprise SaaS

Segment C — “Leisure / casual users”

- Demographic: broad
- Psychographic: wants “just works,” lower tolerance for setup
- Context: common tasks, but less ROI for deep step-by-step in pro apps

Chosen focus segment (to drive mission): Segment A (Creatives + aspiring creators).
Reason: It aligns best with Pear's "productive and creative lives" mission and gives PearLens/PearDesk a signature, defensible experience beyond specs.

3) Needs / pain points / JTBD (what's broken)

Core JTBD (Creative segment)

"When I'm trying to create/edit something in a complex tool, help me accomplish a specific outcome *right now* without breaking my flow."

Pain points (why current solutions fail)

- Tutorial mismatch: YouTube/blog steps don't match *my current screen*, version, or context.
- Context switching cost: alt-tab, pause/rewind, hunt menus → kills creative momentum.
- Low discoverability: pro tools bury features; novices don't even know what to search.
- Cognitive overload: too many choices; users need *the next step*, not a lecture.
- Privacy / friction: screen-sharing or cloud processing feels risky for personal files/work.

Prioritized pain point (the one you solve)

"I can't reliably translate 'what I want' into the correct next action on *my* screen, in real time, without leaving the app."

4) The problem statement (crisp framing)

Creatives using complex software lose time and momentum because existing help tools aren't context-aware to their exact screen state; they require search, switching, and interpretation—creating friction that prevents users from completing tasks and improving skills.

5) Opportunity + constraints (why on-device, why agentic)

Opportunity

Pear can win by making creation tools feel guided, ambient, and frictionless—a "Pear-like" experience that increases retention and ecosystem lock-in.

Constraints (drive architecture)

- Must be fast (sub-second guidance)
- Must be private (screen + files stay local when possible)

- Must be cross-app (Photoshop today, something else tomorrow)
- Must be low cognitive load (PearOS philosophy)

These constraints strongly push toward edge/on-device AI + agentic navigation.

6) Brainstormed solution set (before picking the final)

Here are 5 plausible solutions you would have explored *before* landing on the on-device navigator:

1. Contextual help search inside PearOS
 - Smart search that surfaces relevant help docs based on active app + keywords
 - Weakness: still makes user interpret steps and hunt UI
2. Interactive template library
 - “One-click workflows” / presets for common edits (e.g., portrait retouch pack)
 - Weakness: only covers pre-defined tasks; doesn’t teach or generalize
3. Remote expert / co-pilot (human-in-the-loop)
 - Tap to connect to an expert who guides you
 - Weakness: expensive, scheduling latency, privacy issues, not scalable
4. AR overlay tutorials (static)
 - PearLens overlays arrows/labels for known sequences
 - Weakness: brittle; breaks with version/UI changes; limited coverage
5. On-device agentic step-by-step navigator (your final)
 - Reads current screen state, plans, and guides user *in situ*
 - Strength: adapts to the user’s exact screen context, keeps flow, privacy-preserving

7) Prioritization framework (why solution #5 wins)

Using a simple RICE-style lens (Reach, Impact, Confidence, Effort) like the rubric encourages:

Product Critique Rubric

- Reach: how many Pear users can benefit (especially via PearDesk + PearLens)
- Impact: reduces time-to-completion + increases creative confidence
- Confidence: technical feasibility + user desirability
- Effort: engineering + partnerships

Why #5 wins: it scores highest on *Impact* and *differentiation/lock-in*, and it uniquely matches Pear’s privacy + low-friction design philosophy.

8) Final solution (what it is, in one line)

An on-device, agentic “AI Step Navigator” that understands what’s on the user’s screen, asks the minimum clarifying question, and provides real-time, step-by-step, on-screen guidance to complete a task (e.g., “remove background in Photoshop,” “color-grade this photo,” etc.).

This also “fits” PearLens’s stated intent: overlay context-aware info in the user’s field of view, but

9) Success metrics (what you’d measure)

To keep it measurable (rubric requirement):

Product Critique Rubric

Core product metrics

- Task completion rate (% who finish the intended outcome)
- Time-to-completion (median time from request → done)
- Step friction (undo/backtrack rate, “I’m stuck” events)
- Retention uplift (D7/D30 for users who try Navigator vs not)
- Cross-device usage (PearDesk ↔ PearLens handoff rate)

Business/ecosystem metrics (ties back to Pear’s stated focus)

- PearLens/PearDesk adoption growth and engagement patterns
- Services revenue mix if this becomes a premium “PearCreate+” service

Notes from Theresa:

Wants to see more on segmentation → intrigued by “older” segment who might need re-skill/up-skill

An interesting idea...needs to be more granular, especially when focusing on the chosen segment we’re building it out for. Make sure the features map to hardware devices (mobile, desktop, on the go, etc.)

Do we need to build a laptop → is it a super heavy processor? Is it a scaled down chrome book? Why does Pear not already have a laptop? Again, we need to understand the use case and how that drives the hardware angle that we’re trying to build into. How do we make this user centric?

At Roblox, there are segments that use ipad/tablets (security and customer e.g.)

What is the subsegment that Pear isn’t targeting? At the end of the day, we want to ship this product - i.e. cost IS a factor!

First, go down to a user segment where you can see the user in your mind. If you're not there, then you haven't gone deep enough to empathize with the user. Do the axis (quadrants) multiple times until it crystallizes who you have in your mind.

For the MVP, we need to be *SUPER* clear about who this user is → talk to 5 people, ideally from different segments, to validate the approach we're taking